Hefbomen voor een beleid gericht op duurzame ontwikkeling

INTEGRATIE VAN VERKEERS- EN ECONOMISCHE MODELLEN

Voor Evaluatie van Stedelijk Transportbeleid

ITEM

MD/01/026

Prof. STEF PROOST, KURT VAN DENDER
Centrum voor Economische Studiën
KULeuven
Naamsestraat 69
B-3000 Leuven

Prof. PHILIPPE TOINT, ERIC CORNELIS
Groupe de Recherches sur les Transports
FUNDP
Rempart de la Vierge 8
B-5000 Namur

Inhoud

1. Inleiding

De doelstelling van het ITEM consortium is om economische prijszettingsmodellen en verkeersmodellen (netwerkmodellen) te verenigen, om zodoende de evaluatie te verbeteren van stedelijk transport- en milieubeleid.

Economische modellen van stedelijk transport benadrukken de totale transportvraag per transportmodus, en evalueren beleidsmaatregelen in termen van welvaart (inclusief de kosten van congestie, milieuvervuiling, ongevallen en lawaai). Verkeersmodellen leggen de nadruk op de invloed van kenmerken van het wegennetwerk op verkeersstromen. De combinatie van economische modellen en verkeersmodellen in één methodologie, maakt gedetailleerde analyse mogelijk van een breed spectrum van nieuwe beleidsmaatregelen voor stedelijk transport.

De constructie van het ITEM model vertrekt van twee bestaande modellen : TRENEN II URBAN (CES), een economisch prijszettingsmodel, en ATES (GRT), een typisch verkeersmodel. Beide modellen worden samengevoegd door de netwerkstructuur van ATES te vereenvoudigen, en door de definities van evenwicht en van kosten congruent te maken. De samenvoeging vindt plaats door een geautomatiseerde interactie van de modellen.

Het nieuwe model wordt gebruikt om de introductie van nieuwe vormen van stedelijk transportbeleid (rekeningrijden) te testen, in een experimentele gevalstudie voor Namen.

Het voorgestelde onderzoek valt volledig samen met de vereisten onder hoofding 3 “Responses” van het onderzoeksprogramma inzake duurzame ontwikkeling, en meer specifiek duurzame mobiliteit. Het behandelt de optimering van vooral prijszetting, voor de sturing van modale keuze en van de totale mobiliteit. De doelstelling bij de evaluatie van transportbeleid is de sociale welvaart, omvattende :

- waardering van het milieu via de monetaire waarde van externe kosten zoals lawaai, luchtverontreiniging (ozon, kleine deeltjes, broeikasgassen,...);
- aspecten van verkeersveiligheid via de externe ongevalskosten van de verschillende modi;
- private kosten (tijd, resource kosten, transactiekosten);
- resource kosten van publiek transport en problemen i.v.m. overheidsuitgaven (via de marginale kost van overheidsinkomsten).

Dit is een systematische en allesomvattende manier om duurzame mobiliteit in zijn multidimensionele aspecten te bepalen.

Het hier voorgestelde onderzoek is een voortzetting van onderzoek dat werd geinitieerd door het Programma Transport en Mobiliteit van de DWTC en door het European Communities Transport Research Programme. Bovendien is interactie voorzien met complementair onderzoekswerk, dat reeds werd gestart in de eerste fase van het Programma Duurzame Mobiliteit (consortium “Externe kosten van transport”).

De economische analyse van prijszetting in transportnetwerken krijgt recent meer aandacht in de economische literatuur. Het onderzoek binnen het ITEM-project heeft dan ook toegelaten om bestaande contacten van het CES te verstevigen en uit te breiden. In het bijzonder werd, op informele wijze, samengewerkt met Dr Erik Verhoef (Vrije Universiteit Amsterdam) en Prof. Ken Small (University of California, Irvine). Delen van het onderzoek werden voorgesteld tijdens workshops aan de KULeuven, aan de University of Essex en aan de University of California, Irvine. Verder werden twee onderzoekspapers voortvloeiend uit het project aanvaard voor de World Conference on Transport Research (WCTR) in Seoul, Korea (juli 2001).

2. Samenvatting van de methodologie en de resultaten

2.1 Situering

Overdreven tijdverlies, een verminderde betrouwbaarheid van het transportsysteem, ongevallen, milieu-verontreiniging en gezondheidsschade worden beschouwd als belangrijke economische en sociale problemen die verbonden zijn met de transportsector. Deze problemen krijgen ook veel aandacht in de economische literatuur. Vaak wordt rekeningrijden aanbevolen als zijnde een noodzakelijk element in een beter, en duurzaam, transportbeleid. Hierbij passen twee opmerkingen. Ten eerste wordt, in de economische analyse van rekeningrijden, vaak (impliciet) aangenomen dat een heffing kan worden ingevoerd op het volledige verkeersnetwerk, door middel van technologische oplossingen die zelf geen congestie veroorzaken. Ten tweede wordt congestie vaak op zich bestudeerd, zonder rekening te houden met andere specifieke kenmerken van de transportsector.

De twee genoemde bemerkingen geven aanleiding tot de onderzoeks-onderwerpen van het ITEM project. Er wordt nagegaan hoe een beleid van van rekeningrijden, dat een directe vertaling is van het principe van internalisering van externe kosten, wordt aangepast wanneer niet alle verbindingen in een verkeersnetwerk onderhevig zijn aan een heffing. Daarnaast wordt ook bekeken hoe het principe van rekeningrijden dient te worden aangepast wanneer er schaalvoordelen (van een specifiek type) aanwezig zijn in publiek transport, en wanneer een belangrijk aandeel van de verkeersstroom bestaat uit verplaatsingen van en naar het werk. Bij deze laatste interactie is het belangrijk om rekening te houden met het feit dat de huidige belastingen op arbeid hoog zijn.

De onderzoeksonderwerpen van het ITEM project kunnen worden beschouwd als uitbreidingen van de basis-analyse van de prijszetting in de transportsector. We beginnen in de volgende sectie met een samenvatting van de resultaten van de basisanalyse, voor een gevalstudie voor Brussel. Vervolgens worden, tegen deze achtergrond, de resultaten van het ITEM project samengevat. De bijlagen beschrijven het onderzoekswerk in detail.

2.2 Resultaten van de basis-analyse

De huidige transportprijzen weerspiegelen niet de volledige sociale kosten van transport. Ten eerste worden de transportbelastingen niet gebruikt om de externe kosten van congestie, luchtvervuiling, ongevallen en lawaaihinder te internaliseren. Ten tweede worden ook sommige directe resource-kosten niet aangerekend, bijvoorbeeld wanneer gratis parkeerplaatsen ter beschikking zijn. Een toepassing van het TRENEN model (zie Proost en Van Dender, 2001, voor toelichting bij de methodologie; zie Van Dender, 2001, voor een gedetailleerde beschrijving van de gevalstudie voor Brussel) voor Brussel in 2005 (onder de veronderstelling van ongewijzigd prijsbeleid ten opzichte van 1996), geeft de volgende grootte-ordes voor de kloof tussen prijzen en sociale kosten:

- De resource-kost van parkeren (0.13 Euro/km) wordt niet aangerekend voor de meerderheid (70%) van de verplaatsingen. Deze kost staat voor ca 40% van de totale verplaatsingskost voor een gemiddelde verplaatsing.
- De externe kost van congestie bedraagt ca. 1.79 Euro/km tijdens de piekuren. Toevoeging van externe kosten van luchtvervuiling, ongevallen en lawaaihinder brengt de marginale externe kost per kilometer op 1.83 Euro.

De invoering van een optimaal prijsbeleid met betrekking tot parkeren en met betrekking tot externe kosten leidt tot een verhoging van de maatschappelijke welvaart met 1.3%. Het dagelijkse verkeersvolume neemt af met 9%. Tegelijk vindt een belangrijke modale verschuiving plaats van verplaatsingen met de wagen naar verplaatsingen met het openbaar vervoer, tijdens de piekuren. De optimale transportprijzen (inclusief tijdskosten!) tijdens de piekuren nemen met ca. 40% toe.

Zoals vermeld veronderstelt de gemaakte oefening dat de optimale heffingen worden ingevoerd op het volledige verkeersnetwerk, en wordt niet expliciet rekening gehouden met schaalvoordelen in publiek transport en met de interactie met de arbeidsmarkt. De volgende secties vatten de resultaten samen van het weglaten van deze vereenvoudigende veronderstellingen. De nadruk in deze samenvatting ligt op de inhoudelijke conclusies. Gedetailleerde besprekingen van resultaten, en beschouwingen van methodologische aard worden uitgewerkt in de bijlagen.

2.3 Partiële netwerk-heffingen (bijlagen 1, 2 en 3)

<u>Theoretische analyse</u>

Prijsinstrumenten voor de internalisering van marginale externe congestiekosten zijn niet perfect wanneer de heffingen niet volledig gedifferentieerd kunnen worden over tijd en ruimte. Differentiëring van heffingen is duur, en volledige differentiëring wellicht onmogelijk. Bijgevolg zijn imperfecte heffingen de regel, en niet de uitzondering. Een analyse van het probleem van beperkte ruimtelijke differentiëring van congestieheffingen moet gebeuren binnen een ruimtelijk model van prijszetting. De basis-benadering is om een statisch verkeers-netwerkmodel te combineren met een economisch model van optimale belastingen.

In een statisch netwerkmodel gedragen de gebruikers van het netwerk zich volgens de Wardropiaanse principes. Dit wil zeggen dat de gebruikers hun private kosten zo laag mogelijk maken, zonder te coördineren met andere gebruikers, en onder de veronderstelling van perfecte informatie. Het gevolg is dat de gemiddelde reistijden minimaal worden, en dat de reistijden op alle werkelijk gebruikte routes tussen een oorsprong en een bestemming gelijk zijn. Omwille van de congestie-externaliteit is het Wardropiaanse netwerk-evenwicht niet efficiënt vanuit het maatschappelijk standpunt. In de maatschappelijk optimale oplossing worden de marginale reistijden –in plaats van de gemiddelde- gelijk op alle gebruikte paden. De maatschappelijk optimale oplossing kan enkel worden bereikt door perfecte heffingen (gelijk aan de marginale externe congestiekost) op alle netwerkverbindingen. Het ITEM-project gaat na wat de optimale prijsregels zijn wanneer niet alle verbindingen in het netwerk perfect belast kunnen worden (partiële netwerk-heffingen).

De optimale partiële netwerk-prijszettingsregel bepaalt dat de belasting op elke belastbare verbinding afhangt van de marginale externe congestiekost op de verbinding, en van netwerk-interacties. Inzake netwerk-interacties kunnen drie belangrijke effecten worden onderscheiden. Ten eerste, als de belastbare verbinding deel uitmaakt van een route waarvoor niet-belaste alternatieven beschikbaar zijn, zakt de optimale heffing onder de marginale externe congestiekosten op de verbinding. Deze lagere belasting vermijdt overmatige heroriëntering van verkeersstromen naar de niet-belaste routes. Ten tweede, het belasten van een verbinding die deel uitmaakt van een langere route, leidt tot een opwaartse druk op de optimale heffing. De reden is dat de heffing op de verbinding wordt gebruikt om de marginale externe congestiekosten op de volledige route te internaliseren. De eerste twee effecten hebben betrekking op de efficientie van het gebruik van het netwerk voor een gegeven vraag naar verplaatsingen. Het eerste effect is gewoonlijk groter dan het tweede, zodat de optimale belasting onder de externe kosten op de verbinding ligt. Ten derde, echter, kan de optimale belasting op een verbinding boven de marginale externe congestiekost stijgen, omwille van het globale vraagreducerend effect van de belasting. De niet-geïnternaliseerde congestie-externaliteit leidt namelijk niet enkel tot inefficiënt netwerkgebruik, maar ook tot een excessieve vraag naar verplaatsingen. Wanneer een belasting op een verbinding in eerste instantie gebruikt wordt om de gloable transport vraag te verlagen, is het optimale niveau vaak hoger dan de marginale externe congestiekost op de link. Welk van de drie effecten domineert, is een empirische vraag (zie verder).

De bepaling van optimale netwerkprijzen dient rekening te houden met het feit dat het netwerk gebruikt wordt door huishoudens met verschillende locaties. De verschillende locaties impliceren direct dat de transportkosten voor identieke bestemmingen verschillend zijn voor verschillende huishoudens. Deze verschillen worden weerspiegeld in de optimale prijsregels. Het theoretische belang hiervan is dat de optimale heffingen verschilllend zijn van de marginale externe congestiekosten, zelfs als alle verbindingen in het netwerk belast kunnen worden. Dit impliceert dat de verdelingseffecten van netwerkheffingen dienen geëxpliciteerd te worden in het nagestreefde welvaartsobjectief.

Gebruik makend van een illustratief model, wordt aangetoond dat partiële netwerk-heffingen goede resultaten kunnen opleveren, op voorwaarde dat de juiste verbindingen belast worden op een manier die de netwerk-interacties weerspiegelt. Verder wordt gesuggereerd dat alternatieve beleidsinstrumenten, zoals bijvoorbeeld parkeerheffingen, ook goed presteren, wanneer de initiële verdeling van verkeersstromen over het netwerk niet te inefficient is. De reden hiervoor is dat deze instrumenten geschikt zijn om de beoogde vraagverminderingen te bewerkstelligen.

Toegepaste analyse

Binnen het ITEM project werd een simulatiemodel ontwikkeld voor de evaluatie van partiële netwerk-heffingen in algemene statische netwerkmodellen. Het simulatiemodel bestaat uit een vraagmodule, waarin het consumentenevenwicht wordt berekend, en uit een netwerkmodule, waarin het netwerkevewicht wordt bepaald. Het simulatiemodel itereert tussen beide modules tot een simultaan evenwicht in beide modules bereikt wordt.

De reden om te kiezen voor een simulatiemodel, in plaats van een optimeringsmodel, is dat een optimeringsmodel geconfronteerd wordt met een niet-continue objectieffunctie. Deze discontinuiteit vindt haar oorsprong in de complementariteitsvoorwaarde, die het netwerkevenwicht kenmerkt. Intuitief is het probleem dat verschillende prijzen kunnen leiden tot het gebruik van verschillende routes voor eenzelfde oorsprong en bestemming. De verzameling van gebruikte routes is bijgevolg endogeen. Dit is een discreet aspect, in een overigens continu optimeringsprobleem. Gebruiksklare algoritmes voor dit soort problemen zijn (nog) niet voorhanden.

Het simulatiemodel wordt toegepast op een gestileerd, maar niet triviaal netwerkmodel voor Namen. Het model wordt gecalibreerd op een dataset met vraaggegevens voor de ochtendpiek tijdens een gemiddelde werkdag. Door middel van een ‘grid search’ techniek (d.w.z. berekening van alle relevante oplossingen van het model), wordt vastgesteld dat optimale belastingen op een beperkt aantal netwerkverbindingen zeer effectief zijn in termen van sociale welvaart. Een systeem van optimale belastingen op vier (van een totaal van dertig) netwerk-verbindingen levert driekwart van de welvaartswinst van een systeem met optimale belastingen op alle (dertig) netwerk-verbindingen. Het optimale systeem van vier netwerkverbindingen illustreert dat de netwerkinteracties die in de theoretische analyse geidentificeerd worden, van belang zijn. Twee van de vier verbindingen is vooral interessant vanwege de vraagreducerende functie. De andere twee hebben hoofdzakelijk betrekking op het efficientere gebruik van het netwerk. Verder wordt ook geïllustreerd dat de verzameling van gebruikte paden inderdaad varieert naargelang het systeem van heffingen. De discontinuiteit van het welvaartsobjectief in een optimeringsmodel is dus niet enkel een theoretisch curiosum.

De belangrijke beleidsconclusie van de analyse is dat beperkte systemen van congestie-heffingen zeer effectief kunnen zijn, mits ze zorgvuldig ontworpen worden.

Toepassingen van de simulatie-techniek op grotere netwerken zijn op dit moment nog in ontwikkeling. Hiervoor werd een efficiënte software ontworpen en getest.

2.4 Schaalvoordelen in publiek transport (bijlage 4)

Analyses van congestieheffingen zijn vaak beperkt tot prive-voertuigen. Uitbreidingen naar publiek transport houden in de regel geen rekening met schaalvoordelen in publiek transport. Een specifiek type van schaalvoordelen betreft de voordelen van dichtheid van de vraag. Wanneer de vraag voor een gegeven regio toeneemt, kan de publiek transport operator reageren door de frequentie van de dienstverlening te verhogen. Hierdoor verlaagt de gemiddelde wachttijd voor elke gebruiker van publiek transport.

Wanneer in de analyse van optimale prijszetting van stedelijk transport met de voordelen van vraagdichtheid rekening gehouden wordt, is het belangrijkste effect dat de optimale prijzen van publiek transport neerwaarts herzien worden. Het effect op de optimale prijzen van privé voertuigen is klein. Als enkel de prijzen van publiek transport geoptimaliseerd worden, terwijl de prijzen van privé voertuigen op het huidige niveau blijven, is het mogelijk dat de optimale prijs voor publiek transport tijdens de spitsuren gelijk is aan nul. Tijdens de daluren hebben nultarieven echter een negatief effect op de welvaart.

2.5 Interacties tussen congestieheffingen en belastingen op arbeid (bijlage 5)

Een belasting op verplaatsingen van en naar het werk is uiteindelijk een belasting op arbeid, onafhankelijk van de aanwezigheid van (congestie-)externaliteiten. In het geval dat alle woon-werk-verplaatsingen per auto gebeuren, en woon-werk-verplaatsingen het enige verplaatsingsmotief vormen, is het van geen belang of een belasting op arbeid of een belasting op de verplaatsing wordt gebruikt om belastinginkomsten te genereren (tenminste als de woon-werk-verplaatsing onvermijdelijk samenhangt met het arbeidsaanbod). Als er meerdere transportmodi beschikbaar zijn, zijn correcte relatieve prijzen nodig om de sociaal efficiënte modale distributie van woon-werk-verplaatsingen te bekomen. Bij aanwezigheid van meerdere verplaatsingsmotieven, is het wenselijk dat de heffingen verschillen tussen de motieven.

Door middel van een numeriek model wordt het belang van de vermelde interacties geanalyseerd. Differentiatie van congestie-heffingen tussen verplaatsingsmotieven is belangrijk in termen van sociale welvaart, zeker wanneer de belastingen op arbeid vastliggen. De beleidsconclusie is dat, wanneer rekeningrijden ingevoerd wordt, de heffingen aanleiding moeten geven tot een verlaging van de belastingen op arbeid (via een directe verlaging, of via fiscale aftrekbaarheid).

3. Besluit

De analyses binnen het ITEM project suggereren dat de conclusie van de basisanalyse overeind blijft: congestieheffingen zijn mogelijk en wenselijk. Congestieheffingen zijn wenselijk in de zin dat een differentiatie van de transportprijzen in overeenstemming met de marginale externe kosten, welvaartsverbeterend is. De welvaartsverbetering volgt uit de verhoogde efficiëntie van het stedelijke transportsysteem. De heffingen zijn mogelijk, ook wanneer rekening gehouden wordt met de beperking dat niet het volledige verkeersnetwerk aan heffingen onderhevig kan zijn. De analyses maken echter duidelijk dat het ontwerp van een beperkt systeem van congestieheffingen niet eenvoudig is, en dat eenvoudige regels (zoals het belasten van de drukste verbindingen) in een aantal gevallen contra-productief kunnen zijn. Er wordt verder ook aangetoond dat het belangrijk is om de heffingen voor woon-werk verplaatsingen verschillend te maken van heffingen voor verplaatsingen in verband met vrije-tijdsbesteding.

De prijszetting van publiek transport kan niet los gezien worden van de prijsstructuur van het private transport. Wanneer de prijzen van het private vervoer nauwer aansluiten bij de optimale prijzen, kunnen de prijzen voor het publiek vervoer overeenkomstig stijgen. Dit verbetert de financiële situatie van het publieke vervoer, en laat toe de kwaliteit van de dienstverlening te verbeteren.

Bijlagen

Bijlage 1	**Pricing transport networks with fixed residential location**
Bijlage 2	**Construction of a simulation model for Namur**
Bijlage 3	**An experimental case study for Namur**
Bijlage 4	**Economies of density and optimal transport pricing**
Bijlage 5	**Congestion pricing with multiple trip purposes**

Bijlage 1

Pricing transport networks with fixed residential location

1. Introduction

Economic theory suggests that congestion pricing is the preferred way to internalise traffic congestion externalites. Most often the congestion pricing literature has abstracted from spatial aspects in discussing and extending this result. Our goal is to investigate the properties of a welfare maximising congestion pricing system in the context of a congestible static traffic network. More specifically, we consider a network that is simultaneously used by multiple households, who employ several transport modes to produce trips to various destinations. [1] The destinations may or may not be substitutes. The residential location of all consumers is taken to be fixed. Government is assumed to be a social welfare maximiser, with or without access to individualised lump sum transfers and –possibly restricted- taxes on network links. The paper covers three topics.

First, it is shown that the non-convexity in household location choice, which leads to the choice of only one location, implies in general that equal households have different utility levels at the social welfare optimum. This is called Mirrlees inequality, after Mirrlees (1972) who first noted the problem. It follows from the fact that transport costs for the same destination differ between households. We briefly discuss some ways of dealing with this issue. In the Herbert-Stevens model, the social surplus is maximised subject to target utility levels for all households (Fujita, 1990). Since to any Herbert-Stevens model corresponds a social welfare function with particular welfare weights and appropriate lump sum transfers for all households, we will rely on the social welfare maximisation approach for the description of the optimal pricing conditions.

Next the optimal network pricing conditions are derived, allowing for constraints on the set of links which can be taxed, i.e. allowing for partial network pricing. The constraints may be interpreted as limiting congestion pricing to some transport modes, or to some links. In order to focus on the effect of partial pricing we first allow for optimal, individualised lump sum transfers. Next we elimitate the individualised lump sum transfers, using tax revenue redistribution according to ex ante fixed shares instead. The interaction between optimal network prices and Mirrlees inequality is analysed, as well as the role of revenue redistribution.

Finally, simple networks are used to illustrate the impact of Mirrlees inequality and of pricing constraints on the effectiveness of transport pricing schemes in terms of social welfare. The use of a network structure allows to compare various pricing instruments: link congestion tolls, pricing of transport modes, parking charges, transit pricing, uniform pricing, etc. We mainly limit ourselves to partial pricing schemes and to parking charges. It is illustrated that the effectiveness of imperfect schemes may be drastically smaller than that of first best

---

[1] The model is general enough to allow for multiple time periods (e.g. peak, off-peak) and different trip purposes (e.g. leisure-related trips and commuting trips). These topics are not developed in this paper however.

pricing, and that parking charges outperform partial pricing schemes under some conditons. Furthermore, assumptions on tax revenue redistribution strongly affect the optimal tax structure.

The main contributions of the present analysis are (a) to emphasise the impact of transport cost differences, caused by different locations, on the welfare properties of an optimal pricing scheme, (b) to provide a second-best network pricing rule within a simple general equilibrium context, and (c) to clarify that optimal network prices strongly depend on the particular welfare objective, especially when optimal lump sum transfers are not available. Furthermore, the analysis allows for substitution between destinations and for multimodality, while extension to a multiperiod framework is straightforward.

The most important analytical simplification is the assumption that location is fixed. Apart from the fact that this requires us to allow for individualised rather than uniform lump sum transfers, the assumption is not very restrictive. In particular, endogenising household location choice, e.g. by introducing a land market, would add to the analytical complexity but would not sidestep the issue of Mirrlees inequality. This problem is caused by the non-convexity of preferences, hence it also appears in endogenous location models. The interaction between transport prices and residential location choice is abstracted from in this paper, but we do allow for changes in choice of destination in response to changing transport prices. Therefore our analysis can be taken to represent a short to medium run assessment of the spatial impacts of transport network pricing.

Section 2 ties the present analysis to various strands of the congestion pricing literature. The theoretical analysis, with the discussion of the social welfare objective and the optimal pricing conditions, is in section 3. Section 4 discusses the effectiveness of partial pricing schemes by means of an example. Section 5 concludes.

2. Literature on partial transport network pricing

We briefly discuss three relevant strands of the literature: (a) economic analyses of pricing of two parallel links, (b) analysis of partial network tolling in transportation science, and (c) economic analysis of pricing general static traffic networks.

The economic literature on congestion pricing has most often implicitly assumed that all links in the network can be taxed. Otherwise, the analysis of network effects has been limited mainly to the case of one origin-destination pair connected by two links. This strand of the literature can be traced back to Knight's (1924) comment on Pigou (1920), which states that inefficient pricing disappears under private road ownership, given sufficient competition. In the absence of competition for road ownership, a monopoly road owner will internalise marginal external congestion costs and charge a mark-up depending on the price elasticity of demand (Small, 1992). A related question concerns the situation where a privately and a publicly owned road between the same origin and destination co-exist. Various authors (Marchand, 1968; McDonald, 1995; Braid, 1996; Verhoef et al., 1996; Liu and McDonald, 1999; Arnott et al., 1996) have studied this problem, usually assuming that the private road is tolled and the public road is not.

The analytical results depend to some extent on the representation of congestion. The traditional flow congestion model represents the time cost of road use as an increasing function of flow (per unit of time). The bottleneck model describes the formation of a queue at a bottleneck, in case demand exceeds capacity of this bottleneck. Departure time is endogenous in this model.

Concerning one-link pricing in a two-link model, the fundamental results for the flow congestion model are that:

- The optimal tax is the result of a trade-off between the objectives of aggregate demand reduction and optimal assignment of traffic flows. The first objective pushes the tax up, the second may push it up or down, depending on the cost characteristics of the two routes.
- The relative efficiency of second-best taxes decreases as the price elasticity of demand rises.
- The relative efficiency of second-best taxes increases as the tolled route is shorter and less congestible.

The introduction of consumer heterogeneity increases the relative welfare gain obtained from partial pricing schemes: pricing only one road leads to separating equilibria, which mitigates the welfare loss of the partial pricing scheme (Verhoef and Small, 1999; Small and Yan, 2001). In line with earlier studies on equity effects of road pricing, these models predict welfare gains or losses for users with different values of time (different user classes).

The basic insight from the bottleneck congestion model is that the optimal tax consists of a positive and time dependent component to prevent queueing on the tolled route and of a negative uniform component which is meant to alleviate congestion on the untolled route. De Palma and Lindsey (2000) analyse competition between two roads for a wider set of ownership arrangements and allow tolling on both roads.

The main results from the second best pricing literature in two link networks are that (a) second-best schemes can be effective for reallocating traffic flows, (b) second-best taxes are much smaller than first-best taxes, and (c) second-best taxes produce much smaller welfare gains than first-best taxes. These results are confirmed for a wide range of models and parameter values.

Congestion pricing in a multimodal network affects transport demand levels, mode choice and path choice. Path choice, or network assignment, is an important research area in transportation science. In the most basic static traffic network model (e.g. Sheffi, 1985), network users behave as travel cost-mimimisers, in a non-coordinated fashion. Wardrop (1952) shows that this leads to equal average travel costs for all used paths between an origin and a destination. This is not a welfare maximising solution, in which marginal travel costs are equalised over all used paths belonging to the same origin-destination pair. Only recently, attention has gone to the design of optimal partial pricing schemes. Contributions in this area focus on mathematical properties and on algorithm design. Construction of general algorithms is not straightforward given the combinatorial aspects of the problem (Larsson and Patriksson, 1998, Labbé et al., 1998; Ferris and Kanzow, 2000).

An economic analysis of optimal partial pricing systems in general static networks is in Verhoef (1998c), who employs a partial equilibrium approach. Demand for an origin-destination pair depends on the price of that origin-destination pair only, and social welfare is the sum of Marshallian consumer surplusses in all origin-destination markets. The optimal second-best link tax is seen to depend on the pricing conditions on other links, as well as on cost and demand interactions. It may be above or below external congestion costs on the link. The optimal second-best tax configuration is not necessarily unique. A first-best solution is to set all link taxes equal to marginal external costs on the link. Our analysis generalises the second-best pricing rule, drawing attention to the importance of assumptions on the availability of individualised lump sum transfers and on the redistribution of congestion tax revenues. Instead of the primal formulation of the optimal tax problem (which is implicit in the partial equilibrium analysis), we will be using the more convenient dual approach. It will be shown that the partial equilibrium results hold as long as individualised lump sum transfers are available, but not under other assumptions on tax instruments and tax revenue redistribution. It will, e.g., not be the case that marginal social cost pricing is optimal when all links in the network can be taxed.

3. Theory

This section starts with the representation of the network and its usage by consumers. Subsequently we discuss the government problem under first best and second best network pricing conditions.

3.1 Consumer equilibrium in a static transport network

Denote a static transport network by a graph $G(N,A)$, where N is a set of nodes and A a set of links. The graph is strongly connected, so that each node can be reached from every other node through at least one sequence of links. Such a sequence is called a path. Links are congestible when time costs (c_a) are increasing in traffic flow on the link (f_a, in passenger car units per unit of time): $\left(\frac{\partial c_a}{\partial f_a} \equiv c'_a\right) \geq 0, \forall a \in A$.[2] The network technology is simplified by assuming $\frac{\partial c_a}{\partial f_b} = 0, \forall a \neq b \in A$.

Consider N_i identical consumers at each trip origin $i=1,\dots,I$ in the network ($i \in N, \forall i$). The trip origin coincides with residential location, and it is fixed. Consumers' utility functions (2.1) are defined over a composite numéraire commodity x_i and over transport commodities $q^r_{i,j}$.[3] Each $j=1,\dots,J$ represents a trip destination ($j \in N, \forall j$). Trip origins and destinations are connected by paths r. The set of paths is denoted $P_{i,j}$. The utility functions are assumed to possess all regularity conditions to allow for an optimal taxation analysis.

$$U_i = U_i\left(x_i, \mathbf{q}^{\mathbf{r}}_{\mathbf{i,j}}\right), \forall i = 1,\dots, I \text{ where } j = 1,\dots, J \text{ and } r \in P_{i,j} \tag{2.1}$$

A path represents a route for a given transport mode, or it refers to different transport modes. Link flow is then defined as the sum of demands for all paths that use the link: $f_a = \sum_i \sum_j \sum_{r \in P_{i,j}} d^a_{i,j,r} N_i q^r_{i,j}, \forall a$.[4] The indicator variable $d^a_{i,j,r}$ equals 1 when *link a* belongs to path r, and zero otherwise.

Substitution between destinations is allowed for. In reality, the degree of substitutability depends on the trip motive and on the time horizon. Shopping destinations may be thought of as substitutable in the short run. Commuting destinations, or employment locations, probably only exhibit a significant degree of substitutability in the long run.

Each household faces a budget constraint (2.2). The exogenous income Y_i, the lump sum transfer T_i, transport time costs c_a and taxes t_a are stated in generalised terms. This implies that the value of marginal time savings or losses is taken to be constant. The consumer price

---

[2] Allowing for zero derivatives is useful for the introduction of virtual links, which may represent flow independent costs (e.g. parking costs, waiting times, or some types of taxes).

[3] Since transport is actually a derived demand, specifying transport as an argument of the utility function is a reduced form.

[4] In principle a conversion from trip demand to flow in terms of passenger car units is required, when transport modes with different occupancy rates are considered. We abstract from this in the theoretical analysis, for reasons of clarity.

of a commodity $q^r{}_{i,j}$ is the sum of time costs and taxes incurred on all network links which are used to reach destination j, starting from origin i and using path r.

$$Y_i + T_i = x_i + \sum_j \sum_r \left(\sum_a d^a_{i,j,r} (c_a + t_a) q^r_{i,j} \right), \forall i \tag{2.2}$$

Using l_i as multiplier for the budget constraint, the first order conditions for maximising (2.1) subject to (2.2) and non-negativity constraints, are given by equations (2.3) and (2.4). Households neglect the congestion externality. When paths are perfect substitutes, condition (2.4) implies that the prices on all paths $r \in P_{i,j}$ which carry positive flow are equal and not larger than prices on all paths $s \in P_{i,j}$ that do not carry any flow, see (2.5). This corresponds to the Wardropian network equilibrium conditions (Wardrop, 1952), which thus are a special case of the present specification. In order to simplify notation, the subsequent analysis abstracts from the complementarity condition by considering paths with positive flow only.

$$\frac{\partial U_i}{\partial x_i} = l_i, \forall i \tag{2.3}$$

$$\frac{\partial U_i}{\partial q^r_{i,j}} = l_i \left(\sum_a d^a_{i,j,r} (c_a + t_a) \right), \forall i, \forall j, r : q^r_{i,j} > 0 \tag{2.4}$$

$$\left(\sum_a d^a_{i,j,r} (c_a + t_a) \right) = \left(\sum_b d^b_{i,j,s} (c_b + t_b) \right) \text{ if } q^r_{i,j} > 0, q^s_{i,j} > 0 \tag{2.5}$$

The indirect utility function is given by $V_i(c_a+t_a, a=1,\ldots,A, Y_i+T_i)$, *for all i.* As congestion is not an argument of the utility function as such, external congestion costs, which on the link level equal $c'_a f_a$, are valued at the marginal utility of income.

3.2 Optimal congestion pricing in a static transport network

In section 3.2.1 the concept of Mirrlees inequality is introduced, using a simple network example. Next, section 3.2.2 presents the optimal tax structure in case government maximises social welfare by using individualised lump sum transfers T_i and a restricted set of link taxes t_a. Restrictions on link taxes are indicated by k_a, which equals one when a link tax is possible and zero otherwise. The nature of the restriction implies that when a link tax is possible, it can take any value. Allowing for a lump sum transfer simplifies the analysis, as the congestion externality is the only remaining inefficiency. The justification is that it allows us to focus on two main topics. First, the lump sum transfer clarifies how unequal treatment of equal households affects the optimal tax structure, as the transfer is used to equalise social marginal utilities of income. Second, it allows to emphasise the impact of restrictions on link taxes (partial network pricing) on optimal congestion taxes. The main simplification in section 3.2.2 is to allow for a lump sum transfer, not the fact that it is individualised to each household. The latter is required because of the absence of an equilibrating mechanism, such as a land market, in the model. The individualised transfer makes it possible for households with equal preferences to enjoy equal utilities at different residential locations.

In section 3.2.3 it is assumed that lump sum transfers can not be optimised, but that each household receives a given share of the congestion tax revenues. Optimal link taxes then are seen to interact with Mirrlees inequality. This is clarified by a decompostion of the link tax expression into three terms: a Pigouvian term, relating to external congestion costs on the link, a Ramsey-Mirrlees term, relating to transport cost differences, and a network interaction term, relating to restrictions on link taxes. The optimal link taxes will be seen to deviate from marginal external congestion costs on the link, also when all links in the network can be taxed.

3.2.1 Welfare maximisation with Mirrlees inequality

Mirrlees (1972) showed that in a spatial model of land use, maximisation of a social welfare function implies unequal treatment of equal households at the optimum. In the market equilibrium for a monocentric city model, households with identical preferences and incomes realise equal utilities at all locations. As households face different transport costs to the city centre, their marginal utilities of income differ. More specifically, when land rents compensate for the transport cost differences, the marginal utility of income is higher for households residing further out from the centre. This is exploited by the social welfare maximisation, in the sense that households with higher marginal utilities of income are advantaged. Hence the interaction with the land market implies that in a monocentric city households residing further out from the centre have higher utility in the social optimum (see e.g. Straszheim, 1987). Arnott and Riley (1977) show that the issue arises whenever production asymmetries exist, i.e. when a commodity can be produced more cheaply for one household than for another. Unequal treatment of equal households arises whenever a non-Rawlsian, symmetrical and quasi-concave social welfare function is used.

The source of the problem (in the monocentric city model as well as in the present model) is that restricting household location choice to the choice of one location, implies the assumption that preferences over locations are non-convex (Fujita, 1990). When locations differ between households, transport costs for the same destination differ. In the context of a static transport network, different locations imply the use of different paths to reach the same destination. The difference in transport costs causes the production costs for possibly identical consumers to differ, so that the utility possibility frontier is asymmetrical. This is illustrated in figures 3.1 and 3.2.

Figure 3.1 displays a network consisting of two *links a and b* with link costs $c_a>0$ and $c_b>0$. Household 1 uses *links a and b* (alternatively, *path ab*) to reach destination D. Household 2 uses *link b*. Both households have identical preferences (as described by equation (2.1)), but face different transport costs. We assume that in the no intervention equilibrium household utilities are equal. Since our model contains no compensating land market, the transport cost differences imply that the income for household 2 is lower than for household 1. The marginal utility of income is higher for household 2 than for household 1.[5] The utility possibility frontier for this economy (figure 3.2) is given by curve CB instead of CA. The Pareto-efficient point where both households are treated equally is D. This point can not be reached unless a Rawlsian social welfare function is used. In fact, the optimal point lies between the extremes D and E (e.g. E'), where E is optimal under a Benthamite social welfare

[5] Hence, in contrast to the monocentric city model, the marginal utility of income is higher for households living closer to the trip destination. This implies that in our model, the direction of the unequal treatment of households is reversed in comparison to the monocentric city model.

function (which exhibits no inequality aversion, and therefore maximally exploits the transport cost differences).

Figure 3.1 A two serial links network with two household locations and one destination

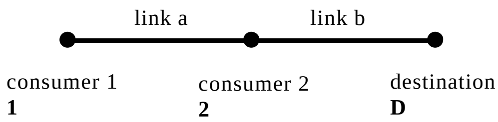
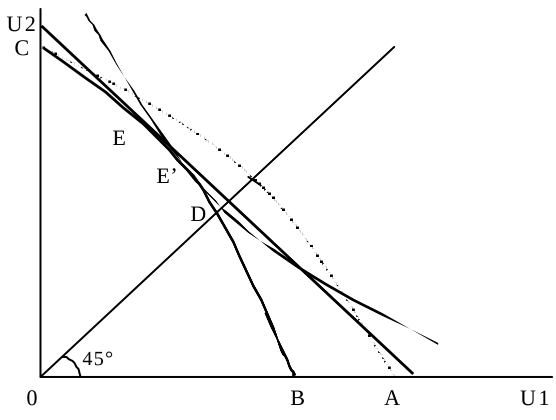


Figure 3.2 Utility possibility frontier and social welfare optima with differing transport costs



One way of dealing with Mirrlees inequality is to accept it. The optimal lump sum transfer and the associated market equilibrium can be computed. This equilibrium then is a benchmark to which the benefits of network congestion taxes can be measured. The government budget constraint for this case is given by (2.6). Alternatively, as unequal treatment of equal households may not be desirable from a normative point of view, it can be restricted. In general this can be done by restricting the transfers, or by imposing target utility levels.

$$\sum_i T_i = \sum_a k_a t_a f_a \tag{2.6}$$

The transfers can be restricted by requiring that they are financed only by congestion tax revenues. The congestion tax revenues can be redistributed to households in various ways. Equation (2.7) corresponds to a system where each household is compensated by exactly the amount of taxes it pays, while equation (2.8) refers to redistribution of tax revenues according to given shares s_i. As will be illustrated in section 3.2.3, restricting the transfers does not neutralise the social welfare impact of Mirrlees inequality. The optimal tax structure and the tax levels are affected by the tendency to increase the utility of households with low transport costs.

$$T_i = \sum_j \sum_r \left(\sum_a d^a_{i,j,r} k_a t_a q^r_{,j} \right), \forall i \tag{2.7}$$

$$T_i = \frac{s_i}{N_i} \sum_a k_a t_a f_a, \forall i, \text{ where } \sum_i N_i s_i = 1 \tag{2.8}$$

Imposing target utility levels can be done by computing Pareto-improvements, i.e. fixing utility levels of all but one households to the reference utility level. Alternatively a target utility level may be specified for all households. This is called the Herbert-Stevens approach (Fujita, 1990). The social welfare criterion then is to minimise the costs of attaining these utility levels. As incomes are taken to be fixed, this can be reformulated as maximising a social surplus, which is equal to total available income less the costs of reaching the target utilities. As the target utilities must be reached in a way consistent with household preferences, the minimal costs can be computed through the expenditure function. This function assumes that income is sufficient to attain the target utility, so that lump sum transfers are implicitly allowed. The surplus (deficit) which remains after subtraction of the costs for reaching the target utilities, is interpreted as a benefit (cost) to the rest of the economy.

There is no theoretical reason to prefer Herbert-Stevens over social welfare maximisation, as it can be seen that to each Herbert-Stevens model (i.e. to each set of target utilities) corresponds a social welfare maximisation programme with particular welfare weights for each household and with lump sum transfers available. This is the case as long as the optimal allocation as determined by a social welfare function, is invariant with respect to the total income which is available in the economy. The main characteristic of the Herbert-Stevens approach -full control over target utilities- may nevertheless be useful in computational applications, as no ex ante information on social welfare weights is required. The degree of unequal treatment of equals can be controlled by the modeller. The downside of the approach lies in the interpretation of the surplus as a transfer to the rest of the economy. For the further analysis the standard social welfare approach will be used, as it describes optimal pricing conditions for particular welfare weights or for particular utility targets.

3.2.2 Optimal network pricing with individualised lump sum transfers

Maximising programme (2.9) using transfers T_i and available link taxes t_a produces first order conditions (2.10) and (2.11). The lump sum transfer is used to equalise social marginal utilities of income, and the social welfare evaluation of the effect on indirect utilities of a link tax change is equalised to the social value of the tax revenue change.

$$\Im = W\left(N_i V_i, i=1,...,I\right) + m\left(\sum_a \left(\sum_i \sum_j \sum_r d^a_{i,j,r} k_a t_a N_i q^r_{i,j}\right) - \sum_i N_i T_i\right) \tag{2.9}$$

$$\frac{\partial W}{\partial V_i} = \frac{m}{l_i}, \forall i \tag{2.10}$$

$$\sum_i N_i \frac{\partial W}{\partial V_i}\frac{\partial V_i}{\partial t_a} + m\left(\sum_i \sum_j \sum_r d^a_{i,j,r} N_i q^r_{i,j} + \sum_b k_b t_b \frac{\partial f_b}{\partial t_a}\right) = 0, \forall a : k_a = 1 \tag{2.11}$$

Equation (2.12) gives the indirect utility effect of a marginal change in the link tax. Substituting (2.10) and (2.12) in all conditions (2.11) where $k_a=1$, using the definition of link flow, and solving the system of $\sum_a k_a$ equations for t_a, produces (2.13).

$$\frac{dV_i}{dt_a} = -l_i\left(\sum_r \sum_j d^a_{i,j,r} q^r_{i,j} + \sum_r \sum_j \left(\sum_b d^b_{i,j,r} c'_b\, q^r_{i,j} \frac{d f_b}{d t_a}\right)\right), \forall i,\ a : k_a = 1 \tag{2.12}$$

$$t_a = c'_a\, f_a + \sum_{z:k_z=1}\left(\frac{\sum_{b\neq a}\left(c'_b\, f_b - k_b t_b\right)\frac{d f_b}{d t_z}}{\frac{d f_a}{d t_z}}\right), \forall a : k_a = 1; b, z \in A \tag{2.13}$$

From (2.13) it is easily observed that when $k_a=1$ for all a, setting all link taxes equal to marginal external costs on the link ($t_a = c'_a f_a$) satisfies the optimality conditions. This is not necessarily the only possible tax configuration, as the network topology may allow equivalent tax systems. The condition is that taxes on each path for each consumer and each destination equal marginal external costs. In some networks this can be achieved without marginal social cost pricing on each link. An example is the network depicted in figure 3.1. If this network is used by one consumer located at point 1, taxing *link a* or *link b* for the full marginal external congestion costs of *path ab* is efficient. Adding a second consumer at point 2 leaves marginal social cost pricing on each link as the only possibility. In any non-trivial network with a large number of links, paths and consumers, the existence of equivalent systems becomes less likely.[6] We therefore conclude that in general the combination of marginal social cost pricing and optimal transfers is the only solution to the first best network pricing problem.

---

[6] The only degree of freedom in setting link taxes in a context of variable transport demand is the case discussed in the text. This degree of freedom disappears when the two links are collapsed into

Now consider the case where $k_a=1$ for one link in the network only, i.e. only one link can be taxed. (2.13) then reduces to (2.14). In the latter, the tax t_a differs from marginal external congestion costs by the marginal effect of a reduction in f_a, caused by a rise in t_a, on congestion on other links. This marginal effect on other links is corrected for the marginal effect on the link under consideration. The denominator is negative. The numerator takes either sign, leaving the sign of the correction factor undetermined in general. Observe that the other *links b* belong to 2 classes. Class 1 contains links that belong to paths which also contain *link a*: $C1=\left\{b\in A: d^{b}_{i,j,r}=d^{a}_{i,j,r}=1,\forall i,j,r\right\}$. Class 2 contains links that belong to paths that do not contain *link a*: $C2=\left\{b\in A: d^{b}_{i,j,r}=1 \text{ and } d^{a}_{i,j,r}=0,\forall i,j,r\right\}$. The intersection of C1 and C2 is non-empty, as any link can both be part of a path of which a link is taxed and of other paths which are not being taxed. For links in C1, a marginal increase in t_a decreases flow. For C2, flow increases as far as the links are used for substitute paths or substitute destinations. The sign of the second term in (2.14) depends on the relative importance of both classes of links in terms of flow, on the slope of the cost function of those links, and on the size of the flow reactions. When C2 is relatively large, t_a tends to drop below $c'_a f_a$. This occurs when alternative paths are available which are perfect substitutes, in which case a high t_a distorts the assignment of traffic on the network too much. A second possibility is that a consumer has the choice between two substitute destinations. Excessive taxation of the path to one shopping destination causes excessive congestion on paths leading to the other one. Only when t_a belongs to paths which can not be easily substituted, will the tax rise above marginal external congestion costs.

$$t_a = c'_a\, f_a + \frac{\sum_{b\neq a} c'_b\, f_b \dfrac{\partial f_b}{\partial t_a}}{\dfrac{\partial f_a}{\partial t_a}} \tag{2.14}$$

Returning to (2.13), it can be seen that the general second best tax expression has a similar interpretation to (2.14). One difference is that account should be taken of the deviation between taxes and marginal external costs on all other links. The second difference is that the marginal flow effects of all possible link taxes on all links matter for the determination of the optimal t_a. Section 4.3 illustrates the empirical relevance of the network interactions in terms of welfare effects and tax levels.

3.2.3 Optimal network pricing with restricted lump sum transfers

When all links in the network can be taxed and when optimal lump sum transfers are available, marginal social cost pricing constitutes a possible solution to the network pricing problem. Distributional issues are addressed through the transfers, whatever the social welfare weights are. This separation of tax functions holds when not all links can be taxed, as long as lump sum transfers can be optimised.[7] Congestion taxes will be set optimally

---

one, i.e. when the network graph is constructed with the minimal number of links. In a fixed demand setting, multiple tolling equilibria are the rule rather than the exception (Larsson and Patriksson, 1998).

[7] In other words, the presence of a non-convexity in location choice does not prevent application of the second theorem of welfare economics. The normative properties of the social

according to the first order conditions for the network taxes, implying deviations from marginal social cost pricing. Optimal transfers are used to optimally redistribute income, given the link taxes. The analysis changes when lump sum transfers are restricted, as this implies that link taxes are used for redistributional purposes, irrespective of link tax constraints.

In general, when optimal lump sum transfers are not available, it is the case that $t_a = t_a\left(c'_b\ f_b, k_b, \forall b \in A; V_i, \frac{\partial W}{\partial V_i} N_i, i=1,...,I\right)$, that is: link taxes depend on congestion and pricing constraints on the network, on preferences, and on the welfare weights of the households who use the network. Therefore, the unequal treatment of equal households is reflected in the optimal link taxes. Some insight into this interaction can be gained from considering a simple network (cfr figure 3.1). The network consists of 2 serial links, which connect the residence of two representative identical households to a common destination D. The demand for this destination is denoted D_1 and D_2. Both links are congestible. *Link b* is shared by both households. The flow on *link a* is D_1, the flow on *link b* equals D_1+D_2. In the no-tax equilibrium, incomes are such that household utilities are equal. This implies a higher income for household 1 than for household 2, and a higher marginal utility of income for household 2 than for household 1. Assume a Benthamite social welfare function (i.e. no correction for Mirrlees inequality through inequality aversion), and assume that congestion tax revenues are redistributed in a lump sum way according to a predetermined share s for household 1 and $(1-s)$ for household 2, see (2.8). Government then solves programme (2.15), which is equivalent to maximising the sum of indirect utilities after substitution of tax revenue into the indirect utility functions. Therefore, $l_i=\mu_i,\ \forall i$. Since transport costs are higher for household 1, we get that $\mu_1<\mu_2$. Denote this by $\mu_1=a\mu_2$, with $0<a<1$.

$$\begin{aligned}\Im = V_1 + V_2 + m_1\left(s\left(k_a t_a D_1 + k_b t_b\left(D_1 + D_2\right)\right) - T_1\right)\\ + m_2\left(\left(1-s\right)\left(k_a t_a D_1 + k_b t_b\left(D_1 + D_2\right)\right) - T_2\right)\end{aligned} \tag{2.15}$$

From the first order condition for t_a we get (2.16), or equivalently (2.17). It describes the condition for the optimal tax for a given level of t_b. Similar expressions can be found for t_b.

$$t_a = \left(\frac{a}{as-s+1}\right)c'_a D_1 - \left(1-\frac{a}{as-s+1}\right)\frac{D_1}{\frac{\partial D_1}{\partial t_a}} - \left(k_b t_b - c'_b \frac{aD_1 + D_2}{as-s+1}\right) \tag{2.16}$$

welfare optimum –unequal treatment of households with identical preferences- may not be appealing however.

$$t_a = \frac{wc'_a D_1 + \left(\kappa_b t_b - wc'_b \left(D_1 + \frac{D_2}{\alpha} \right) \right)}{1 - \frac{(1-w)}{\left| \varepsilon_{1a} \right|}} \tag{2.17}$$

$$\text{where } w = \frac{\alpha}{\alpha\sigma - \sigma + 1}$$

The first term on the right hand side of (2.16) states that a fraction of marginal external congestion costs on *link a* is charged. We call this the *Pigouvian component* of the link tax. The second term relates to demand effects, and it gets a weight equal to one minus the weight of the Pigouvian component. It is called the *Ramsey-Mirrlees component* of the link tax, where 'Ramsey' refers to its dependence on the price elasticity of demand, and 'Mirrlees' to the fact that the term appears because of the existence of transport cost differences. The weight of the first (second) term is positive and increasing (decreasing) in α and in σ. An increase in α implies an increase of the relative social welfare contribution of household 1, and causes the tax to be closer to the marginal external cost. In general, such an increase could follow from attaching a higher weight to household 1 in the social welfare function, or from a decrease in the transport cost difference between both households. An increase in σ means, in the example, that more of the tax revenue goes to household 1, the household with higher transport costs. This is equivalent to saying that congestion tax revenues can be used less easily to exploit Mirrlees inequality. Consequently, the tax is closer to a Pigouvian tax while the Ramsey-Mirrlees component drops. As can be seen from (2.17), the Ramsey-Mirrlees component implies that the tax decreases as the price elasticity of demand for D_1 rises.

The third term on the right of (2.16) is the *network interaction component* of the tax. It shows that t_a is corrected for the deviation of t_b from the social welfare value of congestion on *link b* (D_1 is weighted by α). The network interaction term is increasing in α and in σ, implying that using t_a as an indirect way of addressing congestion externalities on *link b* becomes more and more feasible as the social welfare contribution of household 1 increases. This stands to reason, as in the example only household 1 uses both links. Therefore, the larger the social welfare impact of household 1, the more the network tax system is directed towards internalising the externalities affecting household 1. The indirect effects on household 2 receive less attention.

Notice that t_a differs from marginal external costs on *link a* as long as $\alpha<1$, i.e. as long as transport cost differences exist. This is the case irrespective of whether *link b* can be taxed. When $\alpha=1$, the Ramsey-Mirrlees component reduces to zero, and t_a equals marginal external congestion costs on *link a* in case both links can be taxed (which implies a zero network interaction term).

To conclude, equations (2.16) and (2.17) illustrate that, in the absence of optimal lump sum transfers, optimal link taxes consist of a Pigouvian, a Ramsey-Mirrlees and a network interaction component. The importance of the components depends on the shape of the social welfare function, on the redistribution of congestion tax revenues and on network interactions (including pricing conditions in the network). All tax components are different from zero, also when all links in the network can be taxed optimally, because of the interaction between Mirrlees inequality and network taxes. This interaction disappears only when optimal lump sum transfers are possible or when all households share the same location.

4. Examples

4.1 Properties of the network examples

This section presents empirical illustrations of the main issues which were introduced in the theoretical analysis, on the basis of two examples. The examples are stylised and are not meant to generate definitive policy conclusions. Nevertheless, they are derived from realistic network cost and demand data (obtained from a network model for the city of Namur, Belgium), in order to obtain reasonable orders of magnitude for the model parameters. The appendix contains an overview of the background to the examples. First, the network of figure 3.1 is given a numerical content in section 4.2, so that the importance of the impact of Mirrlees inequality can be evaluated under various tax redistribution assumptions. Second, section 4.3 assumes that individualised lump sum transfers compensate each household exactly for the amount of congestion taxes it pays. This neutralises Mirrlees inequality as much as possible, allowing us to focus on the impacts of network pricing restrictions in a three link network.

The size of the examples is the minimal size which captures the relevant interactions described in section 3. For the analysis of the impact of Mirrlees inequality, a network of two serial links used by households at two different locations who have one common destination (see figure 3.1), is sufficient. The analysis of network interactions under partial pricing uses a three link network (figure 4.4). The impact of preference and cost characteristics on the effectiveness of partial pricing schemes is clarified by sensitivity analysis.

Determination of optimal link congestion taxes when not all links in the network can be taxed, is computationally difficult for a general network graph. Recent research in transportation science focusses on the design of algorithms for optimal partial network pricing (e.g. Labbé et al, 1998). Difficulties arise when the set of paths which are actually used for each origin-destination pair is endogenous. Introduction of link tolls may then induce the usage of paths which were not used before the price change, and vice versa. This turns the problem into a mixed integer or a combinatorial programme, which is difficult to solve, even for small networks (for a more detailed explanation, cfr Van Dender, 2001d). When the set of used paths is exogenous, the partial network pricing problem is an instance of inverse optimisation. In inverse optimisation, cost function parameters are found which turn a given solution into the optimal solution. The simplest case of a network in which the set of paths is exogenous, is a network in which all paths are used. This will be our assumption, so that standard nonlinear optimisation algorithms can be used (the models are programmed in GAMS (Brooke et al, 1996) and solved using CONOPT).

In all examples, household preferences are represented by nested CES functions, defined over a composite commodity and over trip destinations: $U_i = U_i\left(x_i, T_i\left(\mathbf{q_{i,j}}\right)\right), \forall i$, where T_i is the sub-utility function for transport commodities. The elasticities of substitution, which determine the curvature of the indifference curves, are chosen so as to approximate a price elasticity of transport demand of –0.3 in the reference equilibrium. This value is in line with the literature (e.g. Small, 1992). The remaining information required for the calibration of the utility functions is taken from the cost and demand data of the Namur network model. A Benthamite social welfare function is used, meaning that social welfare is the sum of household utilities. Finally we note that, while the theoretical analysis allows for negative taxes, they will be bounded below at zero in the applications.

4.2 Example 1: Mirrlees inequality in a two link serial network

The first example illustrates the impact of differences in transport costs on optimal network prices. For the network configuration depicted in figure 3.1, we assume that both links are 2 km long. The linear approximations to the user cost functions, the reference demands for the representative consumers at each origin, the link flows and the user costs are in table 4.1. The reference equilibrium is constructed so that the utility of households at both locations is equal. This implies that per capita income for households at location 2 is 4.2% lower that for those at location 1, in the assumption that the transport expenditures contained in the example represent 10% of total expenditures.[8] The remaining 90% is spent on a composite numéraire commodity. The monetary value of the marginal utility of income in the reference situation is 0.851 Euro for households 1 and 0.888 Euro for households 2.

Table 4.1 Cost and demand characteristics of the two link network (reference equilibrium)

Aggregate demand (trips)

1 – D	2583
2 – D	1227

Costs (Eurocent) and flow (vehicles)

	Intercept	Slope	Flow	Link travel cost
Link a	20	0.01	2583	45.83
Link b	45	0.01	3672	83.097

Full exploitation of Mirrlees inequality implies that income from households at location 1 is transferred to consumers at location 2 up to the point where the social marginal utility of income is equalised across households. In this particular example this implies transferring all income to households at location 2. This is the ultimate consequence of abstracting from land markets and from crowding externalities at each location. Possible alternatives are to compute Herbert-Stevens optima, or to restrict the amount of lump sum transfers to the amount of congestion tax revenue raised. With respect to Herbert-Stevens optima, it can be shown that any combination of target utility levels can be reached at lower costs when link taxes are available, and that any given utility level can be reached at lower cost for households at location 2 than for those at location 1. Taxing *link a* consequently generates a larger surplus than taxing *link b*, for identical target utilities.

The social welfare gains from optimal transport prices in the example are limited, because congestion is small. When households 1 receive all the congestion tax revenues, welfare increases by 0.2%, and this gain linearly increases to 0.4% when households 2 receive all the revenues. Further impacts of restricting lump sum transfers according to an exogenous redistribution rule are illustrated in figures 4.1 to 4.3. The horizontal axes show the share of congestion tax revenues which goes to households at location 2. It is no surprise that utility

8 This clearly is an overestimate, which is made for computational reasons. Lower expenditure shares would not affect the relative effects of the policies we analyse.

of ‘households 2’ is increasing in the tax revenue share when both links are taxed (figure 4.1). More remarkable is that the utility of ‘households 1’ drops below the reference level, once more than 12% of congestion tax revenues goes to households 2. This indicates that taxes are affected by Mirrlees inequality as well as by marginal congestion costs, as is confirmed in figure 4.3. Taxes are nearly twice as high as marginal external congestion costs on *link a* when households 1 receive all the revenue (in case both links can be taxed, series tA/mecA). The ratio increases as more tax revenues go to households 2, because both the direct tax effects and the income redistribution effects promote the exploitation of Mirrlees inequality. For *link b*, taxes are below marginal external congestion costs except when all tax revenues are given to households 2, in which case they are equal to marginal external congestion costs (in case both links can be taxed, series tB/mecB). When all tax revenues go to households 2, there is no reason to tax below marginal external costs on *link b*, as the efficiency gains from internalisation are not counteracted by transferring income to the high cost households 1. Also, there is no reason for taxing above marginal external costs, as Mirrlees inequality can be exploited fully by the tax on *link a*. Consequently, a tax equal to marginal external costs is obtained on *link b*.

Figure 4.2 shows that the share of households 1 in aggregate social welfare (as given by $N_1V_1/(N_1V_1+N_2V_2)$) decreases strongly as more tax revenues are given to households 2 (in case both links can be taxed, series SW share hh1). The more important message is that the share decreases below the reference level as soon as more than some 12% of congestion tax revenues are redistributed to households 2.

The interaction between partial pricing, tax redistribution assumptions and tax levels is straightforward in this example. Taxing *link a* only, indirectly decreases congestion for households 2, thereby re-enforcing the effects of Mirrlees inequality. The optimal tax is higher than in full network pricing, and it increases above marginal external costs at a quicker rate as the revenue share of households 2 grows (figure 4.3, series tA/mecA(tB=0)). As is shown by figure 4.2, the share of households 1 in social welfare drops compared to full network pricing, and it drops below the reference level as soon as a congestion tax on *link a* is implemented (even if households 1 receive all revenues). When only *link b* can be taxed, the optimal tax is slightly above marginal external costs in case all revenues go to households 1 (figure 4.3). It is also higher than the optimal tax under full network pricing. The desire to use the tax on *link b* to internalise congestion on *link a*, is counteracted by the effect of excessively high taxes on the utility of households 2. When more of the revenue goes to households 2, higher taxes will counteract to a lesser degree the tendency to advantage households 2, so that the tax rises above marginal external costs, at nearly the same rate as under full network pricing. Figure 4.2 illustrates that taxing *link b* only, leads to a higher share of households 1 in social welfare.

Figure 4.2 indicates that partial taxation causes a nearly parallel shift in the welfare share of both households. It is affected by the tax revenue shares to a limited extent only. The effect on link tax restrictions is nearly independent of the effect of tax revenue shares, while both as such have a strong impact on the welfare shares. The independence is not complete however. When all tax revenues go to households 2, taxing *link a* only has nearly the same effect on the welfare distribution as taxing both links.

A further possibility for restricting lump sum transfers is to impose that each household is compensated by the amount of congestion taxes that it pays. This prevents use of the revenues for exploitation of Mirrlees inequality, but tax levels are still affected by the Ramsey-Mirrlees term. Table 4.2 presents the main results for optimal network pricing under this form of redistribution. Despite the fact that the optimal taxes with full network pricing (row 2) are affected by the Ramsey-Mirrlees term, the share of households in social welfare is virtually unchanged. The deviation between taxes and the marginal external costs is rather small, see columns (4) and (5). Partial pricing also has no significant effect on the distribution of aggregate welfare either. Furthermore, the efficiency gains from partial pricing are nearly as high as those of efficient pricing. When the exploitation of transport cost differences is ruled out, partial taxation is an effective instrument in the two link serial network example. The reason is that congestion in the network is limited, so that the efficiency gains from congestion pricing are small.

Table 4.2 Effects of network pricing with exact compensation[a]

	(1) Utility index households 1	(2) Utility index households 2	(3) SW share households 1	(4) tA/mecA	(5) tB/mecB
tA=0, tB=0	1	1	0.703	0	0
tA*, tB*	1.0024	1.0018	0.704	1.083	0.963
tA*, tB=0	1.00015	1.0022	0.703	2.492	0
tA=0, tB*	1.0027	1.0005	0.704	0	1.406

[a] The * sign indicates that a tax is optimised.

In summary, the example illustrates that the interaction between Mirrlees inequality and network externalities leads to deviations of optimal taxes from marginal external costs. This is the case with full and with partial network pricing, and the deviations depend on the specific tax redistribution assumptions. With full network pricing, links which are predominantly used by households with low transport costs tend to be priced below marginal social cost, and those mostly used by households with high transport costs are priced above social cost. This tendency is re-inforced as the tax redistribution mechanism favours low cost households. It is clear that deviations between taxes and marginal external congestion costs under partial network pricing are not only explained by network interaction, but that account must be taken of the Ramsey-Mirrlees effect as well. In the example, this is shown by the faster increase of the tax on *link a* above marginal external congestion costs when only *link a* can be taxed, as compared to full network pricing. Finally, the impact of the Ramsey-Mirrlees term is neutralised to a large extent (but not completely) when each household is compensated by exactly the amount of congestion taxes it pays. We will use this assumption in the next example, enabling ourselves to focus on the importance of link tax restrictions.

Figure 4.1 **Household utility index with full network pricing**

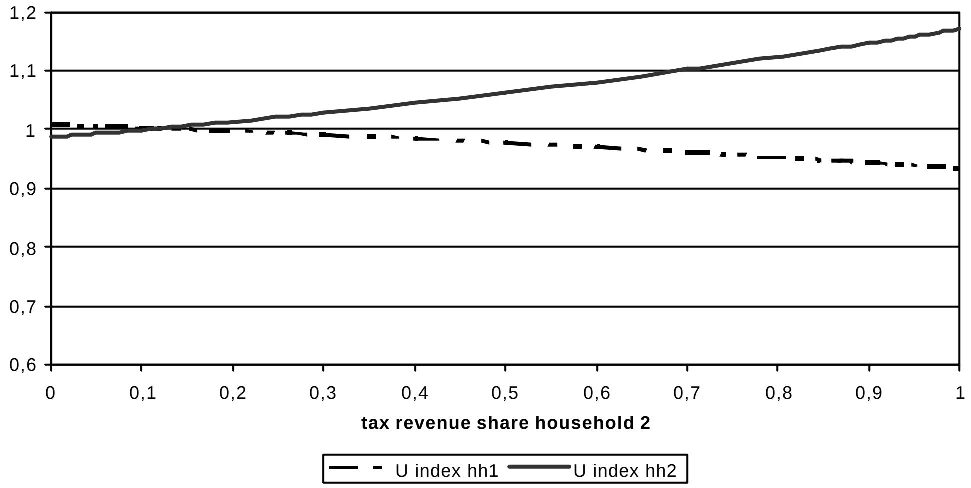


Figure 4.2 **Distribution of social welfare with full and partial network pricing** [9]

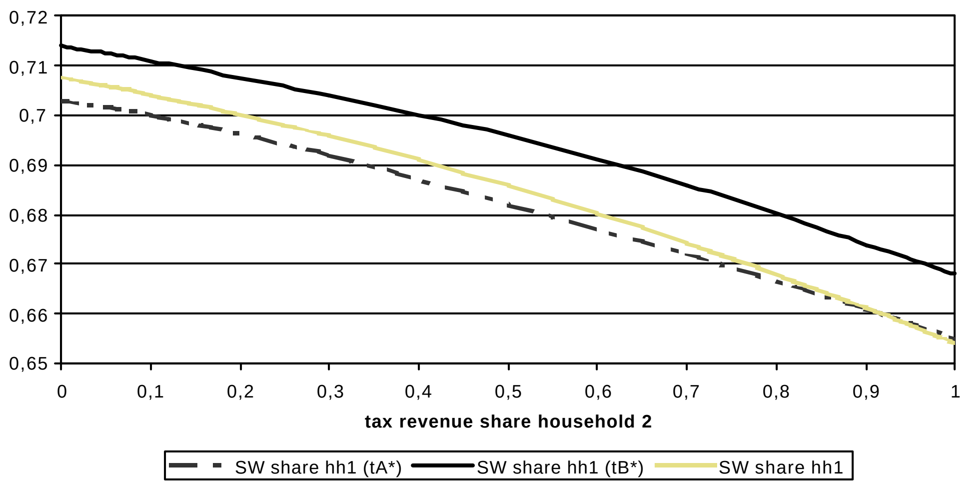
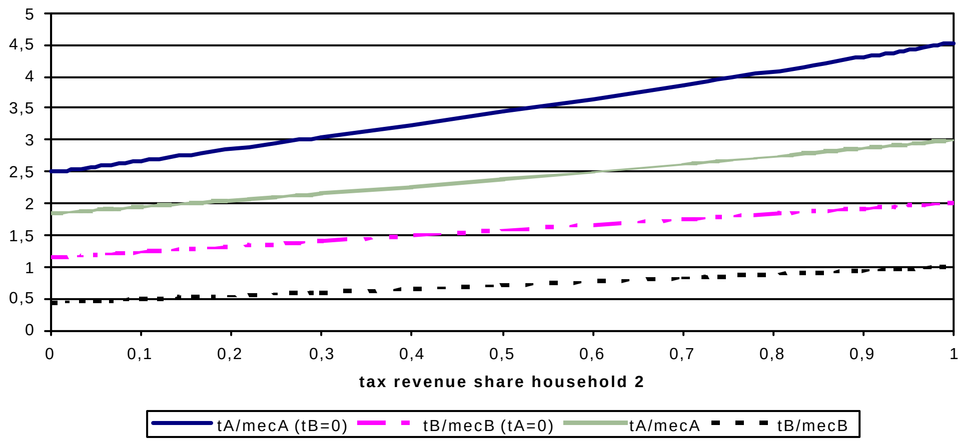


Figure 4.3 **Taxes / marginal external congestion costs, with full and partial network pricing**



4.3 Example 2: network interactions in a 3 link network

Example 2 highlights the impact of link tax restrictions on the effectiveness of network congestion pricing. The network, see figure 4.4, consists of three directed links (*a,b,c*) and three nodes (1,2,3). It is a stylised representation of high capacity roads which are used for through traffic in Namur, during the morning peak. *Links a and b* are west of the city centre, *link c* is to the east. The idea is to test partial pricing schemes, that is: taxes on (combinations of) *links a, b and c*. The relative efficiency of the different combinations is compared to the full optimum solution in which all links are priced. As mentioned, it will be assumed that households are compensated for the congestion taxes that they incur (conform equation (2.7)). We start by analysing a central scenario in section 4.3.1, comparing partial pricing schemes and optimal parking charges. Section 4.3.2 takes the same preference structure and analyses the impact of variations in cost characteristics. Section 4.3.3 looks at variations in the preference structure, for the cost structure on the network as defined in the central scenario.

4.3.1 The central scenario

Table 4.3 gives the central scenario reference demands from the representative consumer at each trip origin, for the three origin-destination pairs and the cost function parameters for the three links. For trips from node 1 to node 3, *paths ab and c* can be chosen. In the reference equilibrium *link c* is under-used from the social point of view (see below). With respect to the structure of preferences, we assume that there is no substitution between destinations 2 and 3 for households travelling through node 1. The paths from node 1 to node 3 (*ab and c*) are perceived as perfect substitutes. Conform to the Wardropian network equilibrium condition (see equation (2.5)), the sum of link travel costs on *a and b* is equal to the cost on *link c*.

We first discuss the effects of partial link pricing schemes. Next, the effects of optimal parking taxes at one or both destinations are presented.

The main results from the partial pricing combinations are summarised in table 4.4. The maximal welfare gain is obtained by taxing all network links (*a, b, c*) at approximately marginal external cost. The impact of Mirrlees inequality on optimal prices is small because of the assumption of exact compensation, and because of the presence of alternative paths for traffic going from node 1 to node 3 (network interaction). The optimal prices reduce demand by approximately 6% for all origin-destination pairs, and increase the share of *link c* for trips from node 1 to node 3 from 14.5% to 18%. Optimal pricing on all links achieves efficiency in terms of demand levels as well as in terms of network assignment (distribution of flows over the network), if combined with exact compensation of households through the tax redistribution mechanism. The results suggest that the contribution of demand reduction to the welfare improvement is larger than that of improving the assignment, however. This finding is confirmed within other (larger) network configurations, which are not reported here.

---

[9] The * sign indicates that a tax is optimised.

Figure 4.4 Topology for network example 2

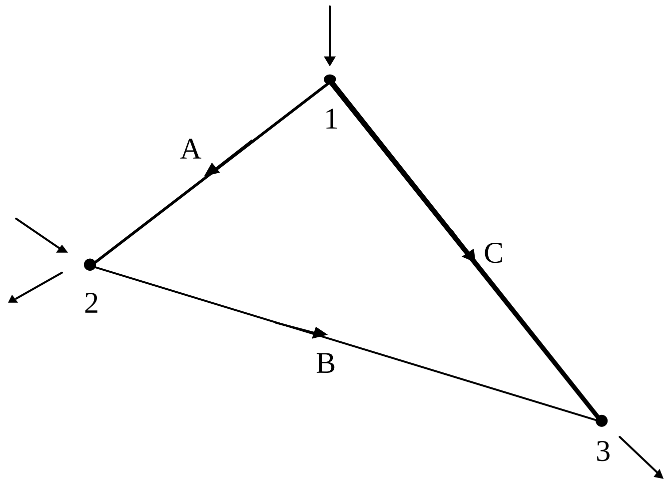


Table 4.3 Cost and demand characteristics of the three link network (reference equilibrium)

Aggregate demand (trips)

1 – 2	993
1 – 3	1590
2 – 3	1089

Costs (Eurocent) and flow (vehicles)

	Intercept	Slope	Flow	Link travel cost
Link a	57	0.004	2353	66.4
Link b	83	0.007	2449	100.1
Link c	150	0.072	230	166.5

Table 4.4 The effects of partial pricing in the three link network

	Reference	Optimal tax on link(s)						
		a, b, c	a	b	c	a, b	a, c	b, c
(1) percentage of maximal welfare gain								
	0	100	46.6	53.9	0	54.4	72.8	92.1
(2) percentage demand change per origin-destination pair								
1,2	0	-5.6	-2.5	-1.5	0	-1.8	-5.8	-4.0
1,3	0	-5.6	-2.5	-1.5	0	-1.8	-5.7	-4.2
2,3	0	-6.3	0.4	-3.6	0	-2.8	0.4	-8.7
(3) ratio of taxes over marginal external costs per link								
A	0	1.005	0.93	0	0	0.21	2.1	0
B	0	0.998	0	0.55	0	0.45	0	1.4
C	0	0.999	0	0	0	0	0.7	0.9
(4) distribution of demand for 1,3 over paths								
AB	85.5	82.0	79.2	78.9	85.5	78.7	81.8	82.2
C	14.5	18.0	20.8	21.1	14.5	21.3	18.2	17.8

The share of the maximal welfare improvement which is achieved by partial pricing schemes ranges from 0% (*link c*) to 92% (*links b and c*). A tax on *link c* will shift traffic going from node 1 to node 3, to *path ab*, which is not desirable in terms of assignment. This negative effect can not be compensated strongly enough by a demand reduction (as paths are perfect substitutes), so that it is optimal to set the tax on *link c* equal to zero if no other links can be taxed. On the contrary, taxes on *links a* and *b* simultaneously improve the assignment and affect demand levels. They permit substantial positive welfare gains. The tax on *link b* performs better than the tax on *link a* because it affects traffic originating from both nodes 1 and 2, on the link suffering from the strongest congestion problem (i.e. *link b*). Note that taxes for one link pricing schemes are below marginal external costs and that the demand reductions are both smaller and more diverse than for complete network pricing. There even is a small demand increase from node 2 to node 3 when only *link a* is taxed. Furthermore, the network assignment is shifted towards over-usage of *link c*, instead of under-usage in the reference equilibrium (always from the socially optimal point of view). Referring to equation (2.14), the example suggests that the second term on the right hand side is negative and large, such that taxes are substantially lower than marginal external congestion costs. However, in larger networks there is a larger probability that taxes rise above marginal external costs, in order to achieve sufficient demand reductions (cfr Van Dender, 2001d).

With respect to two-link pricing schemes, it is interesting to note that the welfare gain of a combined tax on the serial *links a and b* is much smaller than the sum of the welfare gains of a single tax on *links a* and *b*. The reverse holds for parallel links: the welfare gain of a combined tax on *links a and c* (*b and c*) is much larger than the sum of the gains from single taxes on *a* and *c* (*b* and *c*). Taxing a sequence of links does not outperform taxing one link in the sequence by a lot (compare the gains from scheme *ab* and scheme *b*), when the network interactions for both types of taxes are similar. On the contrary, taxation of parallel links allows to –imperfectly- control assignment choices and demand levels, allowing much larger welfare gains. For the same reasons, it is not surprising that taxes may rise above external costs in two-link pricing schemes, at least in a three link example. For schemes *ac* (*bc*), the tax on *link a* (*b*) is used to –imperfectly- control traffic on the sequence *ab*.

While link taxes are generally thought of as being expensive to implement, parking charges at trip destinations may be a cheaper policy instrument. The example allows to compare the welfare effects of parking taxes at destination 2 and/or destination 3, to the effects of link taxation. Note that *optimal* parking charges, given policy constraints, have been computed. Also, in the present context parking charges are used to internalise congestion externalities, but not to correct for inefficiencies related to parking as such.

Intuitively it is clear that the effectiveness of parking charges depends on the contribution of excess transport demand levels and of inefficient assignment to the global transport inefficiency problem, as parking charges can not correct assignment inefficiencies but they can be used to restrain demand. Since the problem of excessive demand in this example (as well as in other network configurations) is more important than the assignment inefficiencies, it is not surprising that an optimal parking charge at both trip destinations realises only 14% less welfare gains than optimal congestion taxes on all network links. An optimal parking charge at destination 2 only, produces 58% of the maximal welfare gain (67% of the gain of optimally charging for parking at both destinations). Similarly, an optimal parking charge at destination 3 only, produces 79% of the maximal welfare gain (92% of the gain of optimally charging for parking at both destinations). In other words, optimal parking charges outperform all one-link pricing schemes, and their performance is comparable to that of two-link pricing schemes.[10] The impact of parking charges on trip demand levels and on assignment is summarised in table 4.5.

Table 4.5 Effect of optimal parking charges on trip demand levels and on path shares, In comparison with first best link taxes

	Reference	First best link taxes	Optimal parking charge at		
			All destinations	Destination 2	Destination 3
(1) Percentage demand change with respect to reference equilibrium					
1,2	0	-5.6	-5.6	-5.8	-4.1
1,3	0	-5.6	-5.5	-5.3	-4.2
2,3	0	-6.3	-6.6	0.2	-8.6
(2) Distribution of demand for 1,3 over paths					
AB	85.5	82.0	86.1	85.6	86.1
C	14.5	18.0	13.9	14.4	13.9

An optimal parking charge at both destinations achieves virtually the same demand reductions as optimal congestion taxes on all links. As could be expected, demand reductions are lower for destinations where no parking charge is implemented. Demand for destination 3 increases slightly when only destination 2 is subject to the charge, which is the consequence of reduced congestion on *path ab*. The network assignment is not improved by parking charges, as there is a switch to the over-used path (*ab*) in all parking scenarios. Because of the demand reduction, the costs of the assignment inefficiency are reduced however. In sum, the decreased performance of parking taxes (minus 14% compared to optimal link taxes) gives an idea of the contribution of assignment inefficiencies to the overall transport inefficiences.

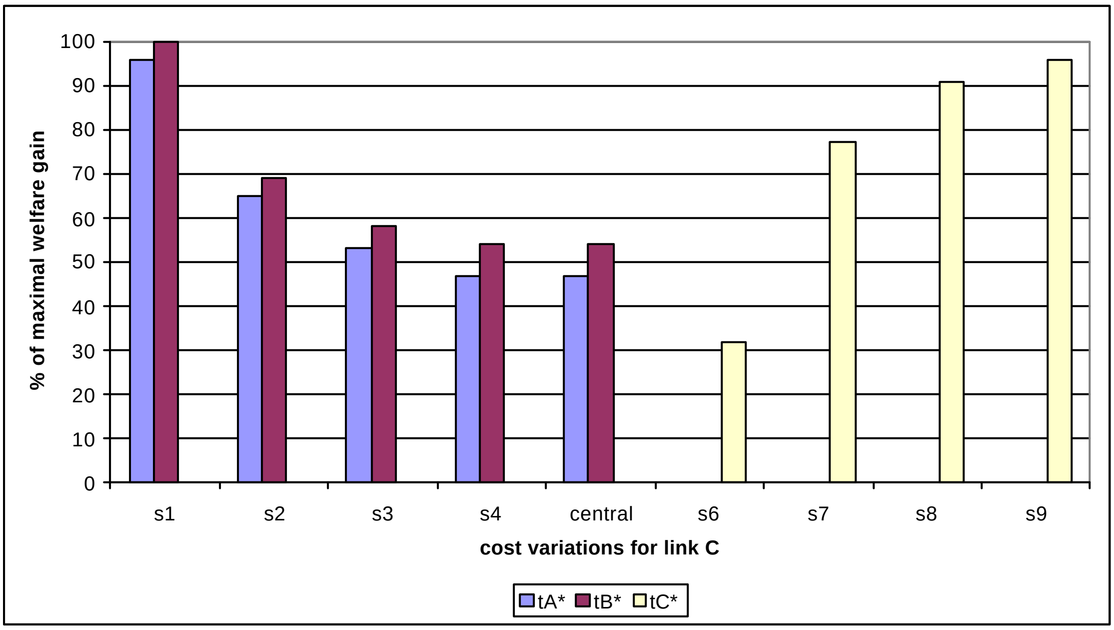
4.3.2 Cost variations

The impact of cost characteristics is analysed for the preference structure of the central scenario, i.e. there are representative households at two locations (nodes 1 and 2), who travel to two destinations (nodes 2 and 3), where substitutability between destinations is excluded. For trips from node 1 to node 3, two paths are available (*path ab* and *path c*). We consider the effect of altering the cost function for *link c* in two ways, keeping the cost functions for *links a* and *b* constant. First, the effect of increasing the slope of *link c* is assessed (case 1). As the slope increases, the link becomes more congestible and hence more expensive from the private point of view, as well as in terms of marginal external congestion costs. Second, the intercept of *link c* is increased (case 2). This can be interpreted as increasing the length of the link, which increases the private costs of using *link c* without directly affecting marginal external congestion costs. The impact of these experiments on the performance of one link pricing schemes is summarised in figure 4.5. Scenarios to the left of the central scenario concern case 1 (changing slope of *link c*), where the slope is lowest for scenario s1. The

[10] It should be noted that the relative performance of parking charges will decrease in a network with more variation in origins, destinations and trip distances. Furthermore, parking charges do not deter through traffic (e.g. Glazer and Niskanen, 1992).

scenarios to the right concern case 2 (changing length of *link c*), where the length is the smallest for s9.[11]

Figure 4.5 Share of maximal welfare gain from optimal one-link pricing schemes under cost variations



As can be observed, the performance of pricing *link a* or *link b* increases as the slope of *link c* decreases (scenarios 1 to 4). When the slope of *link c* is very low (scenario 1), a tax on *link b* is sufficient to generate the maximal possible welfare gain. It is not possible to realise welfare gains by taxing *link c*, unless its length decreases (scenarios 6 to 9). In those cases where the optimal tax on *link c* is positive, the optimal tax on *links a and b* is zero. These results can not be explained by considering the cost structure of the network alone. Account must be taken of the social valuation of the network assignment in case no taxes are introduced. In other words, the degree of inefficiency of users decisions –which depends on the network cost structure- should be considered. This is discussed in the next paragraphs.

Figures 4.6 and 4.7 show the index of relative efficiency of partial pricing schemes, as well as of ‘network heterogeneity’. The latter concept is based on two standard solutions of the network assignment problem in transportation science, the user equilibrium and the system optimum. The user equilibrium follows from uncoordinated cost minimisation by individual users. The resulting network assignment is consistent with the Wardropian version of the consumer equilibrium conditions described in equation (2.5), if the demand levels from the consumer optimisation are used. The system optimum minimises costs from the social point of view, for given demand levels. It corresponds to a command and control assignment procedure. The amount of network heterogeneity indicates to which extent the user equilibrium assignment deviates from the system optimum assignment, for the reference demand levels. Since demand levels are fixed at the reference levels, and since assignment

[11] Each scenario halves the slope (scenario 4 to scenario 1) of link C, or halves the length of link C (scenario 6 to 9) with respect to the previous case.

choices are limited to the choice between *paths ab and c* for trips from node 1 to node 3, the amount of network heterogeneity in our example can be described by the deviation between the user equilibrium share of traffic from node 1 to node 3 on *link c.* Figures 4.6 and 4.7 show an index of this deviation.

The figures indicate that heterogeneity drops as the cost of *link c* is increased, either because of an increase in the slope or the intercept. In the case of figure 4.6, the user equilibrium flow on *link c* is too low as compared to the system optimum, and the difference decreases as the slope of *link c* rises. For figure 4.7, the user equilibrium flow on *link c* is too high, from the social point of view for low link lengths. Hence the direction of the assignment inefficiency is reversed between both figures.

In case 1 (figure 4.6) taxing *links a or b* is becomes less efficient when the slope of *link c* rises. This is intuitively straightforward: as *link c* becomes more congestible, it becomes less attractive to tax *links a* and *b*, because this generates excessive congestion on *link c*. If *link c* is not strongly congestible, taxing *link a* is a good way to decrease demand from origin 1, and taxing *link b* tackles demand from both origins (hence the difference in relative efficiency between taxing *link a* and *link b*). Using these instruments generates a limited amount of inefficient assignment only. The efficiency cost of inefficient assignment rises as the slope of *c* rises, however. At first sight, it is remarkable that taxing *link c* only generates no efficiency gains in this network, even if its slope is high. The reason is that the high slope as such leads to a user equilibrium in which *link c* is under-used: the user equilibrium is concordant with the system optimum to a substantial degree. A non-negative tax can not correct for the remaining inefficiency.

In case 2 (figure 4.7) only taxing *link c* generates efficiency gains if only one link in the network can be taxed. This again follows from the direction of network heterogeneity (under-usage of *path ab* from the social point of view). Taxing *link c* only becomes less effective as the length of *c* increases, because the increase in user costs caused by the length increase brings the user equilibrium closer the social optimum, irrespective of whether a corrective tax is used or not. This also explains while the effectiveness drops to zero when the direction of heterogeneity switches sign.

It is interesting that the relative efficiency of partial pricing schemes depends on network heterogeneity, which in turn depends on the cost structure of the network. Partial pricing schemes become less effective as networks become less heterogenous (more homogenous). The reason is that in homogeneous networks, path switching –when possible- is not strongly discouraged through cost function differences between paths. This makes partial pricing schemes a bad instrument for aggregate demand reduction. Unfortunately, demand reduction is the main objective of network pricing in a homogenous network, as assignment becomes more efficient as homogeneity increases.[12] Partial pricing therefore is useful only in strongly heterogenous networks. Of course, full network pricing does not suffer from this problem, as it simultaneously affects trip demand levels and assignment inefficiencies.

[12] This suggests that uniform pricing schemes become relatively attractive in homogenous networks.

The present discussion is relevant to the policy problem in which a (number of) links on which a tax will be introduced, must be selected from a network. In a homogenous network, the choice of a link does not matter much. Taxing any link will generate only modest welfare gains anyway. The discussion of parking charges in section 4.3.1 suggests that other instruments than link taxes should be considered to obtain the desired demand effects. In a heterogenous network, the choice of a link becomes very important. Those links that are over-used from the social point of view should be taxed. These are not necessarily the links with the highest congestion, however (see figure 4.6). It is necessary to compare the fixed demand user equilibrium and system optimum. This however is a relatively simple computation, available from most standard network equilibrium software.

Figure 4.6 **Index of relative efficiency of one-link-pricing for increasing congestibility of link C heterogeneity for central scenario=1, and heterogeneity is decreasing in slope of link C)**

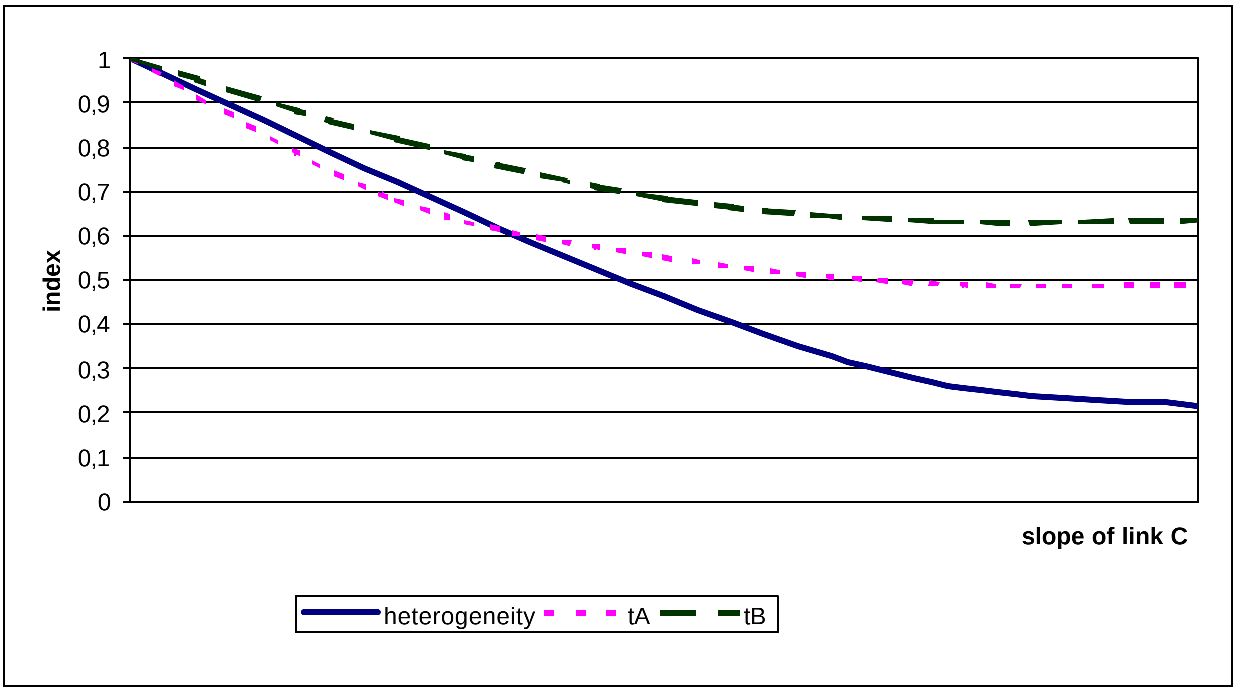
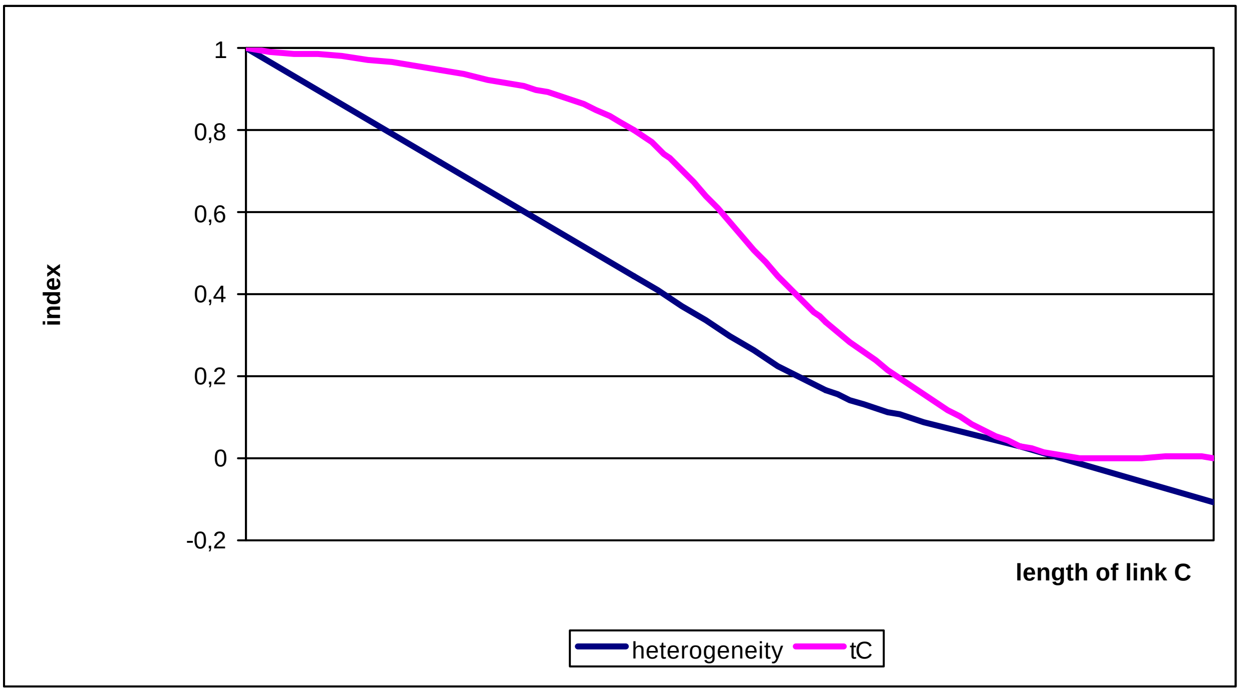


Figure 4.7 **Index of relative efficiency of one-link-pricing for increasing length of link C (heterogeneity for central scenario=1, and heterogeneity is decreasing in length of link C)**



4.3.3 Preference variations[13]

This section presents results concerning the impact of variations in the preference structure on the effects of partial network pricing schemes, while retaining the cost structure of the central scenario. The scenarios are summarised in table 4.3. First, a scenario is considered in which substitutability between destinations (for households travelling through node 1) is positive, in contrast to zero as in the central scenario ('substitute destinations'). Destinations may be substitutes in the long run, independent of trip motives. In the short run, subsitutability can be substantial for shopping destinations. Second, the substitutability between *paths ab* and *c* (for households travelling from node 1 to node 3) is decreased ('imperfect substitute paths'). This may be interpreted as paths representing different transport modes. The congestion functions are left unchanged however.

Table 4.4 Effect of preference variations on the effectiveness of partial pricing systems (% of maximal welfare gain)

	Optimal tax on links:						
	a, b, c	a	b	c	a,b	a,c	b,c
Central scenario	100	46.6	53.9	0	54.4	72.8	92.1
Substitute destinations	100	45.0	53.6	0	54.4	67.5	89.4
Imperfect substitute paths	100	46.6	53.9	0	54.4	72.8	92.1

Table 4.5 Effect of preference variations on the demand impacts of partial pricing systems (% demand change with respect to reference equilibrium)

	Optimal tax on links:						
	a, b, c	a	b	c	a,b	a,c	b,c
Central scenario							
1,2	-5.6	-2.5	-1.5	0	-1.8	-5.8	-4.0
1,3	-5.6	-2.5	-1.5	0	-1.8	-5.7	-4.2
2,3	-6.3	0.4	-3.6	0	-2.8	0.4	-8.7
Substitute destinations							
1,2	-5.5	-3.4	-0.7	0	-1.5	-7.2	-2.1
1,3	-5.7	-2.1	-1.7	0	-1.9	-4.8	-4.4
2,3	-6.3	0.4	-3.5	0	-2.7	0.4	-8.4
Imperfect substitute paths							
1,2	-5.6	-2.5	-1.5	0	-1.8	-5.8	-4.0
1,3	-5.6	-2.5	-1.5	0	-1.8	-5.7	-4.2
2,3	-6.3	0.4	-3.6	0	-2.8	0.4	-8.7

The absolute welfare gains in first-best are equal in all three scenarios. As can be read from table 4.3, the effect of allowing substitution between destinations on the performance of partial pricing schemes is negative, but limited. Table 4.5 indicates that, when destinations

[13] Other scenarios than the ones discussed here have been tested, but are left out for reasons of brevity. One of these scenarios suggests that consideration of different trip motives (leisure related transport and commuting) may be important: congestion for commuters can cheaply be decreased by a tax which is sufficiently high to keep leisure related trips off the network during the peak period. This scenario merits more detailed analysis (cfr Parry and Bento, 1999, and Van Dender, 2001c for a non-spatial analysis of the interaction).

are substitutes, demand will react in order to avoid the link taxes. E.g., when *link b* is taxed the demand decrease for destination 2 (for travellers going through node 1) is much smaller in the 'substitute destinations' scenario than in the central scenario, while the demand decrease for destination 3 is larger. This can be interpreted as saying that in a long run perspective –in which locations of trip destinations become endogenous- partial pricing schemes have undesirable side-effects in terms of locational patterns. Transport and trip destinations will shift to untaxed parts of the network. The first-best pricing system does not suffer from this problem.

Decreased substitutability between paths has no impact on the effectiveness of partial systems, nor on the demand reactions. This is not a general result, as other cost configurations show that decreased substitutability between paths tends to increase the welfare potential of partial pricing schemes. The reason for this is that limited substitution between paths allows the social planner to reduce demand for a given path (mode) without large re-assignment effects to other paths (modes). The fact that this phenomenon does not appear in this network example is due to the cost structure, which is such that users' assignment decisions are driven by the relative congestion conditions on different paths (modes), and not by their preference over different paths (modes).

5. Concluding comments

We have presented optimal link tax rules for a general static transport network used by households whose residential location is fixed. First, the impact of the impossibility to tax all network links on the remaining link taxes was discussed, for an economy in which individualised lump sum transfers are available. Both the theoretical analysis and the empirical illustrations indicate that in such a case link taxes will deviate strongly from marginal external congestion costs on the link, because of network interactions. In most circumstances taxes will be much lower than marginal external congestion costs.

Second, when optimal individualised lump sum transfers are not possible but congestion tax revenues are redistributed to households according to given shares, the optimal link taxes are influenced by Mirrlees inequality. This concept refers to the tendency of the optimal tax structure to favour households with relatively lower transport costs. It was shown, theoretically and by means of an illustration, that optimal link taxes will deviate from marginal external congestion costs, even in the absence of link pricing constraints.

The analysis shows that the welfare potential of partial pricing systems strongly depends on the characteristics of the no-tax network assignment, and on the particular links which are taxed. In general, partial pricing systems are less performant when the network is less heterogenous. Heterogeneity here refers to the deviation between the user equilibrium and the system optimum. This is so because partial systems have important negative effects on the assignment in a more homogenous network, and their ability to reduce demand is limited. A further suggestion is that the contribution of improved network assignment to transport efficiency is smaller than that of restraining demand. The fundamental reason is that the user assignment and the cost minimising assignment are similar to a substantial degree, certainly in homogenous networks. When the network is homogenous, alternative instruments (e.g. parking charges) should be considered instead of partial link pricing schemes.

The model can be used for alternative applications. One of these is the economic assessment of link capacity changes, most notably capacity reductions. Capacity reductions are politically attractive. The assessment of such policies requires a network approach. A second, obvious and important application of the present analysis concerns the problem of link selection. The analysis has concentrated on the description of optimal link taxes in case not all links can be taxed. Alternatively, one could ask which links should be selected for taxation, given that only a limited number of links can be taxed (cfr Verhoef, 2000).

Finally, we mention two caveats. First, transport has been modelled as a direct argument of utility instead of as a derived demand. While this is standard practice, it involves some limitations. The most important one is that the complementarity between labour supply and commuting transport is not made explicit. This also implies that interactions between labour tax distortions and transport taxes are abstracted from. Nevertheless, the same basic interactions as in the present analysis would show up in a model taking the complementarity into account. Second, the assumption of fixed location rules out compensating reactions to transport cost changes in the land market. While this clearly is a simplification, it will again be the case that the same type of interactions appear in a model with a land market.

Appendix Construction of the network examples

The data are for the examples in section 4.2 and 4.3 obtained from a network model (Cornélis and Van Dender; 2001)) for the city of Namur (Belgium) and from the TRENEN model (Proost and Van Dender, 2000). Namur is a regional centre with ca. 100,000 inhabitants, located 50km south-east of Brussels. During the morning peak (1.5 hour length), some 26,000 car trips make use of the Namur road network (data for 1990). 51% of these are through trips, of which the origin and the final destination are outside the Namur area. In the examples, only those through trips are considered which enter the Namur area from the north and the west entry points and which exit the area at the south-east exit points. These trips represent 31% of all through traffic, or 16% of total traffic during the morning peak. The implicit assumption is that demand and route choice for all other trips is fixed. The parameters of the link time cost functions are adapted, so that capacity roughly reflects the fixed demands for the other trips. Furthermore, linear approximations to the link cost functions, at the reference link loads, are used. Monetary cost components other than tolls are taken to be independent of congestion. They are fixed at 0.22 Euro per kilometre, so that on the per trip level they depend on route choice. Time units are converted to time costs by using a constant value of time of 8 Euro/hour. Prices are in 1990 terms.

Preferences are represented by nested CES utility functions, which are calibrated using information on prices and quantities in the reference equilibrium and on elasticities of substitution. With respect to the reference expenditure shares, we assume that morning peak trips, evening peak trips and other trips each stand for one third of the daily transport distance. A morning peak through trip is taken to be on average 10km long, of which 3km use the Namur network. The remaining 7km are assumed to take place under free flow conditions, which is a reasonable approximation as congestion is concentrated in urban areas. Consequently, the part of the morning peak trip on the Namur network represents roughly one tenth of the daily transport distance. Its share in daily transport costs is higher however, because not all other transport takes place under congested conditions. Assuming that the morning and the evening peak are symmetrical, that other trips benefit from free flow conditions, and that congestion increases the time cost by a factor of 1.38 on average[14], the time costs of the morning peak on the Namur area stand for 13% of daily transport time costs. The share of transport time costs in the daily time budget for consumption (8 hours) is (1.38*(6km/60kmh)+(24km/60kmh))/8h = 12.5%. The share of morning peak travel time on the Namur network in the daily time budget is 1.63%. For the money budget share, we assume that daily transport expenditures represent 15% of average daily expenditures, i.e. the same share as in average yearly expenditures. The morning peak expenditures on the Namur network (i.e. ca 0.66 Euro) equal 1/10 of that share, or 1.5% of total expenditures.

When the value of marginal time savings in transport equals the marginal value of time in other consumption activities, the daily time budget for consumption is valued at 64 Euro. The daily money budget is 0.66Euro/0.015 = 44Euro. The generalised daily budget is 108 Euro, of which the share of transport during the morning peak on the Namur network is 1.58%. It is clear that considering such a small share of total expenditures strongly limits the total welfare

[14] Figure derived from the network example used in section 4.3

potential to be expected from pricing reforms. As our focus is on the relative gains from different instruments, the total gain is of less importance than the choice of a small but realistic example.

Bijlage 2

Construction of a simulation model for Namur

1. Introduction

The first task undertaken by the GRT was to add some useful features to our ATES software. Let us first recall the purpose of ATES. This software is intended to allow the assignment of traffic flows on a road network. Several algorithms are implemented and could be used during the assignment process : all or nothing assignment (with or without link capacities constraints), user equilibrium, probit assignment, stochastic user equilibrium or system optimum. Before this ITEM project, ATES only dealt with fixed demand of private vehicles. So, according to the ITEM objectives, we developed some new features for ATES:

- taking into account an elastic demand
- assigning also public transport vehicles
- introducing more flexibility in the costs

Concerning the elastic demand, it is clear, from an economic point of view, that the trips demand will be function of the trip prices. As one of the basic hypothesis of ATES was that the demand matrix was a priori fixed and did not depend on the travel price, we need to go beyond this hypothesis if we want a more coherent modelling of transport economics. Therefore, we modify the ATES model adding a new assignment algorithm where the demand assigned on the network is a function of the trips costs and thus of the costs associated to each link. This means that, as the congestion level has an impact on the travel time, it will also affect the demand amount. In this development, we use an algorithm described in Bell and Iida, "Transportation Network Analysis" (Wiley, 1997). This implements a deterministic user equilibrium assignment with an elastic demand. This algorithm is based on the successive average method.

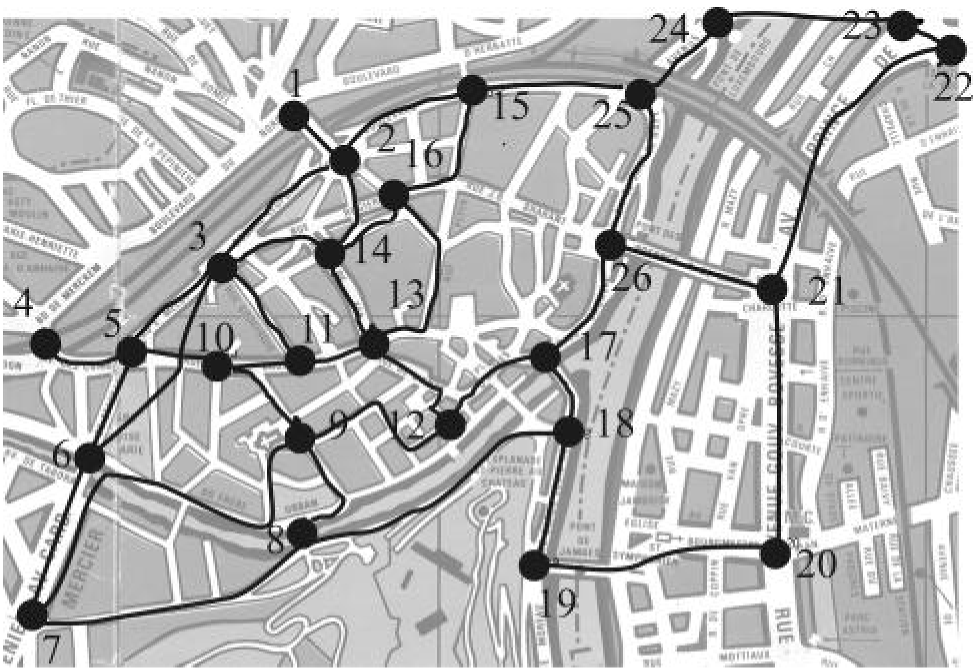
On a ‘classical’ urban road network, the private cars share the infrastructure (or at least most of it if we exclude ‘bus lanes’) with the public transport vehicles (buses). Thus both categories contribute to the road congestion level. So, taking also into account the public transport vehicles into ATES leads to a more realistic modeling. Hence, we updated the software so that the assignment process could also assign buses on the links. The first step is to provide the model with a new dataset concerning the public transport supply. In a data file built according to the same rules as the others (it means that this file is parsed, using the Lex and Yacc facilities, using semantic rules allowing a ‘near current language’ syntax with natural page setting), the user gives, for each public transport line, the followed paths (as a list of links) and the number of vehicles serving this line during the modelled period. Then, these vehicles are taken into account and assigned on there paths, taking so part in the congestion level on these links. A parameter, easily modifiable, was also added to ATES representing the factor between the road space occupied by a bus and the one for a car.

In order to allow a better modelling of the aspects related to the costs of the trips, we modify ATES so that its prices representation was more flexible. For example, the user can add, into the data, a toll for given links. These data are specifically associated to a link and, thus, could be different from link to link (specifying a two-way street as two one-way links allows also to associate different tools (and therefore different costs) to each way). We also isolate the routines where the link travel time and cost are computed. Working this way allows an easier rewriting of these routines if we want to modify the used algorithms. In the current version, the link travel time is a function on the traffic flow on this link (using Davidson type function) and the link cost is the sum of the link travel time and of the link toll.

Finally, for our experiments (see next part), we use ATES for a user equilibrium assignement of a fixed demand and we model the taxes put on the links as tolls on these links.

The second step was to determine a test site for our experiments. We decide to focus on the city center of Namur. Indeed, the GRT dispose of data for this city. To be consistent with the limits of the models developed by the CES, we need to restrict the Namur modelization to a mid-size network without too much nodes and links. So, our detailed description of Namur road network must be redrawn to fill in these constraints. This gives us a network with 26 nodes and 66 links. The figure 1 gives the localisation of nodes and links onto a map of Namur and the table 1 reproduces the data file describing the network which is provided to ATES.

Figure 1 : Namur mid-size network



```
NETWORK
# for Namur

# Nodes list
#1 Pont de Louvain (penetration E411 Champion) 11179
#2 Place Leopold 11171
#3 Gare 20172
#4 Pont d'Heuvy (penetration N4 Belgrade - E42) 11149
#5 Omalius 11165
#6 Pont de la Liberation 12160
#7 Place Wietz (penetration Charleroi) 11162
#8 Pont de l'Eveche 12157
#9 Rue Saintraint, Place St Aubain 12158
#10 FUNDP 20221
#11 Rue de Bruxelles - Rue Godefroid 12153
#12 Place Maurice Servais 20210
#13 4 coins 12152
#14 Hotel de Ville 20176
#15 Cite Administrative 12148
#16 Rue Rogier 12147
#17 Maison de la Culture 12149
#18 Grognon 11177
#19 Pont de Jambes (penetration Dinant N92) 11161
#20 Place de la Wallonie (penetration Dinant rive droite) 11160
#21 Place Josephine-Charlotte 11163
#22 "Sarma" (penetration N4 - Erpent) 11159
#23 Grands Malades (penetration Liege - E411 Loyers) 11153
#24 Saint Camille (penetration E411 Bouge - Chaussee de Hannut) 11152
#25 Pont de Luxembourg 11173
#26 Pont des Ardennes 11175

#links list
#name – origin – direction – destination – length(m) – max. speed (km/h)- capacity
#(veh.) – sensibility (to congestion)
pt_Louvain 1 = 2 150 60 5000 0.1
derriere_CA 2 > 3 340 60 3500 0.2
rue_Borgnet 3 > 2 480 60 3000 0.2
bld_Melot 3 > 5 505 60 3000 0.2
omalius_Gare 5 > 3 414 60 4200 0.2
av_Combattants 5 = 4 328 60 3500 0.2
av_Stassart 6 = 5 242 60 3000 0.2
tunnel_Omalius 6 = 3 836 60 3500 0.2
av_Cardinal_Mercier 6 = 7 431 60 3500 0.2
av_Reine_Astrid 7 = 8 767 60 3000 0.2
pt_Eveche 8 = 9 84 60 4000 0.25
bld_Frere_Orban 7 = 9 582 60 4000 0.25
vers_FUNDP 9 = 10 510 50 3500 0.25
rue_Bxl 5 = 10 370 60 3000 0.2
rue_Bxl_bas 11 = 10 299 60 3000 0.2
rue_Godefroid 11 > 3 370 60 3000 0.2
rue_St_Jacques 13 > 11 200 50 4500 0.2
rue_brasseurs 9 > 12 597 40 2000 0.2
rue_Fer_ht 3 > 14 205 60 3000 0.2
rue_fer_bas 14 > 13 260 60 3000 0.2
rue_Rogier 14 > 16 405 60 2500 0.2
rue_Galiot_et_Nameche 14 = 2 385 60 3000 0.2
rue_Cuvelier_et_Pepin 13 > 16 700 60 2500 0.2
rue_Gnl_Michel 16 > 15 225 60 3000 0.2
bld_Cauchy 15 = 2 370 70 5000 0.2
bld_Cauchy_casernes 25 = 15 405 70 5000 0.2
bld_Albert_1 24 = 25 1500 90 4000 0.1
bld_Smet_Nayer 25 = 26 315 70 6000 0.2
pt_Ardennes 26 = 21 315 60 5000 0.2
bld_Brunnel 26 = 17 360 60 6000 0.2
pt_France 17 = 18 205 60 4000 0.2
rue_bord_eau 18 = 8 756 60 4500 0.2
bld_Huart 18 = 19 344 60 4000 0.2
pt_Jambes_Av_Materne 19 = 20 600 60 3500 0.2
av_Bovesse 20 = 21 600 70 4000 0.2
av_prince_Liege 21 = 22 700 70 6000 0.2
ch_Liege 22 = 23 749 90 5000 0.2
pt_Grands_Malades 23 = 24 360 60 3500 0.2
rue_Marchovelette 17 > 12 200 40 6500 0.3
rue_Ange 12 > 13 502 40 6500 0.3

END
```

Table 1 : Data file for Namur network description

Then, we had to obtain an O/D matrix with the trips demand to be assigned on the network. As we only disposed of desaggregated data, we needed to write a piece of software allowing us to describe the demand between the few nodes used for the Namur mid-size network. The demand was given for the morning peak (between 7 am and 9 am). The obtained matrix is presented in Table 2.

```
MATRIX
#demand for Namur
#during the
#morning peak
 1 2 50
 1 3 359
 1 4 533
 1 5 16
 1 7 491
 1 8 39
 1 9 69
 1 10 352
 1 11 28
 1 12 210
 1 13 122
 1 14 368
 1 15 34
 1 16 201
 1 17 31
 1 18 24
 1 19 113
 1 20 424
 1 21 87
 1 22 229
 1 24 181
 1 25 113
 2 1 3
 2 3 2
 2 4 2
 2 7 1
 2 10 2
 2 12 2
 2 13 1
 2 14 2
 2 20 2
 2 24 1
 3 1 126
 3 2 18
 3 4 129
 3 5 7
 3 7 153
 3 8 14
 3 9 17
 3 10 122
 3 11 6
 3 12 66
 3 13 35
 3 14 181
 3 15 14
 3 16 64
 3 17 10
 3 18 5
 3 19 25
 3 20 111
 3 21 19
 3 22 42
 3 24 51
 3 25 30
 4 1 407
 4 2 32
 4 3 229
 4 5 24
 4 7 502
4 8 39
4 9 69
4 10 310
4 11 33
4 12 203
4 13 102
4 14 315
4 15 25
4 16 161
4 17 35
4 18 21
4 19 133
4 20 378
4 21 59
4 22 167
4 24 132
4 25 88
5 1 15
5 2 1
5 3 12
5 4 18
5 7 14
5 8 1
5 9 3
5 10 12
5 11 2
5 12 8
5 13 4
5 14 13
5 15 1
5 16 5
5 17 1
5 18 1
5 19 3
5 20 13
5 21 2
5 22 4
5 24 4
5 25 3
7 1 482
7 2 30
7 3 324
7 4 553
7 5 9
7 8 52
7 9 74
7 10 427
7 11 33
7 12 273
7 13 128
7 14 382
7 15 27
7 16 199
7 17 31
7 18 25
7 19 298
7 20 487
7 21 89
7 22 215
7 24 164
7 25 111
8 1 9
8 2 1
8 3 6
8 4 13
8 7 16
8 9 3
8 10 9
8 11 1
8 12 5
8 13 3
8 14 7
8 15 1
8 16 5
8 17 1
8 18 1
8 19 4
8 20 12
8 21 2
8 22 4
8 24 3
8 25 3
9 1 16
9 2 1
9 3 7
9 4 14
9 7 24
9 8 4
9 10 14
9 11 4
9 12 12
9 13 7
9 14 8
9 16 8
9 17 2
9 19 3
9 20 16
9 21 4
9 22 5
9 24 4
9 25 5
10 1 17
10 2 1
10 3 9
10 4 15
10 5 1
10 7 23
10 8 2
10 9 4
10 11 5
10 12 10
10 13 5
10 14 13
10 15 1
10 16 9
10 17 1
10 18 1
10 19 4
10 20 18
10 21 3
10 22 4
10 24 5
10 25 5
11 1 4
11 2 1
11 3 7
11 4 6
11 7 6
11 8 1
11 9 1
11 10 3
11 12 3
11 13 2
11 14 5
11 15 1
11 16 3
11 17 1
11 19 2
11 20 5
11 21 1
11 22 2
11 24 1
11 25 2
12 1 22
12 2 2
12 3 15
12 4 32
12 7 32
12 8 4
12 9 6
12 10 21
12 11 3
12 13 13
12 14 19
12 15 3
12 16 11
12 17 4
12 18 3
12 19 8
12 20 26
12 21 5
12 22 14
12 24 10
12 25 8
13 1 8
13 2 1
13 3 5
13 4 13
13 7 13
13 8 2
13 9 3
13 10 8
13 11 1
13 12 8
13 14 8
13 15 2
13 16 4
13 17 2
13 18 2
13 19 3
13 20 9
13 21 2
13 22 5
13 24 3
13 25 4
14 1 38
14 2 5
14 3 24
14 4 43
```

```
14 7 47
14 8 5
14 9 8
14 10 30
14 11 3
14 12 16
14 13 11
14 15 5
14 16 19
14 17 5
14 18 2
14 19 10
14 20 66
14 21 10
14 22 25
14 24 15
14 25 12
15 1 12
15 2 3
15 3 10
15 4 17
15 7 19
15 8 3
15 9 3
15 10 9
15 11 2
15 12 8
15 13 6
15 14 10
15 16 9
15 17 3
15 19 5
15 20 14
15 21 3
15 22 8
15 24 5
15 25 6
16 1 12
16 2 2
16 3 8
16 4 17
16 7 19
16 8 3
16 9 4
16 10 11
16 11 2
16 12 10
16 13 7
16 14 10
16 15 4
16 17 3
16 18 1
16 19 5
16 20 14
16 21 3
16 22 8
16 24 5
16 25 6
17 1 4
17 2 1
17 3 2
17 4 7
17 7 6
17 8 1
17 9 2
17 10 3
17 11 1
17 12 3
17 13 3
17 14 4
17 15 1
17 16 3
17 18 1
17 19 2
17 20 6
17 21 2
17 22 3
17 24 2
17 25 2
18 1 4
18 2 1
18 3 2
18 4 8
18 7 7
18 8 1
18 9 2
18 10 5
18 11 1
18 12 3
18 13 3
18 14 5
18 15 1
18 16 3
18 17 1
18 19 3
18 20 8
18 21 2
18 22 3
18 24 2
18 25 2
19 1 196
19 2 11
19 3 108
19 4 222
19 5 5
19 7 260
19 8 19
19 9 34
19 10 166
19 11 11
19 12 96
19 13 54
19 14 141
19 15 12
19 16 80
19 17 14
19 18 16
19 20 244
19 21 45
19 22 123
19 24 75
19 25 48
20 1 233
20 2 13
20 3 124
20 4 252
20 5 4
20 7 282
20 8 23
20 9 38
20 10 183
20 11 11
20 12 103
20 13 66
20 14 167
20 15 14
20 16 94
20 17 20
20 18 18
20 19 98
20 21 110
20 22 182
20 24 99
20 25 60
21 1 22
21 2 2
21 3 15
21 4 26
21 5 1
21 7 26
21 8 2
21 9 4
21 10 18
21 11 1
21 12 12
21 13 9
21 14 18
21 15 2
21 16 10
21 17 2
21 18 2
21 19 6
21 20 39
21 22 20
21 24 14
21 25 7
22 1 239
22 2 15
22 3 125
22 4 229
22 5 2
22 7 251
22 8 19
22 9 33
22 10 177
22 11 13
22 12 110
22 13 60
22 14 171
22 15 13
22 16 97
22 17 15
22 18 13
22 19 68
22 20 298
22 21 81
22 24 88
22 25 57
23 1 265
23 2 15
23 3 149
23 4 215
23 5 4
23 7 238
23 8 17
23 9 33
23 10 183
23 11 11
23 12 112
23 13 58
23 14 164
23 15 14
23 16 97
23 17 10
23 18 13
23 19 61
23 20 274
23 21 79
23 22 314
23 24 102
23 25 60
24 1 241
24 2 17
24 3 142
24 4 184
24 5 4
24 7 198
24 8 15
24 9 28
24 10 144
24 11 10
24 12 87
24 13 50
24 14 144
24 15 15
24 16 82
24 17 12
24 18 10
24 19 44
24 20 188
24 21 61
24 22 134
24 25 52
25 1 4
25 2 1
25 3 4
25 4 7
25 7 6
25 8 1
25 9 2
25 10 4
25 11 1
25 12 3
25 13 2
25 14 4
25 15 1
25 16 3
25 17 1
25 18 1
25 19 2
25 20 6
25 21 2
25 22 3
25 24 4
END
```

Table 2 : The O/D matrix for the morning peak

Before running ATES, a last step was the ‘calibration’ where, for example, we adjusted the capacity parameter for each link. It was quite difficult to appreciate this factor as each link in the model network is in fact the ‘agregation’ of several streets on the field.

2. Demand structure for the Namur case study

2.1 Utility structure

A utility and social welfare structure is constructed for morning peak, evening peak and offpeak periods. The utility structure is taken to represent a mid-term model: no substitutability between origins (fixed locations of consumers), limited substitutability between destinations (variable across trip motives). Social welfare is the unweighted sum of utilities of representative consumers across locations, where O is the set of origins. The next section describes the proposed structure for the set of origins.

2.2 Origins

In principle a utility tree needs to be defined for every origin defined in the reference demand matrix for the stylised Namur network. Inspection of this matrix shows that demand levels are very low for a number of origins, esp. those in the city centre (which follows from the fact that we use a morning peak demand matrix). This allows to reduce the size of the model, by reducing the number of origins. Origins 6 and 26 are dropped because of zero demand in the reference case (these are pure connector nodes: no trips originate from them or have them as destination). Trips originating from nodes 2, 5 and 25 have been reassigned to other nodes – with high reference demands- on the basis of proximity. In the city centre, 2 clusters of origins are defined (west and east). This aggregation procedure reduces the number of origins, hence the number of representative consumers for whom a utility tree is defined, from 26 to 12.

2.3 Destinations

Utility at each location (origin) is derived from consumption of an aggregate good and transport (trips). The following trip types are distinguished:

- Inward and through trips. In the demand matrix, through trips have an entry-exit node as their destination.[15] This is not the real final trip destination however. We assume through trips account for 50% of the trips which are reported to have such a destination node in the reference demand matrix. The remaining 50% have the entry-exit node as their real destination.

---

[15] The subset of entry-exit nodes is (1,4,6,7,8,19,20,22,23,24).

There is no substitution between through and inward trips.

- Mandatory and non-mandatory trips. School and work trips are assumed to be mandatory, implying that the demand is fixed within the morning peak. Shopping and other trips are non-mandatory. Demand is elastic, meaning that the trip time can be changed or trips can be abandoned entirely.
- Both inward and through trips consist of 60% mandatory trips (divided into 50% work trips and 50% school trips) and 40% non-mandatory trips (of which 50% are shopping trips and the remaining 50% are 'other trips').
- The precise destinations of through trips are not known. Given the large geographical diversity of these trips, it is safe to assume zero substitutability between destinations. We will however allow a limited degree of substitutability between some entry-exit nodes for through trips. This reflects flexibility in route choice for through trips.

  Inward trips for work and for 'other' purposes are further clustered into geographical areas (sets of nodes), between which the degree of substitution is very low. Within each area, zones (smaller sets of nodes) are distinguished. Between these zones, substitutability is low. Within these zones, the degree of substitution is fairly high. It represents the possibility of parking at neighbouring nodes and walking to the real destination node. In the first model version, the walking costs will be neglected.
- Inward school trips can have a limited number of destinations, between which no substitution is possible. (There is one exception: one school has 2 entrances.)

  Inward shopping trips are clustered into 2 zones: Namur centre and Jambes. Within the zone 'Namur centre' substitutability is high.
- Define set O={origins O1-012}. Welfare (objective of the maximisation) is the sum of consumer surplusses in each of these origins. The set of origins is interpreted as the set of representative consumers.
- Define sets for each level of the utility tree.

L1 {X,N}	consumption aggregates
L2 {MP,OP,EP}	time periods
L3 {IN,THRU}	general trip direction MP
L4 {MAN, NMA}	general trip type
L5 {WORK,SCHOOL,SHOP,OTH}	detail trip type
L6THRU {EAST,WEST}	direction through trips of all types
L7THRW {SW,NW}	subdirection through trips west
L8THSW {(19,20),(22,23)}	subdirection through trips south west
L6IMWO {E,SC,NWC}	cfr annex
L7IMWOE {(24,25),(22,23),(20,21),19}	cfr annex
L7IMWOSC {(17,18),(12,13,14)}	cfr annex

L7IMWONWC {W,NC}	cfr annex
L8IMWONWCW {(7,8),(9,10,11)}	cfr annex
L8IMWONWCNC {(4,5),(1,2,3,15,16)}	cfr annex
L6IMS {(3,10),9,11,12,13,14,20,26}	cfr annex
L6INS {CENTRE,JAMBES}	cfr annex
D {D1-D26}	destinations

- Define parameters: shares (α, determined by calibration) and elasticities of substitution (σ, exogenous, chosen in order to produce reasonable price elasticities). For the calibration, exogenous prices and quantities for a reference user equilibrium are required.
- For calibrating the shares, start from lowest level and work upwards, applying the following system of equations for each component (:=a branch of the tree, e.g. (X,N), (MAN,NMA) appears as a component for IN and for THRU trips, etc.)).
- Price indices are computed using CES price index formulas (except for the lowest level, where prices are the exogenous reference equilibrium prices).
- At L1, the expenditures for non-transport consumption need to be determined exogenously. One possibility is to fix the price of other consumption to 1 and to fix the expenditure share to 10-15% of total expenditures.
- Replication and counterfactuals use the fixed shares. Price index equations do not change.

3. Model applications

The applications will emphasise the extra features of the present model in comparison to an elastic demand network model. That is, we will present:

- Effects of pricing reform on social welfare ànd on the distribution of this welfare across consumers.
- An analysis of the contribution of global demand reduction and of more efficient network usage for any given demand level, to the global welfare improvement.

In particular, the following types of applications are possible:

- Simulation of a range of tax levels on a (number of) link(s). It is expected that social welfare will first increase with the tax level and then decrease. This is a way of finding the optimal tax level on a link through simulation.
- Selection of a link to be taxed. For a given tax level (or for a given share of external costs on the link), compute the welfare impact of introducing this tax on each link in the network. This is a pragmatic way of link selection, to be compared to ‘common sense rules for link selection’ and to ‘complicated economic rules for link selection’.
- Selection of a time period to be taxed (all periods, one period,…) or of a transport mode to be taxed (car, bus).

- For each of these applications, various types of revenue use are possible.

  Rule 1: <u>equal redistribution</u>

  Compute tax revenue from network equilibrium at iteration 1

  Distribute this revenue equally over all consumers, by updating the total budget at iteration 2

  Recompute demand

  Rule 2: <u>exact compensation</u>

  Compute tax revenue from network equilibrium at iteration 1

  Determine which consumers use the taxed link at iteration 1, and how much

  Determine paid taxes for all consumers at iteration 1 and redistribute accordingly

  Recompute demand

  Rule 3: <u>subgroups</u>

  Redistribute tax revenue according to zone of trip origin, or trip motive, or trip destination

For all these experiments, the GRT developed, according to the formulas given by CES, a software. The CES algorithm implemented by the GRT can be used for the calibration step and then for the replication. This model uses exogenous demands and prices (provided by ATES, see further to know how). For each customer, this means here each trips origin (thus the 26 nodes of the Namur mid-size network), the data provided are the number of trips, by car and by public transport, to each destination and the prices of these trips (the prices for private vehicles trips are computed in ATES and the prices for public transport trips are taken as the same plus a given fare). We provide these inputs for the three considered periods : morning peak, evening peak and off peak. Then, these trips are shared amount the different trip purposes and for the two types (through and inward) of trips according to the rules given here above. Going up through the utility tree, we compute the shares during the calibration step. It must be noticed that we add a supplementary level to the utility tree given here above: at the end of each of its branch, we add a one level tree representing the sharing of demand between cars and public transport vehicles.

Arriving at the top of the tree, we are able to compute the consumer surplus. All the computed shares are stocked into a file and will be used in further replication steps. The consumer surplus are also kept as they will be used as budget constraint in the replication. For all these computations, linear systems are to be solved during the shares calibrations. Therefore, our routines use the C-Lapack solver.

Before all this, an initial step is needed for obtaining the necessary demand for each ATES run. As we only have a demand matrix for the morning peak, we build the other matrices this way:

- the evening peak demand matrix is the transposed morning peak matrix (each origin becomes destination and vice-versa)
- the off peak demand matrix is the quarter of the sum of the two peaks matrices

This is for the private vehicles demand (i.e. the matrices used for the assignments into ATES). For the public transport demands, we take a priori parts of the car demands. How these parts are computed takes into account if the destination allows through and inward trips or only inward trips (i.e. if the destination is an entry/exit point or not), the trip purpose and a last factor representing the part of modal choice for public transport.

When we have all these inputs, i.e. the supply (given by the road network) and the demand, we can start the computations.

In summary, the followed process is as follows :

- Run three times ATES for each considered period : morning peak, off peak and evening peak
- Keep all the demands and the travel costs (these last ones are output by ATES as the sum of all the link costs on the path from origin to destination)
- With these exogenous inputs, run CES for calibration
- The results produced by this calibration are the shares and the consumer surplus (for each consumer, i.e. each origin)
- Going back from top to branches of the utility tree, we compute, at lowest level, the new demands (according to the used methodology, they must be, during the calibration step, equal to the input demands)
- These demands (more exactly the parts of the demands related to the car trips) are provided to ATES for a new round of three runs.
- The newly output prices and the demands are now used in CES for a replication step. This means that the shares are considered as fixed (we take the values computed during the calibration phase) and that the consumer surplus are budget constraints (i.e., arriving at the top of the utility tree, we compute the highest level demand (x0) and price (p0). The product x0*p0 must be equal to the consumer surplus (CS). If not, x0 is taken as CS/p0 for continuing the computations when going back through the utility tree to lowest level where new demands are produced).
- These last two steps are repeated until convergence. The convergence criterion is computed as no more divergence between two successive computed demand for each origin **and each destination**. Divergence is understood as an absolute difference of more than 1 vehicle and a relative difference greater than 1%.

We have written a small script allowing to automatically launching all these programs according to the process followed in this method. This script is presented in Table 3.

Following this scheme, we obtain a basis scenario describing a «normal» situation where the only costs supported by the customers are the travel times (plus a fare for public transport). Starting from there, we can now study how taxing links can affect the demands and principally the welfare (computed as the sum of all consumers surplus plus the tax revenues).

The effects of taxing links were computed for different atxes and different links. Indeed, we compute the impacts of taxes taken from 0 to 50 with a step of 5 (the units are seconds as the taxes are added to the travel time) on each of the network links. This means 660 runs of the script here above mentioned. The results of all these computations were analyzed by CES (KUL).

From a computer science point of view, we developed different softwares for the purpose of this phase of the research. All these were written in C. We include their sources as annexes of this report.

The first one is called ***make_demand***. Its goal is to produce off-peak and evening peak demand matrices for cars as well as public transport demands for the three considered periods (morning peak, off-peak and evening peak) from a given morning peak O/D matrix (for cars).

The second and main one is called ***ces***. It is intended to implement the utility tree and compute calibration. The main file is ***ces.c***. It first includes ***declar.c*** containing all the variable declarations. Then it assigns values (between 0.0 and 1.5) to all the elasticities (the σ from the calibration equations). The next step is an initialisation process (u.o. for demands and prices). Then data are read : budgets constraints (at first step, «-1» values in the data file indicate that they are not yet any budget constraint and that the consumer surplus must be stored to be used as budget constraints in following iterations), demands (for the three considered periods for cars) , prices (resulting from ATES runs) and public transport flows (also for the three considered periods but shared between the different trip purposes and the two trip categories (through and inward)). The main part of the program is a loop on the consumers (origins). In this loop, three files are included: ***tree.c***, ***treeoff.c*** and ***treeeven.c***. In these files, we go up in the utility tree to compute shares. These files are respectively associated to the branches corresponding to each studied period: morning peak, off-peak and evening peak. In each of these files, the following steps are included : adding a fare to the trip price (beeing the «by car» price) to obtain the public transport price), sharing the car demand between the trip purposes (work, school, shopping, other) and the trip categories (through and inward), including the file where the shares for the «car or public transport» decision are computed (these files are respectively ***carorpt.c***, ***carorptoff.c*** and ***carorpteven.c***), computing the shares for all levels of the utility tree going up to the level where the three branches related to each considered period join. All these shares computations mean solving linear systems; this is achieved through routines called ***x_system*** (where ***x*** is replaced by the number of equations (i.e. ***two***, ***three***, ***four***, ***five***, ***six*** or ***eight***) and stored in the ***routine.c*** file. All these routines (but the ***two_system*** one ,very simple and easily directly computable) uses the ***dgesv_*** solver from the LAPACK package (more exactly from C-LAPACK, the C translation of this package) (the used routines from this package are stored in the ***lapack.c*** file). The shares for the two last level (morning peak, off-peak or evening peak and then travel or other goods) are then computed in this main loop of ces. When we are at the top of the tree, we go back to the branches computing the demands (replication step given, here in the calibration, same demands as original ones). These computations are achieved in the ***replic.c***, ***replicoff.c*** and ***repliceven.c*** files, each of them also including, respectively, ***ptreplic.c***, ***ptreplicoff.c*** or ***ptrepliceven.c*** for the last level when the «final» sharing of demands between cars and public transport are computed. Finally, updated flows are built (for each origin and each destination) and compared with previous ones to determine convergence. The last step writes the output files.

The last program is ***cesnocal***. It follows the same structure as the previous one except that the shares are no more computed but retrieved from data file (produced with previous program, ***ces***, during the calibration phase) and thus considered as fixed. Thus, it is used for going through the utility tree (up and then down) for the replication and counterfactual phase. These differences occur principally in the ***treenocal.c***, ***treenocaloff.c*** and ***treenocaleven.c*** files corresponding respectively to the ***tree.c***, ***treeoff.c*** and ***treeeven.c*** files (as well as in the ***carorptnocal.c***, ***carorptoffnocal.c*** and ***carorptevennocal.c*** corresponding to the ***carorpt.c***, ***carorptoff.c*** and ***carorpteven.c*** files) where the shares are really taken into account.

Annexes

We include here the programs developed by the GRT

1.make_demand

```
/*create off peak and even peaks O/D matrix from morning peak demand*/
/* also create public transport demands*/
/*© Dr. Eric CORNELIS, FUNDP, GRT, 2000*/
/* for Microsoft C
#include <process.h>
*/

#include <math.h>
#include <stdio.h>
#include <malloc.h>

main ()
{

FILE *infile,*fcarmor,*fcaroff,*fcareven,*fptmor,*fptoff,*fpteven;

double demand[27][27],
       off[27][27],
       even[27][27],
  morptiw[27][27],
  morptis[27][27],
  morptio[27][27],
  morptish[27][27],
  morpttw[27][27],
  morptts[27][27],
  morptto[27][27],
  morpttsh[27][27],
  offptiw[27][27],
  offptis[27][27],
  offptio[27][27],
  offptish[27][27],
  offpttw[27][27],
  offptts[27][27],
  offptto[27][27],
  offpttsh[27][27],
  evenptiw[27][27],
  evenptis[27][27],
  evenptio[27][27],
  evenptish[27][27],
  evenpttw[27][27],
  evenptts[27][27],
  evenptto[27][27],
  evenpttsh[27][27];

 int origin, destination;

 int od;

 /*init*/
 for (origin = 1; origin <=26; origin++)
 {
   for (destination =1; destination <=26; destination++)
     {
       demand[origin][destination]=0.0;
     }
 }

 /*read data*/
 if( ( infile = fopen("orig_matrix.dta" , "r" ) ) == NULL )
   {
     printf( "%s", "  \nCould not open file for demand\n" );
   }
 while (fscanf(infile,"%d %d %d \n",&origin,&destination,&od) != EOF)
   {
     printf ("READ origin =%d dest = %d od = %d \n",origin,destination,od);
```

```
      demand[origin][destination] = od;
    }
  fclose(infile);


  /*create evening demand*/
  for (origin = 1; origin <=26; origin++)
  {
    for (destination =1; destination <=26; destination++)
      {
        even[origin][destination]=demand[destination][origin];
        printf ("EVEN origin =%d dest = %d
%f\n",origin,destination,even[origin][destination]);
      }
  }

  /*create off peak demand*/
  for (origin = 1; origin <=26; origin++)
  {
    for (destination =1; destination <=26; destination++)
      {
        off[origin][destination]= ( demand[origin][destination] +
                                 even[origin][destination]    ) *0.25;
        printf ("OFF origin =%d dest = %d %f
\n",origin,destination,off[origin][destination]);
      }
  }

  /*create public transport demands*/
  for (origin = 1; origin <=26; origin++)
  {
    for (destination =1; destination <=26; destination++)
      {
        printf ("PUBLIC TRANSPORT origin =%d dest = %d \n",origin,destination);

        if ((destination == 1) ||
            (destination == 4) ||
            (destination == 7) ||
            (destination == 8) ||
            (destination == 19) ||
            (destination == 20) ||
            (destination == 22) ||
            (destination == 23) ||
            (destination == 24))
          {
            morptiw[origin][destination] = demand[origin][destination]
                                           *0.5*0.3*0.132;
            morptis[origin][destination] = demand[origin][destination]
                                           *0.5*0.3*0.642;
            morptish[origin][destination] = demand[origin][destination]
                                           *0.5*0.2*0.05;
            morptio[origin][destination] = demand[origin][destination]
                                           *0.5*0.2*0.05;
            morpttw[origin][destination] = demand[origin][destination]
                                           *0.5*0.3*0.132;
            morptts[origin][destination] = demand[origin][destination]
                                           *0.5*0.3*0.642;
            morpttsh[origin][destination] = demand[origin][destination]
                                           *0.5*0.2*0.05;
            morptto[origin][destination] = demand[origin][destination]
                                           *0.5*0.2*0.05;
            offptiw[origin][destination] = off[origin][destination]
                                           *0.5*0.3*0.132;
            offptis[origin][destination] = off[origin][destination]
                                           *0.5*0.3*0.642;
            offptish[origin][destination] = off[origin][destination]
                                           *0.5*0.2*0.05;
            offptio[origin][destination] = off[origin][destination]
                                           *0.5*0.2*0.05;
            offpttw[origin][destination] = off[origin][destination]
                                           *0.5*0.3*0.132;
            offptts[origin][destination] = off[origin][destination]
                                           *0.5*0.3*0.642;
            offpttsh[origin][destination] = off[origin][destination]
                                           *0.5*0.2*0.05;
            offptto[origin][destination] = off[origin][destination]
                                           *0.5*0.2*0.05;
```

```
          evenptiw[origin][destination] = even[origin][destination]
                                          *0.5*0.3*0.132;
          evenptis[origin][destination] = even[origin][destination]
                                          *0.5*0.3*0.642;
          evenptish[origin][destination] = even[origin][destination]
                                          *0.5*0.2*0.05;
          evenptio[origin][destination] = even[origin][destination]
                                          *0.5*0.2*0.05;
          evenpttw[origin][destination] = even[origin][destination]
                                          *0.5*0.3*0.132;
          evenptts[origin][destination] = even[origin][destination]
                                          *0.5*0.3*0.642;
          evenpttsh[origin][destination] = even[origin][destination]
                                          *0.5*0.2*0.05;
          evenptto[origin][destination] = even[origin][destination]
                                          *0.5*0.2*0.05;
        }
      else
        {
          morptiw[origin][destination] = demand[origin][destination]
                                          *0.3*0.132;
          morptis[origin][destination] = demand[origin][destination]
                                          *0.3*0.642;
          morptish[origin][destination] = demand[origin][destination]
                                          *0.2*0.05;
          morptio[origin][destination] = demand[origin][destination]
                                          *0.2*0.05;
          morpttw[origin][destination] = 0.0;
          morptts[origin][destination] = 0.0;
          morpttsh[origin][destination] = 0.0;
          morptto[origin][destination] = 0.0;
          offptiw[origin][destination] = off[origin][destination]
                                          *0.3*0.132;
          offptis[origin][destination] = off[origin][destination]
                                          *0.3*0.642;
          offptish[origin][destination] = off[origin][destination]
                                          *0.2*0.05;
          offptio[origin][destination] = off[origin][destination]
                                          *0.2*0.05;
          offpttw[origin][destination] = 0.0;
          offptts[origin][destination] = 0.0;
          offpttsh[origin][destination] = 0.0;
          offptto[origin][destination] = 0.0;
          evenptiw[origin][destination] = even[origin][destination]
                                          *0.3*0.132;
          evenptis[origin][destination] = even[origin][destination]
                                          *0.3*0.642;
          evenptish[origin][destination] = even[origin][destination]
                                          *0.2*0.05;
          evenptio[origin][destination] = even[origin][destination]
                                          *0.2*0.05;
          evenpttw[origin][destination] = 0.0;
          evenptts[origin][destination] = 0.0;
          evenpttsh[origin][destination] = 0.0;
          evenptto[origin][destination] = 0.0;
        }
    }
}

 /*output files headers*/
 if( ( fcarmor = fopen("car_morning.dta" , "w" ) ) == NULL )
   {
     printf( "%s", "  \nCould not open file for morning cars\n" );
   }
 if( ( fcaroff = fopen("car_off.dta" , "w" ) ) == NULL )
   {
     printf( "%s", "  \nCould not open file for off cars\n" );
   }
 if( ( fcareven = fopen("car_even.dta" , "w" ) ) == NULL )
   {
     printf( "%s", "  \nCould not open file for even cars\n" );
   }
 if( ( fptmor = fopen("pt_morning.dta" , "w" ) ) == NULL )
   {
     printf( "%s", "  \nCould not open file for morning public transport\n" );
   }
 if( ( fptoff = fopen("pt_off.dta" , "w" ) ) == NULL )
```

```
    {
      printf( "%s", "  \nCould not open file for off public transport\n" );
    }
  if( ( fpteven = fopen("pt_even.dta" , "w" ) ) == NULL )
    {
      printf( "%s", "  \nCould not open file for even public transport\n" );
    }

  fprintf (fcarmor, "MATRIX\n");
  fprintf (fcaroff, "MATRIX\n");
  fprintf (fcareven, "MATRIX\n");

  fprintf (fcarmor, "#morning cars demand \n");
  fprintf (fcaroff, "#off cars demand \n");
  fprintf (fcareven, "#evening cars demand \n");

 for (origin = 1; origin <=26; origin++)
 {
   for (destination =1; destination <=26; destination++)
     {
       if (origin != destination)
         {
           printf ("WRITE origin =%d dest = %d \n",origin,destination);

           fprintf(fcarmor, "%d %d %f\n", origin, destination,
demand[origin][destination]);
           fprintf(fcaroff, "%d %d %f\n", origin, destination,
off[origin][destination]);
           fprintf(fcareven, "%d %d %f\n", origin, destination,
even[origin][destination]);
           fprintf(fptmor, "%d %d %f %f %f %f %f %f %f %f\n",
                   origin, destination,
morptiw[origin][destination],morptis[origin][destination],

morptio[origin][destination],morptish[origin][destination],morpttw[origin][destination
],

morptts[origin][destination],morptto[origin][destination],morpttsh[origin][destination
]);
           fprintf(fptoff, "%d %d %f %f %f %f %f %f %f %f\n",
                   origin, destination,
offptiw[origin][destination],offptis[origin][destination],

offptio[origin][destination],offptish[origin][destination],offpttw[origin][destination
],

offptts[origin][destination],offptto[origin][destination],offpttsh[origin][destination
]);
           fprintf(fpteven, "%d %d %f %f %f %f %f %f %f %f\n",
                   origin, destination,
evenptiw[origin][destination],evenptis[origin][destination],

evenptio[origin][destination],evenptish[origin][destination],evenpttw[origin][destinat
ion],

evenptts[origin][destination],evenptto[origin][destination],evenpttsh[origin][destinat
ion]);
         }
     }
 }
  /* output files footers*/
  fprintf (fcarmor, "END\n");
  fprintf (fcaroff, "END\n");
  fprintf (fcareven, "END\n");

  fclose (fcarmor);
  fclose (fcaroff);
  fclose (fcareven);
  fclose (fptmor);
  fclose (fptoff);
  fclose (fpteven);

}
```

2.ces

```
/*
  CES nested utility

written by Eric CORNELIS (GRT, FUNDP)
© Dr. Eric CORNELIS, GRT, FUNDP, 2000

*/

/* for Microsoft C
#include <process.h>
*/

#include <math.h>
#include <stdio.h>
#include <malloc.h>

#include "declar.c"

main()

{
  void two_system();
  void three_system();
  void four_system();
  void five_system();
  void six_system();
  void eight_system();

  void printxerror();

  gammabase = 1.55 ;
  gammaishnc = 1.5 ;
  gammaionc = 1.4 ;
  gammaiow = 1.4 ;
  gammaiosc = 1.4 ;
  gammaioe = 1.4 ;
  gammais = 1.3 ;
  gammaiwnc = 1.4 ;
  gammaiww = 1.4 ;
  gammaiwsc = 1.4 ;
  gammaiwe = 1.4 ;
  gammatonw = 0.8 ;
  gammatosw = 0.8 ;
  gammatoe = 0.8 ;
  gammatshnw = 0.8 ;
  gammatshsw = 0.8 ;
  gammatshe = 0.8 ;
  gammatsnw = 0.8 ;
  gammatssw = 0.8 ;
  gammatse = 0.8 ;
  gammatwnw = 0.8 ;
  gammatwsw = 0.8 ;
  gammatwe = 0.8 ;
  gammaish = 1.1;
  gammaiw2 = 1.2 ;
  gammaiw3 = 1.1;
  gammaio2 = 1.2 ;
  gammaio3 = 1.1;
  gammais2 = 0.1 ;
  gammaiw2 = 1.2 ;
  gammaiw3 = 1.1 ;
  gammatosw2 = 0.1 ;
  gammatshsw2 = 0.1 ;
  gammatssw2 = 0.1 ;
  gammatwsw2 = 0.1 ;
  gammaiwnwc = gammaiw3 ;
  gammaionwc = gammaio3 ;
  gammatow = 0.1 ;
  gammatshw = 0.1 ;
  gammatsw = 0.1 ;
  gammatww = 0.1 ;
  gammaio = 0.2 ;
```

```
gammaiw = 0.2 ;
gammato = 0.1 ;
gammatsh = 0.1 ;
gammats = 0.3 ;
gammatw = 0.1 ;
gammaim = 0.1 ;
gammainm = 1.1 ;
gammatm = 0.1 ;
gammatnm = 1.1 ;
gammai = 0.1 ; /*0.1*/
gammat = 0.1 ;/*0.1*/
lastgamma = 0.8 ;
gammatime = 0.2 ;
gammafinal = 0.9;
gammacarorpt = 0.75;

fare = 100.0;

cs = 0.0;

/* init for convergence tests*/
convmor = convoff = convev = 0 ;

totdemmor=totdemoff=totdemeven = 0.0;
pttotdemmor=pttotdemoff=pttotdemeven = 0.0;

/*init*/


for (origin = 1; origin <= 26; origin++)
  {
    xzero[origin] = -1.0;
    for (destination = 1; destination <= 26; destination++)
     {
       demand[origin][destination] = 0.0;
       prix[origin][destination] = 1000.0;
       demandoff[origin][destination] = 0.0;
       prixoff[origin][destination] = 1000.0;
       demandev[origin][destination] = 0.0;
       prixev[origin][destination] = 1000.0;
       /*      if (origin == destination)
         {
           prix[origin][destination] = 0.0;
           prixoff[origin][destination] = 0.0;
           prixev[origin][destination] = 0.0;
         }
       */
     }
  }
/* read data */

/*budgets*/
if( ( filename = fopen("budgets.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for budget\n" );
  }
while (fscanf(filename,"%d %f\n",&origin,&budget) != EOF)
  {
    xzero[origin] = (double)budget;
  }
fclose(filename);

/*demand*/

/*morning*/
if( ( filename = fopen("mor_demand.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for morning demand\n" );
  }
while (fscanf(filename,"%d %d %f\n",&origin,&destination,&od) != EOF)
  {
    demand[origin][destination] = (double)od;
    totdemmor = totdemmor + od;
  }
fclose(filename);

/*off peak*/
```

```
if( ( filename = fopen("off_demand.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for off demand\n" );
  }
while (fscanf(filename,"%d %d %f \n",&origin,&destination,&od) != EOF)
  {
    demandoff[origin][destination] = (double)od;
    totdemoff = totdemoff +od;
  }
fclose(filename);

/*evening*/
if( ( filename = fopen("even_demand.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for evening demand\n" );
  }
while (fscanf(filename,"%d %d %f \n",&origin,&destination,&od) != EOF)
  {
    demandev[origin][destination] = (double)od;
    totdemeven = totdemeven + od;
  }
fclose(filename);

/*prices*/

/*morning*/
if( ( filen = fopen("mor_prices.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for morning prices\n" );
  }
while (fscanf(filen,"%d %d %f \n",&origin,&destination,&price) != EOF)
  {
    prix[origin][destination] = (double)price;
  }
fclose (filen);

/*off peak*/
if( ( filen = fopen("off_prices.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for off prices\n" );
  }
while (fscanf(filen,"%d %d %f \n",&origin,&destination,&price) != EOF)
  {
    prixoff[origin][destination] = (double)price;
  }
fclose (filen);

/*evening*/
if( ( filen = fopen("even_prices.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for evening prices\n" );
  }
while (fscanf(filen,"%d %d %f \n",&origin,&destination,&price) != EOF)
  {
    prixev[origin][destination] = (double)price;
  }
fclose (filen);

/* public transport flows*/

/*morning*/
if( ( filen = fopen("pt_morning.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for morning pt\n" );
  }
while (fscanf(filen,"%d %d %f %f %f %f %f %f %f %f \n",
             &origin,&destination,&ptiw,&ptis,&ptio,&ptish,
             &pttw,&ptts,&ptto,&pttsh) != EOF)
  {
    morptiw[origin][destination] = (double)ptiw;
    morptis[origin][destination] = (double)ptis;
    morptio[origin][destination] = (double)ptio;
    morptish[origin][destination] = (double)ptish;
    morpttw[origin][destination] = (double)pttw;
    morptts[origin][destination] = (double)ptts;
    morptto[origin][destination] = (double)ptto;
    morpttsh[origin][destination] = (double)pttsh;
```

```
    pttotdemmor = pttotdemmor + ptiw+ptis+ptio+ptish+pttw+ptts+ptto+pttsh;
  }
fclose (filen);

/*off*/
if( ( filen = fopen("pt_off.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for off pt\n" );
  }
while (fscanf(filen,"%d %d %f %f %f %f %f %f %f %f \n",
             &origin,&destination,&ptiw,&ptis,&ptio,&ptish,
             &pttw,&ptts,&ptto,&pttsh) != EOF)
  {
    offptiw[origin][destination] = (double)ptiw;
    offptis[origin][destination] = (double)ptis;
    offptio[origin][destination] = (double)ptio;
    offptish[origin][destination] = (double)ptish;
    offpttw[origin][destination] = (double)pttw;
    offptts[origin][destination] = (double)ptts;
    offptto[origin][destination] = (double)ptto;
    offpttsh[origin][destination] = (double)pttsh;

    pttotdemoff = pttotdemoff + ptiw+ptis+ptio+ptish+pttw+ptts+ptto+pttsh;
  }
fclose (filen);

/*evening*/
if( ( filen = fopen("pt_even.dta" , "r" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for evening pt\n" );
  }
while (fscanf(filen,"%d %d %f %f %f %f %f %f %f %f \n",
             &origin,&destination,&ptiw,&ptis,&ptio,&ptish,
             &pttw,&ptts,&ptto,&pttsh) != EOF)
  {
    evenptiw[origin][destination] = (double)ptiw;
    evenptis[origin][destination] = (double)ptis;
    evenptio[origin][destination] = (double)ptio;
    evenptish[origin][destination] = (double)ptish;
    evenpttw[origin][destination] = (double)pttw;
    evenptts[origin][destination] = (double)ptts;
    evenptto[origin][destination] = (double)ptto;
    evenpttsh[origin][destination] = (double)pttsh;

    pttotdemeven = pttotdemeven + ptiw+ptis+ptio+ptish+pttw+ptts+ptto+pttsh;
  }
fclose (filen);

printf ("TOTAL demands at start morning = %f off =%f even =%f total = %f\n",
        totdemmor,totdemoff,totdemeven,totdemmor+totdemoff+totdemeven);
printf ("TOTAL pt demands at start morning = %f off =%f even =%f total = %f\n",
        pttotdemmor,pttotdemoff,pttotdemeven,pttotdemmor+pttotdemoff+pttotdemeven);

totdemmor=totdemoff=totdemeven = 0.0;
pttotdemmor=pttotdemoff=pttotdemeven = 0.0;

/*output files headers*/
if( ( fcarmor = fopen("car_morning.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for morning cars\n" );
  }
if( ( fcaroff = fopen("car_off.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for off cars\n" );
  }
if( ( fcareven = fopen("car_even.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for even cars\n" );
  }
if( ( fptmor = fopen("pt_morning.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for morning public transport\n" );
  }
if( ( fptoff = fopen("pt_off.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for off public transport\n" );
```

```
  }
if( ( fpteven = fopen("pt_even.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for even public transport\n" );
  }
if( ( fbudget = fopen("budgets.dta" , "w" ) ) == NULL )
  {
    printf( "%s", "  \nCould not open file for writing budgets\n" );
  }

fprintf (fcarmor, "MATRIX\n");
fprintf (fcaroff, "MATRIX\n");
fprintf (fcareven, "MATRIX\n");

fprintf (fcarmor, "#morning cars demand \n");
fprintf (fcaroff, "#off cars demand \n");
fprintf (fcareven, "#evening cars demand \n");

/* loop on origins*/

for ( origin =1; origin <= 26; origin++ )
  {
    sprintf(num,"%2d",origin);
    strcpy (name,"alpha");
    strcat(name,num);
    /* open the file where we put the alpha values for this origin */
    if( ( filelog = fopen(name , "w" ) ) == NULL )
      {
        printf( "%s", "  \nCould not open file for alphas\n" );
      }


    #include "tree.c"
    #include "treeoff.c"
    #include "treeeven.c"

    /* level 8 */

    pmor = pow ((ai * pow (pi , (1 - lastgamma)) +
                at * pow (pt , (1 - lastgamma))  )
               , (1.0 / (1 - lastgamma))) ;
    poff = pow ((bi * pow (qi , (1 - lastgamma)) +
                bt * pow (qt , (1 - lastgamma))  )
               , (1.0 / (1 - lastgamma))) ;
    pev =  pow ((ci * pow (ri , (1 - lastgamma)) +
                ct * pow (rt , (1 - lastgamma))  )
               , (1.0 / (1 - lastgamma))) ;

    xmor = (xi*pi + xt*pt )/pmor;
    xoff = (yi*qi + yt*qt )/poff;
    xev  = (zi*ri + zt*rt )/pev;

    three_system (&amor,"amor", &aev,"aev", &aoff,"aoff",
                 xmor, xev, xoff,
                 pmor, pev, poff,
                 gammatime ) ;

    /* level 9 */

    ptrip = pow ((amor * pow (pmor , (1 - gammatime)) +
                 aev * pow (pev , (1 - gammatime)) +
                 aoff * pow (poff , (1 - gammatime))  )
                , (1.0 / (1 - gammatime))) ;
    pother = 1 ;

    xtrip = (xmor*pmor + xev*pev + xoff*poff )/ptrip;
    xother = (xtrip * ptrip) * (17.0/3.0) ;

    two_system (&atrip,"atrip",&aother,"aother",

               xtrip, xother,
               ptrip ,pother ,
               gammafinal);

    p0 = pow ((atrip * pow (ptrip , (1 - gammafinal)) +
              aother * pow (pother , (1 - gammafinal)) )
             , (1.0 / (1 - gammafinal))) ;
```

```
        /* x0*p0 +Taxes*/
        /*x0 = ((calxtrip*calptrip + calxother*calpother )+taxes)/p0*/

        /*budget constraint*/
        if (xzero[origin] < 0.0)
          {
          x0 = (xtrip*ptrip + xother*pother )/p0;
          xzero[origin] = x0*p0;
          }
        else
          {
            x0 = xzero[origin]/p0;
          }

        /*welfare*/

        cs = cs + x0*p0;
        /*cs = cs + xzero[origin];*/

        /* Replication*/

        xother = x0 *aother * pow((p0/pother), gammafinal);

        xtrip = x0 * atrip *  pow((p0/ptrip), gammafinal) ;

        xmor = xtrip * amor *  pow((ptrip/pmor), gammatime) ;
        xoff = xtrip * aoff *  pow((ptrip/poff), gammatime) ;
        xev = xtrip * aev *  pow((ptrip/pev), gammatime) ;

#include "replic.c"
#include "replicoff.c"
#include "repliceven.c"

        /* building updated flows */

        xbyorigin = 0.0;
        dembyorigin = 0.0;

        fprintf (fbudget, "%d %f \n",origin,xzero[origin]);

        for ( destination = 1; destination <= 26; destination++ )
          {
            xcar[destination] = xcarinww[destination] + xcarinws[destination] +
              xcarinwo[destination] + xcarinwsh[destination] +
              xcarthw[destination] + xcarths[destination] +
              xcartho[destination] + xcarthsh[destination] ;
            ycar[destination] = ycarinww[destination] + ycarinws[destination] +
              ycarinwo[destination] + ycarinwsh[destination] +
              ycarthw[destination] + ycarths[destination] +
              ycartho[destination] + ycarthsh[destination] ;
            zcar[destination] = zcarinww[destination] + zcarinws[destination] +
              zcarinwo[destination] + zcarinwsh[destination] +
              zcarthw[destination] + zcarths[destination] +
              zcartho[destination] + zcarthsh[destination] ;

            totdemmor = totdemmor + xcar[destination];
            totdemoff = totdemoff + ycar[destination];
            totdemeven =totdemeven + zcar[destination];

            xpt[destination] = xptinww[destination] + xptinws[destination] +
              xptinwo[destination] + xptinwsh[destination] +
              xptthw[destination] + xptths[destination] +
              xpttho[destination] + xptthsh[destination] ;
            ypt[destination] = yptinww[destination] + yptinws[destination] +
              yptinwo[destination] + yptinwsh[destination] +
              yptthw[destination] + yptths[destination] +
              ypttho[destination] + yptthsh[destination] ;
            zpt[destination] = zptinww[destination] + zptinws[destination] +
              zptinwo[destination] + zptinwsh[destination] +
              zptthw[destination] + zptths[destination] +
              zpttho[destination] + zptthsh[destination] ;

            pttotdemmor = pttotdemmor + xpt[destination];
            pttotdemoff = pttotdemoff + ypt[destination];
            pttotdemeven =pttotdemeven + zpt[destination];
```

```
          xbyorigin = xbyorigin+xcar[destination]+ycar[destination]+zcar[destination]+
                      xpt[destination]+ypt[destination]+zpt[destination];

          dembyorigin = dembyorigin +
demand[origin][destination]+demandoff[origin][destination]+
                        demandev[origin][destination]+
            morptiw[origin][destination]+
            morptis[origin][destination]+
            morptio[origin][destination]+
            morptish[origin][destination]+
            morpttw[origin][destination]+
            morptts[origin][destination]+
            morptto[origin][destination]+
            morpttsh[origin][destination]+
            offptiw[origin][destination]+
            offptis[origin][destination]+
            offptio[origin][destination]+
            offptish[origin][destination]+
            offpttw[origin][destination]+
            offptts[origin][destination]+
            offptto[origin][destination]+
            offpttsh[origin][destination]+
            evenptiw[origin][destination]+
            evenptis[origin][destination]+
            evenptio[origin][destination]+
            evenptish[origin][destination]+
            evenpttw[origin][destination]+
            evenptts[origin][destination]+
            evenptto[origin][destination]+
            evenpttsh[origin][destination];

          /*Convergence tests*/
          if (demand[origin][destination] > 0.0)
            {
              if ((fabs(xcar[destination]-demand[origin][destination]) > 1.0) &&
                  ((fabs(xcar[destination]-
demand[origin][destination])/demand[origin][destination]) > 0.01))
                {
                  convmor++;
                }
            }
          if (demandoff[origin][destination] > 0.0)
            {
              if ((fabs(ycar[destination]-demandoff[origin][destination]) > 1.0) &&
                  ((fabs(ycar[destination]-
demandoff[origin][destination])/demandoff[origin][destination]) > 0.01))
                {
                  convoff++;
                }
            }
          if (demandev[origin][destination] > 0.0)
            {
              if ((fabs(zcar[destination]-demandev[origin][destination]) > 1.0) &&
                  ((fabs(zcar[destination]-
demandev[origin][destination])/demandev[origin][destination]) > 0.01))
                {
                    convev++;
                }
            }
          /*write the outputs files*/

          if (origin != destination)
            {
              if (xcar[destination] <= 0)
                {
                  xcar[destination] = 0;
                }
              if (ycar[destination] <= 0)
                {
                  ycar[destination] = 0;
                }
              if (zcar[destination] <= 0)
                {
                  zcar[destination] = 0;
                }
              fprintf (fcarmor, "%d %d %f \n", origin, destination, xcar[destination]);
```

```
              fprintf (fcaroff, "%d %d %f \n", origin, destination, ycar[destination]);
              fprintf (fcareven, "%d %d %f \n", origin, destination,
zcar[destination]);
            }
          fprintf (fptmor, "%d %d %f %f %f %f %f %f %f %f \n",
                       origin, destination,
                       xptinww[destination],xptinws[destination],
                       xptinwo[destination],xptinwsh[destination],
                       xptthw[destination],xptths[destination],
                       xpttho[destination],xptthsh[destination]);
          fprintf (fptoff, "%d %d %f %f %f %f %f %f %f %f \n",
                       origin, destination,
                       yptinww[destination],yptinws[destination],
                       yptinwo[destination],yptinwsh[destination],
                   yptthw[destination],yptths[destination],
                   yptho[destination],yptthsh[destination]);
          fprintf (fpteven, "%d %d %f %f %f %f %f %f %f %f \n",
                   origin, destination,
                   zptinww[destination],zptinws[destination],
                   zptinwo[destination],zptinwsh[destination],
                   zptthw[destination],zptths[destination],
                   zpttho[destination],zptthsh[destination]);
        }

    }

  /* output files footers*/
  fprintf (fcarmor, "END\n");
  fprintf (fcaroff, "END\n");
  fprintf (fcareven, "END\n");

  fclose (fcarmor);
  fclose (fcaroff);
  fclose (fcareven);
  fclose (fptmor);
  fclose (fptoff);
  fclose (fpteven);
  fclose (fbudget);

  /*testing if there is convergence*/
  printf ("convmor = %d convoff =%d convev = %d \n",convmor,convoff,convev);

  printf ( "CS = %f \n", cs);
  printf ("TOTAL demands at end morning = %f off =%f even =%f total = %f\n",
          totdemmor,totdemoff,totdemeven,totdemmor+totdemoff+totdemeven);
  printf ("TOTAL pt demands at end morning = %f off =%f even =%f total = %f\n",
          pttotdemmor,pttotdemoff,pttotdemeven,pttotdemmor+pttotdemoff+pttotdemeven);

  if( ( filename = fopen("convergent" , "w" ) ) == NULL )
    {
      printf( "%s", "  \nCould not open file for convergence\n" );
    }

  if ((convmor == 0) && (convoff == 0) && (convev == 0))
    {
      fprintf (filename,"stop\n");
    }
  else
    {
      fprintf (filename,"continue\n");
    }
  fclose (filename);

  fclose(filelog);

};


#include "routines.c"

#include "lapack.c"
```

3.declar

```
int origin, /* origin of a trip = a consumer*/
  destination; /* each trip destination*/

int iloop; /*a loop counter*/

int test;

char name[7],num[2];

/* for convergence tests : convergence if these 3 var remain nul*/
int convmor,convoff,convev;

float  od,price,budget,alpha_val;

float ptiw,ptis,ptio,ptish,pttw,ptts,ptto,pttsh;

double totdemmor,totdemoff,totdemeven,
      pttotdemmor,pttotdemoff,pttotdemeven;

double xbyorigin,dembyorigin;

double pond;

double cs,x0total,p0total;

double ratio;

/*input files*/
FILE  *filename, *filen;

/*output files*/
FILE *fcarmor,*fcaroff,*fcareven,*fptmor,*fptoff,*fpteven,*fbudget,*filelog;

/* alpha stored when no calibration*/

double val_alpha[1145];

double

  xzero[27],
  demand[27][27],
  prix[27][27],
  demandoff[27][27],
  prixoff[27][27],
  demandev[27][27],
  prixev[27][27],
  morptiw[27][27],
  morptis[27][27],
  morptio[27][27],
  morptish[27][27],
  morpttw[27][27],
  morptts[27][27],
  morptto[27][27],
  morpttsh[27][27],
  offptiw[27][27],
  offptis[27][27],
  offptio[27][27],
  offptish[27][27],
  offpttw[27][27],
  offptts[27][27],
  offptto[27][27],
  offpttsh[27][27],
  evenptiw[27][27],
  evenptis[27][27],
  evenptio[27][27],
  evenptish[27][27],
  evenpttw[27][27],
  evenptts[27][27],
  evenptto[27][27],
  evenpttsh[27][27],
  gammabase,   /* gamma between nodes (lowest level) */
  gammaishnc,   /* gamma for inward shopping in Namur centre*/
```

```
gammaionc,   /* gamma for inward other north centre*/
gammaiow,    /* gamma for inward other west*/
gammaiosc,   /* gamma for inward other south centre*/
gammaioe,    /* gamma for inward other east*/
gammais,     /* gamma for inward school*/
gammaiwnc,   /* gamma for inward work north centre*/
gammaiww,    /* gamma for inward work west*/
gammaiwsc,   /* gamma for inward work south centre*/
gammaiwe,    /* gamma for inward work east*/
gammatonw,   /* gamma for through other north west*/
gammatosw,   /* gamma for through other south west*/
gammatoe,    /* gamma for through other east*/
gammatshnw,   /* gamma for through shopping north west*/
gammatshsw,   /* gamma for through shopping south west*/
gammatshe,    /* gamma for through shopping east*/
gammatsnw,   /* gamma for through school north west*/
gammatssw,   /* gamma for through school south west*/
gammatse,    /* gamma for through school east*/
gammatwnw,   /* gamma for through work north west*/
gammatwsw,   /* gamma for through work south west*/
gammatwe,    /* gamma for through work east*/
/* same convention for the rest*/
gammaish,
gammaiw2,
gammaiw3,
gammaio2,
gammaio3,
gammais2,
gammaiw2,
gammaiw3,
gammatosw2,
gammatshsw2,
gammatssw2,
gammatwsw2,
gammaiwnwc,
gammaionwc,
gammatow,
gammatshw,
gammatsw,
gammatww,
gammaio,
gammaiw,
gammato,
gammatsh,
gammats,
gammatw,
gammaim,
gammainm,
gammatm,
gammatnm,
gammai,
gammat,
lastgamma,
gammatime,
gammafinal,
gammacarorpt,

xcar[27], /* flow from origin to destination (computed by ATES)*/
pcar[27], /* price for car trip from origin to destination (from ATES)*/
ppt[27], /* price for public transport trip*/
/* prices for each purpose*/
piw[27],pio[27],pis[27],pish[27],ptw[27],pto[27],pts[27],ptsh[27],
xinw[27], /* inward part of the trips*/
xthrough[27], /* through part of the trips*/
xinww[27], /* inward work trips*/
xinws[27], /* inward school trips*/
xinwsh[27], /* inward shopping trips*/
xinwo[27], /* inward other trips*/
xthw[27], /* through work trips*/
xths[27], /* through school trips*/
xthsh[27], /* through shopping trips*/
xtho[27], /* through other trips*/
xptinww[27], /* inward work pt trips*/
xptinws[27], /* inward school pt trips*/
xptinwsh[27], /* inward shopping pt trips*/
xptinwo[27], /* inward other pt trips*/
xptthw[27], /* through work pt trips*/
```

```
  xptths[27], /* through school pt trips*/
  xptthsh[27], /* through shopping pt trips*/
  xpttho[27], /* through other pt trips*/
  xcarinww[27], /* inward work car trips*/
  xcarinws[27], /* inward school car trips*/
  xcarinwsh[27], /* inward shopping car trips*/
  xcarinwo[27], /* inward other car trips*/
  xcarthw[27], /* through work car trips*/
  xcarths[27], /* through school car trips*/
  xcarthsh[27], /* through shopping car trips*/
  xcartho[27], /* through other car trips*/

  /* the alpha */
  acariw24,acariw25,acariw22,acariw23,acariw20,acariw21,acariw26,acariw19,acariw17,
  acariw18,acariw12,acariw13,acariw14,acariw6,acariw7,acariw8,acariw9,
  acariw10,acariw11,acariw4,acariw5,acariw1,acariw2,acariw3,acariw15,acariw16,
  aptiw24,aptiw25,aptiw22,aptiw23,aptiw20,aptiw21,aptiw26,aptiw19,aptiw17,
  aptiw18,aptiw12,aptiw13,aptiw14,aptiw6,aptiw7,aptiw8,aptiw9,
  aptiw10,aptiw11,aptiw4,aptiw5,aptiw1,aptiw2,aptiw3,aptiw15,aptiw16,
  acario24,acario25,acario22,acario23,acario20,acario21,acario26,acario19,acario17,
  acario18,acario12,acario13,acario14,acario6,acario7,acario8,acario9,
  acario10,acario11,acario4,acario5,acario1,acario2,acario3,acario15,acario16,
  aptio24,aptio25,aptio22,aptio23,aptio20,aptio21,aptio26,aptio19,aptio17,
  aptio18,aptio12,aptio13,aptio14,aptio6,aptio7,aptio8,aptio9,
  aptio10,aptio11,aptio4,aptio5,aptio1,aptio2,aptio3,aptio15,aptio16,
  acaris3,acaris10,acaris9,acaris11,acaris12,acaris13,acaris14,acaris20,acaris26,
  aptis3,aptis10,aptis9,aptis11,aptis12,aptis13,aptis14,aptis20,aptis26,
  acarish3,acarish12,acarish13,acarish14,acarish16,acarish17,acarish20,
  aptish3,aptish12,aptish13,aptish14,aptish16,aptish17,aptish20,
  acartw7,acartw8,acartw4,acartw19,acartw20,acartw22,acartw23,acartw1,acartw24,
  apttw7,apttw8,apttw4,apttw19,apttw20,apttw22,apttw23,apttw1,apttw24,
  acarto7,acarto8,acarto4,acarto19,acarto20,acarto22,acarto23,acarto1,acarto24,
  aptto7,aptto8,aptto4,aptto19,aptto20,aptto22,aptto23,aptto1,aptto24,
  acarts7,acarts8,acarts4,acarts19,acarts20,acarts22,acarts23,acarts1,acarts24,
  aptts7,aptts8,aptts4,aptts19,aptts20,aptts22,aptts23,aptts1,aptts24,

acartsh7,acartsh8,acartsh4,acartsh19,acartsh20,acartsh22,acartsh23,acartsh1,acartsh24,
  apttsh7,apttsh8,apttsh4,apttsh19,apttsh20,apttsh22,apttsh23,apttsh1,apttsh24,
  aishn3, aishn12,aishn13,aishn14,aishn16,aishn17,
  aionc1,aionc2,aionc3,aionc5,aionc15,aionc16,
  aionc4,aionc5,
  aiow9,aiow10,aiow11,
  aiow6,aiow7,aiow8,
  aiosc12,aiosc13,aiosc14,
  aiosc17,aiosc18,
  aioe20,aioe21,aioe26,
  aioe22,aioe23,
  aioe24,aioe25,
  ais3,ais10,
  aiwnc1,aiwnc2,aiwnc3,aiwnc5,aiwnc15,aiwnc16,
  aiwnc4,aiwnc5,
  aiww9,aiww10,aiww11,
  aiww6,aiww7,aiww8,
  aiwsc12,aiwsc13,aiwsc14,
  aiwsc17,aiwsc18,
  aiwe20,aiwe21,aiwe26,
  aiwe22,aiwe23,
  aiwe24,aiwe25,
  atonw1,atonw24,
  atosw22,atosw23,
  atosw19,atosw20,
  atoe7,atoe8,ato4,
  atshnw1,atshnw24,
  atshsw22,atshsw23,
  atshsw19,atshsw20,
  atshe7,atshe8,atsh4,
  atsnw1,atsnw24,
  atssw22,atssw23,
  atssw19,atssw20,
  atse7,atse8,ats4,
  atwnw1,atwnw24,
  atwsw22,atwsw23,
  atwsw19,atwsw20,
  atwe7,atwe8,atw4,
  aishnc,aishj,
  ais310,ais9,ais11,ais12,ais13,ais14,ais20,ais26,
  aiwnc45,aiwnc1231516,
```

```
aiww678,aiww91011,
aiwsc1718,aiwsc121314,
aiwe2425,aiwe2223,aiwe202126,aiwe19,
aionc45,aionc1231516,
aiow678,aiow91011,
aiosc1718,aiosc121314,
aioe2425,aioe2223,aioe202126,aioe19,
atosw1920,atosw2223,
atshsw1920,atshsw2223,
atssw1920,atssw2223,
atwsw1920,atwsw2223,
aiww,aiwnc,
aiow,aionc,
atosw,atonw,
atshsw,atshnw,
atssw,atsnw,
atwsw,atwnw,
aioe,aiosc,aionwc,
aiwe,aiwsc,aiwnwc,
atoe,atow,
atshe,atshw,
atse,atsw,
atwe,atww,
aiw,ais,
aio,aish,
atw,ats,
ato,atsh,
aim,ainm,
atm,atnm,
ai,at,
amor,aev,aoff,
atrip,aother,

/* the prices */
pishn3, pishn12,pishn13,pishn14,pishn16,pishn17,
pionc1,pionc2,pionc3,pionc5,pionc15,pionc16,
pionc4,pionc5,
piow9,piow10,piow11,
piow6,piow7,piow8,
piosc12,piosc13,piosc14,
piosc17,piosc18,
pioe20,pioe21,
pioe22,pioe23,
pioe24,pioe25,
pis3,pis10,
piwnc1,piwnc2,piwnc3,piwnc5,piwnc15,piwnc16,
piwnc4,piwnc5,
piww9,piww10,piww11,
piww6,piww7,piww8,
piwsc12,piwsc13,piwsc14,
piwsc17,piwsc18,
piwe20,piwe21,
piwe22,piwe23,
piwe24,piwe25,
ptonw1,ptonw24,
ptosw22,ptosw23,
ptosw19,ptosw20,
ptoe6,ptoe8,
ptshnw1,ptshnw24,
ptshsw22,ptshsw23,
ptshsw19,ptshsw20,
ptshe6,ptshe8,
ptsnw1,ptsnw24,
ptssw22,ptssw23,
ptssw19,ptssw20,
ptse6,ptse8,
ptwnw1,ptwnw24,
ptwsw22,ptwsw23,
ptwsw19,ptwsw20,
ptwe6,ptwe8,
pishnc,pishj,
pis310,pis9,pis11,pis12,pis13,pis14,pis20,pis26,
piwnc45,piwnc1231516,
piww678,piww91011,
piwsc1718,piwsc121314,
piwe2425,piwe2223,piwe202126,piwe19,
pionc45,pionc1231516,
```

```
piow678,piow91011,
piosc1718,piosc121314,
pioe2425,pioe2223,pioe202126,pioe19,
ptosw1920,ptosw2223,
ptshsw1920,ptshsw2223,
ptssw1920,ptssw2223,
ptwsw1920,ptwsw2223,
piww,piwnc,
piow,pionc,
ptosw,ptonw,
ptshsw,ptshnw,
ptssw,ptsnw,
ptwsw,ptwnw,
pioe,piosc,pionwc,
piwe,piwsc,piwnwc,
ptoe,ptow,
ptshe,ptshw,
ptse,ptsw,
ptwe,ptww,
pinw,pins,
pino,pinsh,
pthw,pths,
ptho,pthsh,
pim,pinm,
ptm,ptnm,
pi,pt,
pmor,pev,poff,
ptrip,pother,
p0,

fare, /* price added for public transport */

/* the flows */
xishn3, xishn12,xishn13,xishn14,xishn16,xishn17,
xionc1,xionc2,xionc3,xionc5,xionc15,xionc16,
xionc4,xionc5,
xiow9,xiow10,xiow11,
xiow6,xiow7,xiow8,
xiosc12,xiosc13,xiosc14,
xiosc17,xiosc18,
xioe20,xioe21,
xioe22,xioe23,
xioe24,xioe25,
xis3,xis10,
xiwnc1,xiwnc2,xiwnc3,xiwnc5,xiwnc15,xiwnc16,
xiwnc4,xiwnc5,
xiww9,xiww10,xiww11,
xiww6,xiww7,xiww8,
xiwsc12,xiwsc13,xiwsc14,
xiwsc17,xiwsc18,
xiwe20,xiwe21,
xiwe22,xiwe23,
xiwe24,xiwe25,
xtonw1,xtonw24,
xtosw22,xtosw23,
xtosw19,xtosw20,
xtoe6,xtoe8,
xtshnw1,xtshnw24,
xtshsw22,xtshsw23,
xtshsw19,xtshsw20,
xtshe6,xtshe8,
xtsnw1,xtsnw24,
xtssw22,xtssw23,
xtssw19,xtssw20,
xtse6,xtse8,
xtwnw1,xtwnw24,
xtwsw22,xtwsw23,
xtwsw19,xtwsw20,
xtwe6,xtwe8,
xishnc,
xis310,
xiwnc45,xiwnc1231516,
xiww678,xiww91011,
xiwsc1718,xiwsc121314,
xiwe2425,xiwe2223,xiwe202126,xiwe19,
xionc45,xionc1231516,
xiow678,xiow91011,
```

```
  xiosc1718,xiosc121314,
  xioe2425,xioe2223,xioe202126,xioe19,
  xtosw1920,xtosw2223,
  xtshsw1920,xtshsw2223,
  xtssw1920,xtssw2223,
  xtwsw1920,xtwsw2223,
  xiww,xiwnc,
  xiow,xionc,
  xtosw,xtonw,
  xtshsw,xtshnw,
  xtssw,xtsnw,
  xtwsw,xtwnw,
  xioe,xiosc,xionwc,
  xiwe,xiwsc,xiwnwc,
  xtoe,xtow,
  xtshe,xtshw,
  xtse,xtsw,
  xtwe,xtww,
  xiw,xis,
  xio,xish,
  xtw,xts,
  xto,xtsh,
  xim,xinm,
  xtm,xtnm,
  xi,xt,
  xmor,xev,xoff,
  xtrip,xother,
  x0,

  /*for off peak trips the a in alpha variables becomes b, the p in prices become q
and
    the x in flows become y
*/

  ycar[27], /* flow from origin to destination (computed by ATES)*/
  qcar[27], /* price for car triq from origin to destination (from ATES)*/
  qpt[27], /* price for public transport trip*/
  /* prices for each purpose*/
  qiw[27],qio[27],qis[27],qish[27],qtw[27],qto[27],qts[27],qtsh[27],
  yinw[27], /* inward part of the trips*/
  ythrough[27], /* through part of the trips*/
  yinww[27], /* inward work trips*/
  yinws[27], /* inward school trips*/
  yinwsh[27], /* inward shopping trips*/
  yinwo[27], /* inward other trips*/
  ythw[27], /* through work trips*/
  yths[27], /* through school trips*/
  ythsh[27], /* through shopping trips*/
  ytho[27], /* through other trips*/
  yptinww[27], /* inward work pt trips*/
  yptinws[27], /* inward school pt trips*/
  yptinwsh[27], /* inward shopping pt trips*/
  yptinwo[27], /* inward other pt trips*/
  yptthw[27], /* through work pt trips*/
  ypttths[27], /* through school pt trips*/
  yptthsh[27], /* through shopping pt trips*/
  ypttho[27], /* through other pt trips*/
  ycarinww[27], /* inward work car trips*/
  ycarinws[27], /* inward school car trips*/
  ycarinwsh[27], /* inward shopping car trips*/
  ycarinwo[27], /* inward other car trips*/
  ycarthw[27], /* through work car trips*/
  ycarths[27], /* through school car trips*/
  ycarthsh[27], /* through shopping car trips*/
  ycartho[27], /* through other car trips*/

  /* the alpha */
  bcariw24,bcariw25,bcariw22,bcariw23,bcariw20,bcariw21,bcariw26,bcariw19,bcariw17,
  bcariw18,bcariw12,bcariw13,bcariw14,bcariw6,bcariw7,bcariw8,bcariw9,
  bcariw10,bcariw11,bcariw4,bcariw5,bcariw1,bcariw2,bcariw3,bcariw15,bcariw16,
  bptiw24,bptiw25,bptiw22,bptiw23,bptiw20,bptiw21,bptiw26,bptiw19,bptiw17,
  bptiw18,bptiw12,bptiw13,bptiw14,bptiw6,bptiw7,bptiw8,bptiw9,
  bptiw10,bptiw11,bptiw4,bptiw5,bptiw1,bptiw2,bptiw3,bptiw15,bptiw16,
  bcario24,bcario25,bcario22,bcario23,bcario20,bcario21,bcario26,bcario19,bcario17,
  bcario18,bcario12,bcario13,bcario14,bcario6,bcario7,bcario8,bcario9,
  bcario10,bcario11,bcario4,bcario5,bcario1,bcario2,bcario3,bcario15,bcario16,
  bptio24,bptio25,bptio22,bptio23,bptio20,bptio21,bptio26,bptio19,bptio17,
```

```
  bptio18,bptio12,bptio13,bptio14,bptio6,bptio7,bptio8,bptio9,
  bptio10,bptio11,bptio4,bptio5,bptio1,bptio2,bptio3,bptio15,bptio16,
  bcaris3,bcaris10,bcaris9,bcaris11,bcaris12,bcaris13,bcaris14,bcaris20,bcaris26,
  bptis3,bptis10,bptis9,bptis11,bptis12,bptis13,bptis14,bptis20,bptis26,
  bcarish3,bcarish12,bcarish13,bcarish14,bcarish16,bcarish17,bcarish20,
  bptish3,bptish12,bptish13,bptish14,bptish16,bptish17,bptish20,
  bcartw7,bcartw8,bcartw4,bcartw19,bcartw20,bcartw22,bcartw23,bcartw1,bcartw24,
  bpttw7,bpttw8,bpttw4,bpttw19,bpttw20,bpttw22,bpttw23,bpttw1,bpttw24,
  bcarto7,bcarto8,bcarto4,bcarto19,bcarto20,bcarto22,bcarto23,bcarto1,bcarto24,
  bptto7,bptto8,bptto4,bptto19,bptto20,bptto22,bptto23,bptto1,bptto24,
  bcarts7,bcarts8,bcarts4,bcarts19,bcarts20,bcarts22,bcarts23,bcarts1,bcarts24,
  bptts7,bptts8,bptts4,bptts19,bptts20,bptts22,bptts23,bptts1,bptts24,

bcartsh7,bcartsh8,bcartsh4,bcartsh19,bcartsh20,bcartsh22,bcartsh23,bcartsh1,bcartsh24,
  bpttsh7,bpttsh8,bpttsh4,bpttsh19,bpttsh20,bpttsh22,bpttsh23,bpttsh1,bpttsh24,
  bishn3, bishn12,bishn13,bishn14,bishn16,bishn17,
  bionc1,bionc2,bionc3,bionc5,bionc15,bionc16,
  bionc4,bionc5,
  biow9,biow10,biow11,
  biow6,biow7,biow8,
  biosc12,biosc13,biosc14,
  biosc17,biosc18,
  bioe20,bioe21,bioe26,
  bioe22,bioe23,
  bioe24,bioe25,
  bis3,bis10,
  biwnc1,biwnc2,biwnc3,biwnc5,biwnc15,biwnc16,
  biwnc4,biwnc5,
  biww9,biww10,biww11,
  biww6,biww7,biww8,
  biwsc12,biwsc13,biwsc14,
  biwsc17,biwsc18,
  biwe20,biwe21,biwe26,
  biwe22,biwe23,
  biwe24,biwe25,
  btonw1,btonw24,
  btosw22,btosw23,
  btosw19,btosw20,
  btoe7,btoe8,bto4,
  btshnw1,btshnw24,
  btshsw22,btshsw23,
  btshsw19,btshsw20,
  btshe7,btshe8,btsh4,
  btsnw1,btsnw24,
  btssw22,btssw23,
  btssw19,btssw20,
  btse7,btse8,bts4,
  btwnw1,btwnw24,
  btwsw22,btwsw23,
  btwsw19,btwsw20,
  btwe7,btwe8,btw4,
  bishnc,bishj,
  bis310,bis9,bis11,bis12,bis13,bis14,bis20,bis26,
  biwnc45,biwnc1231516,
  biww678,biww91011,
  biwsc1718,biwsc121314,
  biwe2425,biwe2223,biwe202126,biwe19,
  bionc45,bionc1231516,
  biow678,biow91011,
  biosc1718,biosc121314,
  bioe2425,bioe2223,bioe202126,bioe19,
  btosw1920,btosw2223,
  btshsw1920,btshsw2223,
  btssw1920,btssw2223,
  btwsw1920,btwsw2223,
  biww,biwnc,
  biow,bionc,
  btosw,btonw,
  btshsw,btshnw,
  btssw,btsnw,
  btwsw,btwnw,
  bioe,biosc,bionwc,
  biwe,biwsc,biwnwc,
  btoe,btow,
  btshe,btshw,
  btse,btsw,
  btwe,btww,
```

```
biw,bis,
bio,bish,
btw,bts,
bto,btsh,
bim,binm,
btm,btnm,
bi,bt,
bmor,bev,boff,
btriq,bother,

/* the prices */
qishn3, qishn12,qishn13,qishn14,qishn16,qishn17,
qionc1,qionc2,qionc3,qionc5,qionc15,qionc16,
qionc4,qionc5,
qiow9,qiow10,qiow11,
qiow6,qiow7,qiow8,
qiosc12,qiosc13,qiosc14,
qiosc17,qiosc18,
qioe20,qioe21,
qioe22,qioe23,
qioe24,qioe25,
qis3,qis10,
qiwnc1,qiwnc2,qiwnc3,qiwnc5,qiwnc15,qiwnc16,
qiwnc4,qiwnc5,
qiww9,qiww10,qiww11,
qiww6,qiww7,qiww8,
qiwsc12,qiwsc13,qiwsc14,
qiwsc17,qiwsc18,
qiwe20,qiwe21,
qiwe22,qiwe23,
qiwe24,qiwe25,
qtonw1,qtonw24,
qtosw22,qtosw23,
qtosw19,qtosw20,
qtoe6,qtoe8,
qtshnw1,qtshnw24,
qtshsw22,qtshsw23,
qtshsw19,qtshsw20,
qtshe6,qtshe8,
qtsnw1,qtsnw24,
qtssw22,qtssw23,
qtssw19,qtssw20,
qtse6,qtse8,
qtwnw1,qtwnw24,
qtwsw22,qtwsw23,
qtwsw19,qtwsw20,
qtwe6,qtwe8,
qishnc,qishj,
qis310,qis9,qis11,qis12,qis13,qis14,qis20,qis26,
qiwnc45,qiwnc1231516,
qiww678,qiww91011,
qiwsc1718,qiwsc121314,
qiwe2425,qiwe2223,qiwe202126,qiwe19,
qionc45,qionc1231516,
qiow678,qiow91011,
qiosc1718,qiosc121314,
qioe2425,qioe2223,qioe202126,qioe19,
qtosw1920,qtosw2223,
qtshsw1920,qtshsw2223,
qtssw1920,qtssw2223,
qtwsw1920,qtwsw2223,
qiww,qiwnc,
qiow,qionc,
qtosw,qtonw,
qtshsw,qtshnw,
qtssw,qtsnw,
qtwsw,qtwnw,
qioe,qiosc,qionwc,
qiwe,qiwsc,qiwnwc,
qtoe,qtow,
qtshe,qtshw,
qtse,qtsw,
qtwe,qtww,
qinw,qins,
qino,qinsh,
qthw,qths,
qtho,qthsh,
```

```
  qim,qinm,
  qtm,qtnm,
  qi,qt,

  /* the flows */
  yishn3, yishn12,yishn13,yishn14,yishn16,yishn17,
  yionc1,yionc2,yionc3,yionc5,yionc15,yionc16,
  yionc4,yionc5,
  yiow9,yiow10,yiow11,
  yiow6,yiow7,yiow8,
  yiosc12,yiosc13,yiosc14,
  yiosc17,yiosc18,
  yioe20,yioe21,
  yioe22,yioe23,
  yioe24,yioe25,
  yis3,yis10,
  yiwnc1,yiwnc2,yiwnc3,yiwnc5,yiwnc15,yiwnc16,
  yiwnc4,yiwnc5,
  yiww9,yiww10,yiww11,
  yiww6,yiww7,yiww8,
  yiwsc12,yiwsc13,yiwsc14,
  yiwsc17,yiwsc18,
  yiwe20,yiwe21,
  yiwe22,yiwe23,
  yiwe24,yiwe25,
  ytonw1,ytonw24,
  ytosw22,ytosw23,
  ytosw19,ytosw20,
  ytoe6,ytoe8,
  ytshnw1,ytshnw24,
  ytshsw22,ytshsw23,
  ytshsw19,ytshsw20,
  ytshe6,ytshe8,
  ytsnw1,ytsnw24,
  ytssw22,ytssw23,
  ytssw19,ytssw20,
  ytse6,ytse8,
  ytwnw1,ytwnw24,
  ytwsw22,ytwsw23,
  ytwsw19,ytwsw20,
  ytwe6,ytwe8,
  yishnc,
  yis310,
  yiwnc45,yiwnc1231516,
  yiww678,yiww91011,
  yiwsc1718,yiwsc121314,
  yiwe2425,yiwe2223,yiwe202126,yiwe19,
  yionc45,yionc1231516,
  yiow678,yiow91011,
  yiosc1718,yiosc121314,
  yioe2425,yioe2223,yioe202126,yioe19,
  ytosw1920,ytosw2223,
  ytshsw1920,ytshsw2223,
  ytssw1920,ytssw2223,
  ytwsw1920,ytwsw2223,
  yiww,yiwnc,
  yiow,yionc,
  ytosw,ytonw,
  ytshsw,ytshnw,
  ytssw,ytsnw,
  ytwsw,ytwnw,
  yioe,yiosc,yionwc,
  yiwe,yiwsc,yiwnwc,
  ytoe,ytow,
  ytshe,ytshw,
  ytse,ytsw,
  ytwe,ytww,
  yiw,yis,
  yio,yish,
  ytw,yts,
  yto,ytsh,
  yim,yinm,
  ytm,ytnm,
  yi,yt,

  /*for even peak trips the a in alpha variables becomes c, the p in prices become r
and
```

```
    the x in flows become z
*/

  zcar[27], /* flow from origin to destination (computed cz ATES)*/
  rcar[27], /* price for car trir from origin to destination (from ATES)*/
  rpt[27], /* price for puclic transport trip*/
  /* prices for each purpose*/
  riw[27],rio[27],ris[27],rish[27],rtw[27],rto[27],rts[27],rtsh[27],
  zinw[27], /* inward part of the trips*/
  zthrough[27], /* through part of the trips*/
  zinww[27], /* inward work trips*/
  zinws[27], /* inward school trips*/
  zinwsh[27], /* inward shopping trips*/
  zinwo[27], /* inward other trips*/
  zthw[27], /* through work trips*/
  zths[27], /* through school trips*/
  zthsh[27], /* through shopping trips*/
  ztho[27], /* through other trips*/
  zptinww[27], /* inward work pt trips*/
  zptinws[27], /* inward school pt trips*/
  zptinwsh[27], /* inward shopping pt trips*/
  zptinwo[27], /* inward other pt trips*/
  zptthw[27], /* through work pt trips*/
  zptths[27], /* through school pt trips*/
  zptthsh[27], /* through shopping pt trips*/
  zpttho[27], /* through other pt trips*/
  zcarinww[27], /* inward work car trips*/
  zcarinws[27], /* inward school car trips*/
  zcarinwsh[27], /* inward shopping car trips*/
  zcarinwo[27], /* inward other car trips*/
  zcarthw[27], /* through work car trips*/
  zcarths[27], /* through school car trips*/
  zcarthsh[27], /* through shopping car trips*/
  zcartho[27], /* through other car trips*/

  /* the alpha */
  ccariw24,ccariw25,ccariw22,ccariw23,ccariw20,ccariw21,ccariw26,ccariw19,ccariw17,
  ccariw18,ccariw12,ccariw13,ccariw14,ccariw6,ccariw7,ccariw8,ccariw9,
  ccariw10,ccariw11,ccariw4,ccariw5,ccariw1,ccariw2,ccariw3,ccariw15,ccariw16,
  cptiw24,cptiw25,cptiw22,cptiw23,cptiw20,cptiw21,cptiw26,cptiw19,cptiw17,
  cptiw18,cptiw12,cptiw13,cptiw14,cptiw6,cptiw7,cptiw8,cptiw9,
  cptiw10,cptiw11,cptiw4,cptiw5,cptiw1,cptiw2,cptiw3,cptiw15,cptiw16,
  ccario24,ccario25,ccario22,ccario23,ccario20,ccario21,ccario26,ccario19,ccario17,
  ccario18,ccario12,ccario13,ccario14,ccario6,ccario7,ccario8,ccario9,
  ccario10,ccario11,ccario4,ccario5,ccario1,ccario2,ccario3,ccario15,ccario16,
  cptio24,cptio25,cptio22,cptio23,cptio20,cptio21,cptio26,cptio19,cptio17,
  cptio18,cptio12,cptio13,cptio14,cptio6,cptio7,cptio8,cptio9,
  cptio10,cptio11,cptio4,cptio5,cptio1,cptio2,cptio3,cptio15,cptio16,
  ccaris3,ccaris10,ccaris9,ccaris11,ccaris12,ccaris13,ccaris14,ccaris20,ccaris26,
  cptis3,cptis10,cptis9,cptis11,cptis12,cptis13,cptis14,cptis20,cptis26,
  ccarish3,ccarish12,ccarish13,ccarish14,ccarish16,ccarish17,ccarish20,
  cptish3,cptish12,cptish13,cptish14,cptish16,cptish17,cptish20,
  ccartw7,ccartw8,ccartw4,ccartw19,ccartw20,ccartw22,ccartw23,ccartw1,ccartw24,
  cpttw7,cpttw8,cpttw4,cpttw19,cpttw20,cpttw22,cpttw23,cpttw1,cpttw24,
  ccarto7,ccarto8,ccarto4,ccarto19,ccarto20,ccarto22,ccarto23,ccarto1,ccarto24,
  cptto7,cptto8,cptto4,cptto19,cptto20,cptto22,cptto23,cptto1,cptto24,
  ccarts7,ccarts8,ccarts4,ccarts19,ccarts20,ccarts22,ccarts23,ccarts1,ccarts24,
  cptts7,cptts8,cptts4,cptts19,cptts20,cptts22,cptts23,cptts1,cptts24,

ccartsh7,ccartsh8,ccartsh4,ccartsh19,ccartsh20,ccartsh22,ccartsh23,ccartsh1,ccartsh24,
  cpttsh7,cpttsh8,cpttsh4,cpttsh19,cpttsh20,cpttsh22,cpttsh23,cpttsh1,cpttsh24,
  cishn3, cishn12,cishn13,cishn14,cishn16,cishn17,
  cionc1,cionc2,cionc3,cionc5,cionc15,cionc16,
  cionc4,cionc5,
  ciow9,ciow10,ciow11,
  ciow6,ciow7,ciow8,
  ciosc12,ciosc13,ciosc14,
  ciosc17,ciosc18,
  cioe20,cioe21,cioe26,
  cioe22,cioe23,
  cioe24,cioe25,
  cis3,cis10,
  ciwnc1,ciwnc2,ciwnc3,ciwnc5,ciwnc15,ciwnc16,
  ciwnc4,ciwnc5,
  ciww9,ciww10,ciww11,
  ciww6,ciww7,ciww8,
  ciwsc12,ciwsc13,ciwsc14,
```

```
ciwsc17,ciwsc18,
ciwe20,ciwe21,ciwe26,
ciwe22,ciwe23,
ciwe24,ciwe25,
ctonw1,ctonw24,
ctosw22,ctosw23,
ctosw19,ctosw20,
ctoe7,ctoe8,cto4,
ctshnw1,ctshnw24,
ctshsw22,ctshsw23,
ctshsw19,ctshsw20,
ctshe7,ctshe8,ctsh4,
ctsnw1,ctsnw24,
ctssw22,ctssw23,
ctssw19,ctssw20,
ctse7,ctse8,cts4,
ctwnw1,ctwnw24,
ctwsw22,ctwsw23,
ctwsw19,ctwsw20,
ctwe7,ctwe8,ctw4,
cishnc,cishj,
cis310,cis9,cis11,cis12,cis13,cis14,cis20,cis26,
ciwnc45,ciwnc1231516,
ciww678,ciww91011,
ciwsc1718,ciwsc121314,
ciwe2425,ciwe2223,ciwe202126,ciwe19,
cionc45,cionc1231516,
ciow678,ciow91011,
ciosc1718,ciosc121314,
cioe2425,cioe2223,cioe202126,cioe19,
ctosw1920,ctosw2223,
ctshsw1920,ctshsw2223,
ctssw1920,ctssw2223,
ctwsw1920,ctwsw2223,
ciww,ciwnc,
ciow,cionc,
ctosw,ctonw,
ctshsw,ctshnw,
ctssw,ctsnw,
ctwsw,ctwnw,
cioe,ciosc,cionwc,
ciwe,ciwsc,ciwnwc,
ctoe,ctow,
ctshe,ctshw,
ctse,ctsw,
ctwe,ctww,
ciw,cis,
cio,cish,
ctw,cts,
cto,ctsh,
cim,cinm,
ctm,ctnm,
ci,ct,
cmor,cev,coff,
ctrir,cother,

/* the prices */
rishn3, rishn12,rishn13,rishn14,rishn16,rishn17,
rionc1,rionc2,rionc3,rionc5,rionc15,rionc16,
rionc4,rionc5,
riow9,riow10,riow11,
riow6,riow7,riow8,
riosc12,riosc13,riosc14,
riosc17,riosc18,
rioe20,rioe21,
rioe22,rioe23,
rioe24,rioe25,
ris3,ris10,
riwnc1,riwnc2,riwnc3,riwnc5,riwnc15,riwnc16,
riwnc4,riwnc5,
riww9,riww10,riww11,
riww6,riww7,riww8,
riwsc12,riwsc13,riwsc14,
riwsc17,riwsc18,
riwe20,riwe21,
riwe22,riwe23,
riwe24,riwe25,
```

```
rtonw1,rtonw24,
rtosw22,rtosw23,
rtosw19,rtosw20,
rtoe6,rtoe8,
rtshnw1,rtshnw24,
rtshsw22,rtshsw23,
rtshsw19,rtshsw20,
rtshe6,rtshe8,
rtsnw1,rtsnw24,
rtssw22,rtssw23,
rtssw19,rtssw20,
rtse6,rtse8,
rtwnw1,rtwnw24,
rtwsw22,rtwsw23,
rtwsw19,rtwsw20,
rtwe6,rtwe8,
rishnc,rishj,
ris310,ris9,ris11,ris12,ris13,ris14,ris20,ris26,
riwnc45,riwnc1231516,
riww678,riww91011,
riwsc1718,riwsc121314,
riwe2425,riwe2223,riwe202126,riwe19,
rionc45,rionc1231516,
riow678,riow91011,
riosc1718,riosc121314,
rioe2425,rioe2223,rioe202126,rioe19,
rtosw1920,rtosw2223,
rtshsw1920,rtshsw2223,
rtssw1920,rtssw2223,
rtwsw1920,rtwsw2223,
riww,riwnc,
riow,rionc,
rtosw,rtonw,
rtshsw,rtshnw,
rtssw,rtsnw,
rtwsw,rtwnw,
rioe,riosc,rionwc,
riwe,riwsc,riwnwc,
rtoe,rtow,
rtshe,rtshw,
rtse,rtsw,
rtwe,rtww,
rinw,rins,
rino,rinsh,
rthw,rths,
rtho,rthsh,
rim,rinm,
rtm,rtnm,
ri,rt,

/* the flows */
zishn3, zishn12,zishn13,zishn14,zishn16,zishn17,
zionc1,zionc2,zionc3,zionc5,zionc15,zionc16,
zionc4,zionc5,
ziow9,ziow10,ziow11,
ziow6,ziow7,ziow8,
ziosc12,ziosc13,ziosc14,
ziosc17,ziosc18,
zioe20,zioe21,
zioe22,zioe23,
zioe24,zioe25,
zis3,zis10,
ziwnc1,ziwnc2,ziwnc3,ziwnc5,ziwnc15,ziwnc16,
ziwnc4,ziwnc5,
ziww9,ziww10,ziww11,
ziww6,ziww7,ziww8,
ziwsc12,ziwsc13,ziwsc14,
ziwsc17,ziwsc18,
ziwe20,ziwe21,
ziwe22,ziwe23,
ziwe24,ziwe25,
ztonw1,ztonw24,
ztosw22,ztosw23,
ztosw19,ztosw20,
ztoe6,ztoe8,
ztshnw1,ztshnw24,
ztshsw22,ztshsw23,
```

```
  ztshsw19,ztshsw20,
  ztshe6,ztshe8,
  ztsnw1,ztsnw24,
  ztssw22,ztssw23,
  ztssw19,ztssw20,
  ztse6,ztse8,
  ztwnw1,ztwnw24,
  ztwsw22,ztwsw23,
  ztwsw19,ztwsw20,
  ztwe6,ztwe8,
  zishnc,
  zis310,
  ziwnc45,ziwnc1231516,
  ziww678,ziww91011,
  ziwsc1718,ziwsc121314,
  ziwe2425,ziwe2223,ziwe202126,ziwe19,
  zionc45,zionc1231516,
  ziow678,ziow91011,
  ziosc1718,ziosc121314,
  zioe2425,zioe2223,zioe202126,zioe19,
  ztosw1920,ztosw2223,
  ztshsw1920,ztshsw2223,
  ztssw1920,ztssw2223,
  ztwsw1920,ztwsw2223,
  ziww,ziwnc,
  ziow,zionc,
  ztosw,ztonw,
  ztshsw,ztshnw,
  ztssw,ztsnw,
  ztwsw,ztwnw,
  zioe,ziosc,zionwc,
  ziwe,ziwsc,ziwnwc,
  ztoe,ztow,
  ztshe,ztshw,
  ztse,ztsw,
  ztwe,ztww,
  ziw,zis,
  zio,zish,
  ztw,zts,
  zto,ztsh,
  zim,zinm,
  ztm,ztnm,
  zi,zt,


  /* the parts of trips achieved with public transports for each purpose*/
  /* for inward trips*/
  ptiforwork,ptiforschool,ptiforother,ptiforshop,
  /* for through trips*/
  pttforwork,pttforschool,pttforother,pttforshop,

  /*public transports demands*/
  dpt[27],dptoff[27],dpteven[27],

  /*publlic transports replicated flows*/
  xpt[27],ypt[27],zpt[27];
```

4.tree (treeeven and treeoff are similar)

```
/* compute CES nested structure for each origin */


for ( destination = 1; destination <= 26; destination++ )
{
  /* take car flows from ATES */
  xcar[destination] = demand[origin][destination];

  /* price = travel time computed by ATES */
  pcar[destination] = prix[origin][destination];

  /* for buses we add a fare (ticket price ? !! pb of VOT) */
  ppt[destination] = pcar[destination] + fare;


  /*share the flows between the trip purposes*/
```

```
  /* inward or through */
  if ((destination == 1) ||
      (destination == 4) ||
      (destination == 7) ||
      (destination == 8) ||
      (destination == 19) ||
      (destination == 20) ||
      (destination == 22) ||
      (destination == 23) ||
      (destination == 24))

    {
      xinw [destination] = xcar[destination] * 0.5;
      xthrough [destination] = xcar[destination] - xinw[destination];
    }
  else
    {
      xinw[destination] = xcar[destination];
      xthrough[destination] = 0.0;
    };

  /* for the different purposes, we suppose different parts for pt*/
  ptiforwork = 0.3;
  ptiforschool = 0.4;
  ptiforother = 0.15;
  ptiforshop = 0.15;
  pttforwork = 0.4;
  pttforschool = 0.45;
  pttforother = 0.1;
  pttforshop = 0.05;

  /* inward work */
  xcarinww[destination] = xinw[destination]*0.3;
  /* inward school */
  xcarinws[destination] = xinw[destination]*0.3;
  /* inward shopping */
  xcarinwsh[destination] = xinw[destination]*0.2;
  /* inward other */
  xcarinwo[destination] = xinw[destination]*0.2;
  /* same with pt*/
  xptinww[destination] = morptiw[origin][destination];
  xptinws[destination] = morptis[origin][destination];
  xptinwsh[destination] =morptish[origin][destination];
  xptinwo[destination] = morptio[origin][destination];
  /* through work */
  xcarthw[destination] = xthrough[destination]*0.3;
  xptthw[destination] = morpttw[origin][destination];
  /* through school */
  xcarths[destination] = xthrough[destination]*0.3;
  xptths[destination] = morptts[origin][destination];
  /* through shopping */
  xcarthsh[destination] = xthrough[destination]*0.2;
  xptthsh[destination] = morpttsh[origin][destination];
  /* through other */
  xcartho[destination] = xthrough[destination]*0.2;
  xpttho[destination] = morptto[origin][destination];

  dpt[destination]=xptinww[destination]+xptinws[destination]+
    xptinwsh[destination]+xptinwo[destination]+xptthw[destination]+
    xptths[destination]+xptthsh[destination]+xpttho[destination];

}

/* level 0 (car or pt)*/

#include "carorpt.c"

/* level 1 (lower)*/

/* inward shopping trips in Namur centre*/

six_system
(&aishn3,"aishn3",&aishn12,"aishn12",&aishn13,"aishn13",&aishn14,"aishn14",&aishn16,"a
ishn16",&aishn17,"aishn17",
            xinwsh[3],xinwsh[12],xinwsh[13],xinwsh[14],xinwsh[16],xinwsh[17],
            pish[3],pish[12],pish[13],pish[14],pish[16],pish[17],
            gammaishnc );
```

```
/* inward other trips north-centre 1 2 3 15 16 */

five_system (&aionc1,"aionc1", &aionc2,"aionc2", &aionc3,"aionc3", &aionc15,"aionc15",
&aionc16,"aionc16",
            xinwo[1], xinwo[2], xinwo[3], xinwo[15], xinwo[16],
            pio[1], pio[2], pio[3], pio[15], pio[16],
            gammaionc );

/* inward other trips north-centre 4 5*/

two_system (&aionc4,"aionc4",&aionc5,"aionc5",
            xinwo[4], xinwo[5],
            pio[4], pio[5],
            gammaionc);

/* inward other trips west 9 10 11 */

three_system ( &aiow9,"aiow9", &aiow10,"aiow10", &aiow11,"aiow11",
               xinwo[9], xinwo[10], xinwo[11],
               pio[9], pio[10], pio[11],
               gammaiow );

/* inward other trips west 6 7 8 */

three_system ( &aiow6,"aiow6", &aiow7,"aiow7", &aiow8,"aiow8",
               xinwo[6], xinwo[7], xinwo[8],
               pio[6], pio[7], pio[8],
               gammaiow );

/* inward other trips south-centre 12 13 14 */

three_system ( &aiosc12,"aiosc12", &aiosc13,"aiosc13", &aiosc14,"aiosc14",
               xinwo[12], xinwo[13], xinwo[14],
               pio[12], pio[13], pio[14],
               gammaiosc );

/* inward other trips south-centre 17 18 */

two_system (&aiosc17,"aiosc17", &aiosc18,"aiosc18",
            xinwo[17], xinwo[18],
            pio[17], pio[18],
            gammaiosc);

/* inward other trips east 20 21 26*/

three_system (&aioe20,"aioe20",&aioe21,"aioe21",&aioe26,"aioe26",
              xinwo[20], xinwo[21], xinwo[26],
              pio[20], pio[21], pio[26],
              gammaioe);

/* inward other trips east 22 23 */

two_system (&aioe22,"aioe22",&aioe23,"aioe23",
            xinwo[22], xinwo[23],
            pio[22] ,pio[23] ,
            gammaioe);

/* inward other trips east 24 25 */

two_system (&aioe24,"aioe24",&aioe25,"aioe25",
            xinwo[24], xinwo[25],
            pio[24] ,pio[25] ,gammaioe);

/* inward school trips 3 10 */

two_system (&ais3,"ais3",&ais10,"ais10",
            xinws[3], xinws[10],
            pis[3] ,pis[10] ,
            gammais);

/* inward work trips north-centre 1 2 3 15 16 */

five_system (&aiwnc1,"aiwnc1", &aiwnc2,"aiwnc2", &aiwnc3,"aiwnc3", &aiwnc15,"aiwnc15",
&aiwnc16,"aiwnc16",
            xinww[1], xinww[2], xinww[3], xinww[15], xinww[16],
            piw[1], piw[2], piw[3], piw[15], piw[16],
```

```
            gammaiwnc );

/* inward work trips north-centre 4 5*/

two_system (&aiwnc4,"aiwnc4",&aiwnc5,"aiwnc5",
            xinww[4], xinww[5],
            piw[4] ,piw[5] ,
            gammaiwnc);

/* inward work trips west 9 10 11 */

three_system ( &aiww9,"aiww9", &aiww10,"aiww10", &aiww11,"aiww11",
               xinww[9], xinww[10], xinww[11],
               piw[9], piw[10], piw[11],
               gammaiww );

/* inward work trips west 6 7 8 */

three_system ( &aiww6,"aiww6", &aiww7,"aiww7", &aiww8,"aiww8",
               xinww[6], xinww[7], xinww[8],
               piw[6], piw[7], piw[8],
               gammaiww );

/* inward work trips south-centre 12 13 14 */

three_system ( &aiwsc12,"aiwsc12", &aiwsc13,"aiwsc13", &aiwsc14,"aiwsc14",
               xinww[12], xinww[13], xinww[14],
               piw[12], piw[13], piw[14],
               gammaiwsc );

/* inward work trips south-centre 17 18 */

two_system (&aiwsc17,"aiwsc17",&aiwsc18,"aiwsc18",
            xinww[17], xinww[18],
            piw[17] ,piw[18] ,
            gammaiwsc);

/* inward work trips east 20 21 26 */

three_system (&aiwe20,"aiwe20",&aiwe21,"aiwe21", &aiwe26,"aiwe26",
              xinww[20], xinww[21], xinww[26],
              piw[20], piw[21], piw[26],
              gammaiwe);

/* inward work trips east 22 23 */

two_system (&aiwe22,"aiwe22",&aiwe23,"aiwe23",
            xinww[22], xinww[23],
            piw[22] ,piw[23] ,
            gammaiwe);

/* inward work trips east 24 25 */

two_system (&aiwe24,"aiwe24",&aiwe25,"aiwe25",
            xinww[24], xinww[25],
            piw[24] ,piw[25] ,
            gammaiwe);

/* through other trips north-west 1 24 */

two_system (&atonw1,"atonw1",&atonw24,"atonw24",
            xtho[1], xtho[24],
            pto[1] ,pto[24] ,
            gammatonw);

/* through other trips south-west 22 23 */

two_system (&atosw22,"atosw22",&atosw23,"atosw23",
            xtho[22], xtho[23],
            pto[22] ,pto[23] ,
            gammatosw);

/* through other trips south-west 19 20 */

two_system (&atosw19,"atosw19",&atosw20,"atosw20",
            xtho[19], xtho[20],
            pto[19] ,pto[20] ,
```

```
            gammatosw);

/* through other trips east 7 8 */

two_system (&atoe7,"atoe7",&atoe8,"atoe8",
            xtho[7], xtho[8],
            pto[7] ,pto[8] ,
            gammatoe);

/* through shopping trips north-west 1 24 */

two_system (&atshnw1,"atshnw1",&atshnw24,"atshnw24",
            xthsh[1], xthsh[24],
            ptsh[1] ,ptsh[24] ,
            gammatshnw);

/* through shopping trips south-west 22 23 */

two_system (&atshsw22,"atshsw22",&atshsw23,"atshsw23",
            xthsh[22], xthsh[23],
            ptsh[22] ,ptsh[23] ,
            gammatshsw);

/* through shopping trips south-west 19 20 */

two_system (&atshsw19,"atshsw19",&atshsw20,"atshsw20",
            xthsh[19], xthsh[20],
            ptsh[19] ,ptsh[20] ,
            gammatshsw);

/* through shopping trips east 7 8 */

two_system (&atshe7,"atshe7",&atshe8,"atshe8",
            xthsh[7], xthsh[8],
            ptsh[7] ,ptsh[8] ,
            gammatshe);

/* through school trips north-west 1 24 */

two_system (&atsnw1,"atsnw1",&atsnw24,"atsnw24",
            xths[1], xths[24],
            pts[1] ,pts[24] ,
            gammatsnw);

/* through school trips south-west 22 23 */

two_system (&atssw22,"atssw22",&atssw23,"atssw23",
            xths[22], xths[23],
            pts[22] ,pts[23] ,
            gammatssw);

/* through school trips south-west 19 20 */

two_system (&atssw19,"atssw19",&atssw20,"atssw20",
            xths[19], xths[20],
            pts[19] ,pts[20] ,
            gammatssw);

/* through school trips east 7 8 */

two_system (&atse7,"atse7",&atse8,"atse8",
            xths[7], xths[8],
            pts[7] ,pts[8] ,
            gammatse);

/* through work trips north-west 1 24 */

two_system (&atwnw1,"atwnw1",&atwnw24,"atwnw24",
            xthw[1], xthw[24],
            ptw[1] ,ptw[24] ,
            gammatwnw);

/* through work trips south-west 22 23 */

two_system (&atwsw22,"atwsw22",&atwsw23,"atwsw23",
            xthw[22], xthw[23],
            ptw[22] ,ptw[23] ,
```

```
           gammatwsw);

/* through work trips south-west 19 20 */

two_system (&atwsw19,"atwsw19",&atwsw20,"atwsw20",
           xthw[19], xthw[20],
           ptw[19] ,ptw[20] ,
           gammatwsw);

/* through work trips east 7 8 */

two_system (&atwe7,"atwe7",&atwe8,"atwe8",
           xthw[7], xthw[8],
           ptw[7] ,ptw[8] ,
           gammatwe);

/* level 2 */

/* inward shopping trips Namur Centre or Jambes (20) */

pishnc = pow ((aishn3 * pow (pish[3] , (1 - gammaishnc)) +
              aishn12 * pow (pish[12] , (1 - gammaishnc)) +
              aishn13 * pow (pish[13] , (1 - gammaishnc)) +
              aishn14 * pow (pish[14] , (1 - gammaishnc)) +
              aishn16 * pow (pish[16] , (1 - gammaishnc)) +
              aishn17 * pow (pish[17] , (1 - gammaishnc))  )
             , ( 1.0 / (1 - gammaishnc ))) ;

xishnc = (pish[3]*xinwsh[3] + pish[12]*xinwsh[12] + pish[13]*xinwsh[13] +
pish[14]*xinwsh[14] + pish[16]*xinwsh[16] + pish[17]*xinwsh[17] )/pishnc;

two_system (&aishnc,"aishnc",&aishj,"aishj",
           xishnc, xinwsh[20],
           pishnc ,pish[20] ,gammaish);

/* inward school trips */

pis310 = pow ((ais3 * pow (pis[3] , (1 -gammais)) +
              ais10 * pow (pis[10] , (1 - gammais)) ),
             (1.0 / (1 - gammais))) ;

xis310 = (pis[3]*xinws[3] + pis[10]*xinws[10] )/pis310;

eight_system (&ais310,"ais310", &ais9,"ais9", &ais11,"ais11", &ais12,"ais12",
&ais13,"ais13", &ais14,"ais14", &ais20,"ais20", &ais26,"ais26",
             xis310, xinws[9], xinws[11], xinws[12], xinws[13], xinws[14], xinws[20],
xinws[26],
             pis310, pis[9], pis[11], pis[12], pis[13], pis[14], pis[20], pis[26],
             gammais2 ) ;

/* inward work trips north-centre */

piwnc45 = pow ((aiwnc4 * pow (piw[4] , (1 - gammaiwnc)) +
               aiwnc5 * pow (piw[5] , (1 - gammaiwnc))  )
              , ( 1.0 / (1 - gammaiwnc ))) ;
piwnc1231516 = pow ((aiwnc1 * pow (piw[1] , (1 - gammaiwnc)) +
                    aiwnc2 * pow (piw[2] , (1 - gammaiwnc)) +
                    aiwnc3 * pow (piw[3] , (1 - gammaiwnc)) +
                    aiwnc15 * pow (piw[15] , (1 - gammaiwnc)) +
                    aiwnc16 * pow (piw[16] , (1 - gammaiwnc))  )
                   , ( 1.0 / (1 - gammaiwnc ))) ;

xiwnc45 = (piw[4]*xinww[4] + piw[5]*xinww[5] )/piwnc45;
xiwnc1231516 = (piw[1]*xinww[1] + piw[2]*xinww[2] + piw[3]*xinww[3] +
piw[15]*xinww[15] + piw[16]*xinww[16] )/piwnc1231516;

two_system (&aiwnc45,"aiwnc45",&aiwnc1231516,"aiwnc1231516",
           xiwnc45, xiwnc1231516,
           piwnc45 ,piwnc1231516 ,
           gammaiw2);

/* inward work trips west */

piww678 = pow (( aiww6 * pow (piw[6] , (1 - gammaiww)) +
                aiww7 * pow (piw[7] , (1 - gammaiww)) +
                aiww8 * pow (piw[8] , (1 - gammaiww))   )
              , (1.0 / (1 - gammaiww ))) ;
```

```
piww91011 = pow (( aiww9 * pow (piw[9] , (1 - gammaiww)) +
                  aiww10 * pow (piw[10] , (1 - gammaiww)) +
                  aiww11 * pow (piw[11] , (1 - gammaiww))   )
                , (1.0 / (1 - gammaiww ))) ;

xiww678 = (piw[6]*xinww[6] + piw[7]*xinww[7] + piw[8]*xinww[8] )/piww678;
xiww91011 = (piw[9]*xinww[9] + piw[10]*xinww[10] + piw[11]*xinww[11] )/piww91011;

two_system (&aiww678,"aiww678",&aiww91011,"aiww91011",
           xiww678, xiww91011,
           piww678 ,piww91011 ,
           gammaiw2);

/* inward work trips south centre */

piwsc1718 = pow (( aiwsc17 * pow (piw[17] , (1 - gammaiwsc)) +
                  aiwsc18 * pow (piw[18] , (1 - gammaiwsc))   )
                , (1.0 / (1 - gammaiwsc ))) ;

piwsc121314 = pow (( aiwsc12 * pow (piw[12] , (1 - gammaiwsc)) +
                    aiwsc13 * pow (piw[13] , (1 - gammaiwsc)) +
                    aiwsc14 * pow (piw[14] , (1 - gammaiwsc))   )
                  , (1.0 / (1 - gammaiwsc ))) ;

xiwsc1718 = (piw[17]*xinww[17] + piw[18]*xinww[18] )/piwsc1718;
xiwsc121314 = (piw[12]*xinww[12] + piw[13]*xinww[13] + piw[14]*xinww[14]
)/piwsc121314;

two_system (&aiwsc1718,"aiwsc1718",&aiwsc121314,"aiwsc121314",
           xiwsc1718, xiwsc121314,
           piwsc1718 ,piwsc121314 ,
           gammaiw3);

/* inward work trips east */

piwe2425 = pow ((aiwe24 * pow (piw[24] , (1 - gammaiwe)) +
                aiwe25 * pow (piw[25] , (1 - gammaiwe))  )
               , (1.0 / (1 - gammaiwe ))) ;

piwe2223 = pow ((aiwe22 * pow (piw[22] , (1 - gammaiwe)) +
                aiwe23 * pow (piw[23] , (1 - gammaiwe))  )
               , (1.0 / (1 - gammaiwe ))) ;

piwe202126 = pow ((aiwe20 * pow (piw[20] , (1 - gammaiwe)) +
                  aiwe21 * pow (piw[21] , (1 - gammaiwe)) +
                  aiwe26 * pow (piw[26] , (1 - gammaiwe))  )
                 , (1.0 / (1 - gammaiwe ))) ;

xiwe2425 = (piw[24]*xinww[24] + piw[25]*xinww[25] )/piwe2425;
xiwe2223 = (piw[22]*xinww[22] + piw[23]*xinww[23] )/piwe2223;
xiwe202126 = (piw[20]*xinww[20] + piw[21]*xinww[21]+ piw[26]*xinww[26] )/piwe202126;

four_system (&aiwe2425,"aiwe2425", &aiwe2223,"aiwe2223", &aiwe202126,"aiwe202126",
&aiwe19,"aiwe19",
             xiwe2425, xiwe2223, xiwe202126, xinww[19],
            piwe2425, piwe2223, piwe202126, piw[19],
             gammaiw3 ) ;

/* inward other trips north-centre */

pionc45 = pow ((aionc4 * pow (pio[4] , (1 - gammaionc)) +
               aionc5 * pow (pio[5] , (1 - gammaionc))  )
              , ( 1.0 / (1 - gammaionc ))) ;

pionc1231516 = pow ((aionc1 * pow (pio[1] , (1 - gammaionc)) +
                     aionc2 * pow (pio[2] , (1 - gammaionc)) +
                    aionc3 * pow (pio[3] , (1 - gammaionc)) +
                    aionc15 * pow (pio[15] , (1 - gammaionc)) +
                    aionc16 * pow (pio[16] , (1 - gammaionc))  )
                  , ( 1.0 / (1 - gammaionc ))) ;

xionc45 = (pio[4]*xinwo[4] + pio[5]*xinwo[5] )/pionc45;
xionc1231516 = (pio[1]*xinwo[1] + pio[2]*xinwo[2] + pio[3]*xinwo[3] +
pio[15]*xinwo[15] + pio[16]*xinwo[16] )/pionc1231516;

two_system (&aionc45,"aionc45",&aionc1231516,"aionc1231516",
```

```
            xionc45, xionc1231516,
            pionc45 ,pionc1231516 ,
            gammaio2);

/* inward other trips west */

piow678 = pow (( aiow6 * pow (pio[6] , (1 - gammaiow)) +
                 aiow7 * pow (pio[7] , (1 - gammaiow)) +
                 aiow8 * pow (pio[8] , (1 - gammaiow))   )
               , (1.0 / (1 - gammaiow ))) ;

piow91011 = pow (( aiow9 * pow (pio[9] , (1 - gammaiow)) +
                   aiow10 * pow (pio[10] , (1 - gammaiow)) +
                   aiow11 * pow (pio[11] , (1 - gammaiow))   )
                 , (1.0 / (1 - gammaiow ))) ;


xiow678 = (pio[6]*xinwo[6] + pio[7]*xinwo[7] + pio[8]*xinwo[8] )/piow678;
xiow91011 = (pio[9]*xinwo[9] + pio[10]*xinwo[10] + pio[11]*xinwo[11] )/piow91011;

two_system (&aiow678,"aiow678",&aiow91011,"aiow91011",
            xiow678, xiow91011,
            piow678 ,piow91011 ,
            gammaio2);

/* inward other trips south centre */

piosc1718 = pow (( aiosc17 * pow (pio[17] , (1 - gammaiosc)) +
                   aiosc18 * pow (pio[18] , (1 - gammaiosc))   )
                 , (1.0 / (1 - gammaiosc ))) ;

piosc121314 = pow (( aiosc12 * pow (pio[12] , (1 - gammaiosc)) +
                     aiosc13 * pow (pio[13] , (1 - gammaiosc)) +
                     aiosc14 * pow (pio[14] , (1 - gammaiosc))   )
                   , (1.0 / (1 - gammaiosc ))) ;


xiosc1718 = (pio[17]*xinwo[17] + pio[18]*xinwo[18] )/piosc1718;
xiosc121314 = (pio[12]*xinwo[12] + pio[13]*xinwo[13] + pio[14]*xinwo[14]
)/piosc121314;

two_system (&aiosc1718,"aiosc1718",&aiosc121314,"aiosc121314",
           xiosc1718, xiosc121314,
           piosc1718 ,piosc121314 ,
           gammaio3);

/* inward other trips east */

pioe2425 = pow ((aioe24 * pow (pio[24] , (1 - gammaioe)) +
                 aioe25 * pow (pio[25] , (1 - gammaioe))  )
               , (1.0 / (1 - gammaioe ))) ;

pioe2223 = pow ((aioe22 * pow (pio[22] , (1 - gammaioe)) +
                 aioe23 * pow (pio[23] , (1 - gammaioe))  )
               , (1.0 / (1 - gammaioe ))) ;

pioe202126 = pow ((aioe20 * pow (pio[20] , (1 - gammaioe)) +
                   aioe21 * pow (pio[21] , (1 - gammaioe)) +
                    aioe26 * pow (pio[26] , (1 - gammaioe))  )
                  , (1.0 / (1 - gammaioe ))) ;


xioe2425 = (pio[24]*xinwo[24] + pio[25]*xinwo[25] )/pioe2425;
xioe2223 = (pio[22]*xinwo[22] + pio[23]*xinwo[23] )/pioe2223;
xioe202126 = (pio[20]*xinwo[20] + pio[21]*xinwo[21]+ pio[26]*xinwo[26] )/pioe202126;

four_system (&aioe2425,"aioe2425", &aioe2223,"aioe2223", &aioe202126,"aioe202126",
&aioe19,"aioe19",
            xioe2425, xioe2223, xioe202126, xinwo[19],
            pioe2425, pioe2223, pioe202126, pio[19],
            gammaio3 ) ;

/* through other trips south west */

ptosw1920 = pow ((atosw19 * pow (pto[19] , (1 - gammatosw)) +
                  atosw20 * pow (pto[20] , (1 - gammatosw))  )
                 , (1.0 / (1 - gammatssw ))) ;
```

```
ptosw2223 = pow ((atosw22 * pow (pto[22] , (1 - gammatosw)) +
                 atosw23 * pow (pto[23] , (1 - gammatosw))  )
                , (1.0 / (1 - gammatosw ))) ;

xtosw1920 = (pto[19]*xtho[19] + pto[20]*xtho[20] )/ptosw1920;
xtosw2223 = (pto[22]*xtho[22] + pto[23]*xtho[23] )/ptosw2223;

two_system (&atosw1920,"atosw1920",&atosw2223,"atosw2223",
           xtosw1920, xtosw2223,
           ptosw1920 ,ptosw2223 ,
           gammatosw2);

/* through shopping trips south west */

ptshsw1920 = pow ((atshsw19 * pow (ptsh[19] , (1 - gammatshsw)) +
                  atshsw20 * pow (ptsh[20] , (1 - gammatshsw))  )
                 , (1.0 / (1 - gammatshsw ))) ;

ptshsw2223 = pow ((atshsw22 * pow (ptsh[22] , (1 - gammatshsw)) +
                  atshsw23 * pow (ptsh[23] , (1 - gammatshsw))  )
                 , (1.0 / (1 - gammatshsw ))) ;

xtshsw1920 = (ptsh[19]*xthsh[19] + ptsh[20]*xthsh[20] )/ptshsw1920;
xtshsw2223 = (ptsh[22]*xthsh[22] + ptsh[23]*xthsh[23] )/ptshsw2223;

two_system (&atshsw1920,"atshsw1920",&atshsw2223,"atshsw2223",
           xtshsw1920, xtshsw2223,
           ptshsw1920 ,ptshsw2223 ,
           gammatshsw2);

/* through school trips south west */

ptssw1920 = pow ((atssw19 * pow (pts[19] , (1 - gammatssw)) +
                 atssw20 * pow (pts[20] , (1 - gammatssw))  )
                , (1.0 / (1 - gammatssw ))) ;

ptssw2223 = pow ((atssw22 * pow (pts[22] , (1 - gammatssw)) +
                 atssw23 * pow (pts[23] , (1 - gammatssw))  )
                , (1.0 / (1 - gammatssw ))) ;

xtssw1920 = (pts[19]*xths[19] + pts[20]*xths[20] )/ptssw1920;
xtssw2223 = (pts[22]*xths[22] + pts[23]*xths[23] )/ptssw2223;

two_system (&atssw1920,"atssw1920",&atssw2223,"atssw2223",
           xtssw1920, xtssw2223,
           ptssw1920 ,ptssw2223 ,
           gammatssw2);

/* through work trips south west */

ptwsw1920 = pow ((atwsw19 * pow (ptw[19] , (1 - gammatwsw)) +
                 atwsw20 * pow (ptw[20] , (1 - gammatwsw))  )
                , (1.0 / (1 - gammatwsw ))) ;

ptwsw2223 = pow ((atwsw22 * pow (ptw[22] , (1 - gammatwsw)) +
                 atwsw23 * pow (ptw[23] , (1 - gammatwsw))  )
                , (1.0 / (1 - gammatwsw ))) ;


xtwsw1920 = (ptw[19]*xthw[19] + ptw[20]*xthw[20] )/ptwsw1920;
xtwsw2223 = (ptw[22]*xthw[22] + ptw[23]*xthw[23] )/ptwsw2223;

two_system (&atwsw1920,"atwsw1920",&atwsw2223,"atwsw2223",
           xtwsw1920, xtwsw2223,
           ptwsw1920 ,ptwsw2223 ,
           gammatwsw2);

/* level 3 */

/* inward work trips NWC */

piww = pow ((aiww678 * pow (piww678 , (1 - gammaiw2)) +
            aiww91011 * pow (piww91011 , (1 - gammaiw2)) )
           , (1.0 / (1 - gammaiw2))) ;

piwnc = pow ((aiwnc45 * pow (piwnc45 , (1 - gammaiw2)) +
```

```
              aiwnc1231516 * pow (piwnc1231516 , (1 - gammaiw2)) )
             , (1.0 / (1 - gammaiw2))) ;


xiww = (xiww678* piww678 + xiww91011* piww91011 )/piww;
xiwnc = (xiwnc45* piwnc45 + xiwnc1231516* piwnc1231516 )/piwnc;

two_system (&aiww,"aiww",&aiwnc,"aiwnc",
            xiww, xiwnc,
            piww ,piwnc ,
            gammaiwnwc);

/* inward other trips NWC */

piow = pow ((aiow678 * pow (piow678 , (1 - gammaio2)) +
             aiow91011 * pow (piow91011 , (1 - gammaio2)) )
            , (1.0 / (1 - gammaio2))) ;

pionc = pow ((aionc45 * pow (pionc45 , (1 - gammaio2)) +
              aionc1231516 * pow (pionc1231516 , (1 - gammaio2)) )
             , (1.0 / (1 - gammaio2))) ;

xiow = (xiow678* piow678 + xiow91011* piow91011 )/piow;
xionc = (xionc45*pionc45 + xionc1231516* pionc1231516 )/pionc;

two_system (&aiow,"aiow",&aionc,"aionc",
            xiow, xionc,
            piow ,pionc ,
            gammaionwc);

/* through other trips west */

ptosw = pow ((atosw1920 * pow (ptosw1920 , (1 - gammatosw2)) +
              atosw2223 * pow (ptosw2223 , (1 - gammatosw2))  )
             , (1.0 / (1 - gammatosw2))) ;

ptonw = pow ((atonw1 * pow (pto[1] , (1 - gammatonw)) +
              atonw24 * pow (pto[24] , (1 - gammatonw)) )
             , (1.0 / (1 - gammatonw))) ;

xtosw = (xtosw1920*ptosw1920 + xtosw2223* ptosw2223 )/ptosw;
xtonw = (xtho[1]*pto[1] + xtho[24]* pto[24] )/ptonw;

two_system (&atosw,"atosw",&atonw,"atonw",
            xtosw, xtonw,
            ptosw ,ptonw ,
            gammatow);

/* through shopping trips west */

ptshsw = pow ((atshsw1920 * pow (ptshsw1920 , (1 - gammatshsw2)) +
               atshsw2223 * pow (ptshsw2223 , (1 - gammatshsw2))  )
              , (1.0 / (1 - gammatshsw2))) ;

ptshnw = pow ((atshnw1 * pow (ptsh[1] , (1 - gammatshnw)) +
               atshnw24 * pow (ptsh[24] , (1 - gammatshnw)) )
              , (1.0 / (1 - gammatshnw))) ;

xtshsw = (xtshsw1920*ptshsw1920 + xtshsw2223* ptshsw2223 )/ptshsw;
xtshnw = (xthsh[1]*ptsh[1] + xthsh[24]* ptsh[24] )/ptshnw;

two_system (&atshsw,"atshsw",&atshnw,"atshnw",
            xtshsw, xtshnw,
            ptshsw ,ptshnw ,
            gammatshw);

/* through school trips west */

ptssw = pow ((atssw1920 * pow (ptssw1920 , (1 - gammatssw2)) +
              atssw2223 * pow (ptssw2223 , (1 - gammatssw2))  )
             , (1.0 / (1 - gammatssw2))) ;

ptsnw = pow ((atsnw1 * pow (pts[1] , (1 - gammatsnw)) +
              atsnw24 * pow (pts[24] , (1 - gammatsnw)) )
             , (1.0 / (1 - gammatsnw))) ;

xtssw = (xtssw1920*ptssw1920 + xtssw2223* ptssw2223 )/ptssw;
```

```
xtsnw = (xths[1]*pts[1] + xths[24]* pts[24] )/ptsnw;

two_system (&atssw,"atssw",&atsnw,"atsnw",
           xtssw, xtsnw,
           ptssw ,ptsnw ,
           gammatsw);

/* through work trips west */

ptwsw = pow ((atwsw1920 * pow (ptwsw1920 , (1 - gammatwsw2)) +
             atwsw2223 * pow (ptwsw2223 , (1 - gammatwsw2))  )
            , (1.0 / (1 - gammatwsw2))) ;

ptwnw = pow ((atwnw1 * pow (ptw[1] , (1 - gammatwnw)) +
             atwnw24 * pow (ptw[24], (1 - gammatwnw)) )
            , (1.0 / (1 - gammatwnw))) ;

xtwsw = (xtwsw1920*ptwsw1920 + xtwsw2223* ptwsw2223 )/ptwsw;
xtwnw = (xthw[1]*ptw[1] + xthw[24]* ptw[24] )/ptwnw;


two_system (&atwsw,"atwsw",&atwnw,"atwnw",
           xtwsw, xtwnw,
           ptwsw ,ptwnw ,
           gammatww);

/*level 4*/

/* inward other trips */

pioe = pow ((aioe2425 * pow (pioe2425 , (1 - gammaio3)) +
            aioe2223 * pow (pioe2223 , (1 - gammaio3)) +
            aioe202126 * pow (pioe202126 , (1 - gammaio3)) +
            aioe19 * pow (pio[19] , (1 - gammaio3))        )
           , (1.0 / (1 - gammaio3))) ;

piosc = pow ((aiosc1718 * pow (piosc1718 , (1 - gammaio3)) +
             aiosc121314 * pow (piosc121314 , (1 - gammaio3)) )
            , (1.0 / (1 - gammaio3))) ;

pionwc = pow ((aiow * pow (piow , (1 - gammaionwc)) +
              aionc * pow (pionc , (1 - gammaionwc)) )
             , (1.0 / (1 - gammaionwc))) ;

xioe = (xioe2425*pioe2425 + xioe2223* pioe2223 + xioe202126* pioe202126 +
pio[19]*xinwo[19] )/pioe;
xiosc = (xiosc1718*piosc1718 + xiosc121314* piosc121314 )/piosc;
xionwc = (xiow*piow + xionc* pionc )/pionwc;

three_system (&aioe,"aioe", &aiosc,"aiosc", &aionwc,"aionwc",
             xioe, xiosc, xionwc,
             pioe, piosc, pionwc,
             gammaio ) ;

/* inward work trips */

piwe = pow ((aiwe2425 * pow (piwe2425 , (1 - gammaiw3)) +
            aiwe2223 * pow (piwe2223 , (1 - gammaiw3)) +
            aiwe202126 * pow (piwe202126 , (1 - gammaiw3)) +
            aiwe19 * pow (piw[19] , (1 - gammaiw3))        )
           , (1.0 / (1 - gammaiw3))) ;

piwsc = pow ((aiwsc1718 * pow (piwsc1718 , (1 - gammaiw3)) +
             aiwsc121314 * pow (piwsc121314 , (1 - gammaiw3)) )
            , (1.0 / (1 - gammaiw3))) ;

piwnwc = pow ((aiww * pow (piww , (1 - gammaiwnwc)) +
              aiwnc * pow (piwnc , (1 - gammaiwnwc)) )
             , (1.0 / (1 - gammaiwnwc))) ;

xiwe = (xiwe2425*piwe2425 + xiwe2223* piwe2223 + xiwe202126* piwe202126 +
piw[19]*xinww[19] )/piwe;
xiwsc = (xiwsc1718*piwsc1718 + xiwsc121314* piwsc121314 )/piwsc;
xiwnwc = (xiww*piww + xiwnc* piwnc )/piwnwc;

three_system (&aiwe,"aiwe", &aiwsc,"aiwsc", &aiwnwc,"aiwnwc",
             xiwe, xiwsc, xiwnwc,
```

```
             piwe, piwsc, piwnwc,
             gammaiw ) ;

/* through other trips */

ptoe = pow ((atoe7 * pow (pto[7] , (1 - gammatoe)) +
            atoe8 * pow (pto[8] , (1 - gammatoe))   )
           , (1.0 / (1 - gammatoe))) ;

ptow = pow ((atosw * pow (ptosw , (1 - gammatow)) +
            atonw * pow (ptonw , (1 - gammatow))   )
           , (1.0 / (1 - gammatow))) ;

xtoe = (pto[7]*xtho[7] + pto[8]*xtho[8] )/ptoe;
xtow = (xtosw*ptosw + xtonw* ptonw )/ptow;

three_system (&atoe,"atoe",&ato4, "ato4", &atow,"atow",
             xtoe, xtho[4], xtow,
             ptoe, pto[4], ptow ,
             gammato);

/* through shopping trips */

ptshe = pow ((atshe7 * pow (ptsh[7] , (1 - gammatshe)) +
             atshe8 * pow (ptsh[8] , (1 - gammatshe))   )
            , (1.0 / (1 - gammatshe))) ;

ptshw = pow ((atshsw * pow (ptshsw , (1 - gammatshw)) +
             atshnw * pow (ptshnw , (1 - gammatshw))   )
            , (1.0 / (1 - gammatshw))) ;

xtshe = (ptsh[7]*xthsh[7] + ptsh[8]*xthsh[8] )/ptshe;
xtshw = (xtshsw* ptshsw + xtshnw* ptshnw )/ptshw;

three_system (&atshe,"atshe",&atsh4, "atsh4", &atshw,"atshw",
             xtshe, xthsh[4], xtshw,
             ptshe, ptsh[4], ptshw,
             gammatsh);

/* through school trips */

ptse = pow ((atse7 * pow (pts[7] , (1 - gammatse)) +
            atse8 * pow (pts[8] , (1 - gammatse))   )
           , (1.0 / (1 - gammatse))) ;

ptsw = pow ((atssw * pow (ptssw , (1 - gammatsw)) +
            atsnw * pow (ptsnw , (1 - gammatsw))   )
           , (1.0 / (1 - gammatsw))) ;

xtse = (pts[7]*xths[7] + pts[8]*xths[8] )/ptse;
xtsw = (xtssw*ptssw + xtsnw* ptsnw )/ptsw;

three_system (&atse,"atse",&ats4, "ats4", &atsw,"atsw",
             xtse, xths[4], xtsw,
             ptse, pts[4], ptsw ,
             gammats);

/* through work trips */

ptwe = pow ((atwe7 * pow (ptw[7] , (1 - gammatwe)) +
            atwe8 * pow (ptw[8] , (1 - gammatwe))   )
           , (1.0 / (1 - gammatwe))) ;

ptww = pow ((atwsw * pow (ptwsw , (1 - gammatww)) +
            atwnw * pow (ptwnw , (1 - gammatww))   )
           , (1.0 / (1 - gammatww))) ;

xtwe = (ptw[7]*xthw[7] + ptw[8]*xthw[8] )/ptwe;
xtww = (xtwsw*ptwsw + xtwnw* ptwnw )/ptww;

three_system (&atwe,"atwe",&atw4, "atw4", &atww,"atww",
             xtwe, xthw[4], xtww,
             ptwe, ptw[4], ptww ,
             gammatw);

/* level 5 */
```

```
/* inward mandatory trips */

pinw = pow ((aiwe * pow (piwe , (1 - gammaiw)) +
            aiwsc * pow (piwsc , (1 - gammaiw)) +
            aiwnwc * pow (piwnwc , (1 - gammaiw)) )
           , (1.0 / (1 - gammaiw))) ;

pins = pow ((ais310 * pow (pis310 , (1 - gammais2)) +
            ais9 * pow (pis[9] , (1 - gammais2)) +
            ais11 * pow (pis[11] , (1 - gammais2)) +
            ais12 * pow (pis[12] , (1 - gammais2)) +
            ais13 * pow (pis[13] , (1 - gammais2)) +
            ais14 * pow (pis[14] , (1 - gammais2)) +
            ais20 * pow (pis[20] , (1 - gammais2)) +
            ais26 * pow (pis[26] , (1 - gammais2))   )
           , (1.0 / (1 - gammais2))) ;

xiw = (xiwe*piwe + xiwsc* piwsc + xiwnwc* piwnwc )/pinw;
xis = (xis310*pis310 + pis[9]*xinws[9] + pis[11]*xinws[11] + pis[12]*xinws[12] +
pis[13]*xinws[13] + pis[14]*xinws[14] + pis[20]*xinws[20] + pis[26]*xinws[26])/pins;

two_system (&aiw,"aiw",&ais,"ais",
            xiw, xis,
            pinw ,pins ,
            gammaim);

/* inward non mandatory trips */

pino = pow ((aioe * pow (pioe , (1 - gammaio)) +
             aiosc * pow (piosc , (1 - gammaio)) +
             aionwc * pow (pionwc , (1 - gammaio)) )
            ,(1.0 / (1 - gammaio))) ;

pinsh = pow ((aishnc * pow (pishnc , (1 - gammaish)) +
              aishj * pow (pish[20] , (1 - gammaish))     )
             , (1.0 / (1 - gammaish))) ;

xio = (xioe*pioe + xiosc* piosc + xionwc* pionwc )/pino;
xish = (xishnc*pishnc + xinwsh[20]* pish[20] )/pinsh;

two_system (&aio,"aio",&aish,"aish",
            xio, xish,
            pino ,pinsh ,
            gammainm);

/* through mandatory trips */

pthw = pow ((atwe * pow (ptwe , (1 - gammatw)) +
             atw4 * pow (ptw[4] , (1 - gammatw)) +
             atww * pow (ptww , (1 - gammatw))  )
            , (1.0 / (1 - gammatw))) ;

pths  = pow ((atse * pow (ptse , (1 - gammats)) +
              ats4 * pow (pts[4] , (1 - gammats)) +
             atsw * pow (ptsw , (1 - gammats))  )
             , (1.0 / (1 - gammats))) ;

xtw = (xtwe* ptwe + ptw[4]*xthw[4] + xtww*ptww )/pthw;
xts = (xtse*ptse + pts[4]*xths[4] + xtsw*ptsw )/pths;

two_system (&atw,"atw",&ats,"ats",
            xtw, xts,
            pthw ,pths ,
            gammatm);

/* through non mandatory trips */

ptho = pow ((atoe * pow (ptoe , (1 - gammato)) +
             ato4 * pow (pto[4] , (1 - gammato)) +
             atow * pow (ptow , (1 - gammato))  )
            , (1.0 / (1 - gammato))) ;

pthsh  = pow ((atshe * pow (ptshe , (1 - gammatsh)) +
               atsh4 * pow (ptsh[4] , (1 - gammatsh)) +
               atshw * pow (ptshw , (1 - gammatsh))  )
              , (1.0 / (1 - gammatsh))) ;
```

```
xto = (xtoe*ptoe + pto[4]*xtho[4] + xtow* ptow )/ptho;
xtsh = (xtshe*ptshe + ptsh[4]*xthsh[4] + xtshw* ptshw )/pthsh;

two_system (&ato,"ato",&atsh,"atsh",
           xto, xtsh,
           ptho ,pthsh ,
           gammatm);

/* level 6 */

/* inward trips */

pim = pow ((aiw * pow (pinw , (1 - gammaim)) +
           ais * pow (pins , (1 - gammaim))  )
          , (1.0 / (1 - gammaim))) ;

pinm = pow ((aio * pow (pino , (1 - gammainm)) +
            aish * pow (pinsh , (1 - gammainm)) )
           , (1.0 / (1 - gammainm))) ;

xim = (xiw*pinw + xis* pins )/pim;
xinm = (xio*pino + xish* pinsh )/pinm;

two_system (&aim,"aim",&ainm,"ainm",
           xim, xinm,
           pim ,pinm ,
           gammai);

/* through trips */

ptm = pow ((atw * pow (pthw , (1 - gammatm)) +
           ats * pow (pths , (1 - gammatm))  )
          , (1.0 / (1 - gammatm))) ;

ptnm = pow ((ato * pow (ptho , (1 - gammatnm)) +
            atsh * pow (pthsh , (1 - gammatnm)) )
           , (1.0 / (1 - gammatnm))) ;

xtm = (xtw*pthw + xts* pths )/ptm;
xtnm = (xto*ptho + xtsh* pthsh )/ptnm;

two_system (&atm,"atm",&atnm,"atnm",
           xtm, xtnm,
           ptm ,ptnm ,
           gammat);


/* level 7 */

/* trips */

pi = pow ((aim * pow (pim , (1 - gammai)) +
          ainm * pow (pinm , (1 - gammai))  )
         , (1.0 / (1 - gammai))) ;

pt = pow ((atm * pow (ptm , (1 - gammat)) +
          atnm * pow (ptnm , (1 - gammat)) )
         , (1.0 / (1 - gammat))) ;

xi = (xim*pim + xinm* pinm )/pi;
xt = (xtm*ptm + xtnm* ptnm )/pt;

two_system (&ai,"ai",&at,"at",
           xi, xt,
           pi ,pt ,
           lastgamma);
```

5.carorpt (carorptoff and carorpteven are similar)

```
/*inward work*/

two_system (&acariw24,"acariw24",&aptiw24,"aptiw24",
```

```
           xcarinww[24],xptinww[24],
           pcar[24],ppt[24],
           gammacarorpt                              );

piw[24] = pow ((acariw24 * pow (pcar[24],(1 - gammacarorpt)) +
              aptiw24 * pow (ppt[24],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[24] = (pcar[24]*xcarinww[24] + ppt[24]*xptinww[24])/piw[24];

two_system (&acariw25,"acariw25",&aptiw25,"aptiw25",
           xcarinww[25],xptinww[25],
           pcar[25],ppt[25],
           gammacarorpt                              );

piw[25] = pow ((acariw25 * pow (pcar[25],(1 - gammacarorpt)) +
              aptiw25 * pow (ppt[25],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[25] = (pcar[25]* xcarinww[25] + ppt[25] * xptinww[25])/piw[25];

two_system (&acariw22,"acariw22",&aptiw22,"aptiw22",
           xcarinww[22],xptinww[22],
           pcar[22],ppt[22],
           gammacarorpt                              );

piw[22] = pow ((acariw22 * pow (pcar[22],(1 - gammacarorpt)) +
              aptiw22 * pow (ppt[22],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[22] = (pcar[22]* xcarinww[22] + ppt[22] * xptinww[22])/piw[22];

two_system (&acariw23,"acariw23",&aptiw23,"aptiw23",
           xcarinww[23],xptinww[23],
           pcar[23],ppt[23],
           gammacarorpt                              );


piw[23] = pow ((acariw23 * pow (pcar[23],(1 - gammacarorpt)) +
              aptiw23 * pow (ppt[23],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[23] = (pcar[23]* xcarinww[23] + ppt[23] * xptinww[23])/piw[23];

two_system (&acariw20,"acariw20",&aptiw20,"aptiw20",
           xcarinww[20],xptinww[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

piw[20] = pow ((acariw20 * pow (pcar[20],(1 - gammacarorpt)) +
              aptiw20 * pow (ppt[20],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[20] = (pcar[20]* xcarinww[20] + ppt[20] * xptinww[20])/piw[20];

two_system (&acariw21,"acariw21",&aptiw21,"aptiw21",
           xcarinww[21],xptinww[21],
           pcar[21],ppt[21],
           gammacarorpt                              );

piw[21] = pow ((acariw21 * pow (pcar[21],(1 - gammacarorpt)) +
              aptiw21 * pow (ppt[21],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[21] = (pcar[21]* xcarinww[21] + ppt[21] * xptinww[21])/piw[21];

two_system (&acariw26,"acariw26",&aptiw26,"aptiw26",
           xcarinww[26],xptinww[26],
           pcar[26],ppt[26],
           gammacarorpt                              );

piw[26] = pow ((acariw26 * pow (pcar[26],(1 - gammacarorpt)) +
              aptiw26 * pow (ppt[26],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[26] = (pcar[26]* xcarinww[26] + ppt[26] * xptinww[26])/piw[26];
```

```
two_system (&acariw19,"acariw19",&aptiw19,"aptiw19",
           xcarinww[19],xptinww[19],
           pcar[19],ppt[19],
           gammacarorpt                              );

piw[19] = pow ((acariw19 * pow (pcar[19],(1 - gammacarorpt)) +
               aptiw19 * pow (ppt[19],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[19] = (pcar[19]* xcarinww[19] + ppt[19] * xptinww[19])/piw[19];

two_system (&acariw17,"acariw17",&aptiw17,"aptiw17",
           xcarinww[17],xptinww[17],
           pcar[17],ppt[17],
           gammacarorpt                              );

piw[17] = pow ((acariw17 * pow (pcar[17],(1 - gammacarorpt)) +
               aptiw17 * pow (ppt[17],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[17] = (pcar[17]* xcarinww[17] + ppt[17] * xptinww[17])/piw[17];

two_system (&acariw18,"acariw18",&aptiw18,"aptiw18",
           xcarinww[18],xptinww[18],
           pcar[18],ppt[18],
           gammacarorpt                              );

piw[18] = pow ((acariw18 * pow (pcar[18],(1 - gammacarorpt)) +
               aptiw18 * pow (ppt[18],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[18] = (pcar[18]* xcarinww[18] + ppt[18] * xptinww[18])/piw[18];

two_system (&acariw12,"acariw12",&aptiw12,"aptiw12",
           xcarinww[12],xptinww[12],
           pcar[12],ppt[12],
           gammacarorpt                              );

piw[12] = pow ((acariw12 * pow (pcar[12],(1 - gammacarorpt)) +
               aptiw12 * pow (ppt[12],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[12] = (pcar[12]* xcarinww[12] + ppt[12] * xptinww[12])/piw[12];

two_system (&acariw13,"acariw13",&aptiw13,"aptiw13",
           xcarinww[13],xptinww[13],
           pcar[13],ppt[13],
           gammacarorpt                              );

piw[13] = pow ((acariw13 * pow (pcar[13],(1 - gammacarorpt)) +
               aptiw13 * pow (ppt[13],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[13] = (pcar[13]* xcarinww[13] + ppt[13] * xptinww[13])/piw[13];

two_system (&acariw14,"acariw14",&aptiw14,"aptiw14",
           xcarinww[14],xptinww[14],
           pcar[14],ppt[14],
           gammacarorpt                              );

piw[14] = pow ((acariw14 * pow (pcar[14],(1 - gammacarorpt)) +
               aptiw14 * pow (ppt[14],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[14] = (pcar[14]* xcarinww[14] + ppt[14] * xptinww[14])/piw[14];

two_system (&acariw6,"acariw6",&aptiw6,"aptiw6",
           xcarinww[6],xptinww[6],
           pcar[6],ppt[6],
           gammacarorpt                              );

piw[6] = pow ((acariw6 * pow (pcar[6],(1 - gammacarorpt)) +
               aptiw6 * pow (ppt[6],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinww[6] = (pcar[6]* xcarinww[6] + ppt[6] * xptinww[6])/piw[6];
```

```
two_system (&acariw7,"acariw7",&aptiw7,"aptiw7",
           xcarinww[7],xptinww[7],
           pcar[7],ppt[7],
           gammacarorpt                              );

piw[7] = pow ((acariw7 * pow (pcar[7],(1 - gammacarorpt)) +
              aptiw7 * pow (ppt[7],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[7] = (pcar[7]* xcarinww[7] + ppt[7] * xptinww[7])/piw[7];

two_system (&acariw8,"acariw8",&aptiw8,"aptiw8",
           xcarinww[8],xptinww[8],
           pcar[8],ppt[8],
           gammacarorpt                              );

piw[8] = pow ((acariw8 * pow (pcar[8],(1 - gammacarorpt)) +
              aptiw8 * pow (ppt[8],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[8] = (pcar[8]* xcarinww[8] + ppt[8] * xptinww[8])/piw[8];

two_system (&acariw9,"acariw9",&aptiw9,"aptiw9",
           xcarinww[9],xptinww[9],
           pcar[9],ppt[9],
           gammacarorpt                              );

piw[9] = pow ((acariw9 * pow (pcar[9],(1 - gammacarorpt)) +
              aptiw9 * pow (ppt[9],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[9] = (pcar[9]* xcarinww[9] + ppt[9] * xptinww[9])/piw[9];

two_system (&acariw10,"acariw10",&aptiw10,"aptiw10",
           xcarinww[10],xptinww[10],
           pcar[10],ppt[10],
           gammacarorpt                              );

piw[10] = pow ((acariw10 * pow (pcar[10],(1 - gammacarorpt)) +
              aptiw10 * pow (ppt[10],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[10] = (pcar[10]* xcarinww[10] + ppt[10] * xptinww[10])/piw[10];

two_system (&acariw11,"acariw11",&aptiw11,"aptiw11",
           xcarinww[11],xptinww[11],
           pcar[11],ppt[11],
           gammacarorpt                              );

piw[11] = pow ((acariw11 * pow (pcar[11],(1 - gammacarorpt)) +
              aptiw11 * pow (ppt[11],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[11] = (pcar[11]* xcarinww[11] + ppt[11] * xptinww[11])/piw[11];

two_system (&acariw4,"acariw4",&aptiw4,"aptiw4",
           xcarinww[4],xptinww[4],
           pcar[4],ppt[4],
           gammacarorpt                              );

piw[4] = pow ((acariw4 * pow (pcar[4],(1 - gammacarorpt)) +
              aptiw4 * pow (ppt[4],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[4] = (pcar[4]* xcarinww[4] + ppt[4] * xptinww[4])/piw[4];

two_system (&acariw5,"acariw5",&aptiw5,"aptiw5",
           xcarinww[5],xptinww[5],
           pcar[5],ppt[5],
           gammacarorpt                              );

piw[5] = pow ((acariw5 * pow (pcar[5],(1 - gammacarorpt)) +
              aptiw5 * pow (ppt[5],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[5] = (pcar[5]* xcarinww[5] + ppt[5] * xptinww[5])/piw[5];
```

```
two_system (&acariw1,"acariw1",&aptiw1,"aptiw1",
           xcarinww[1],xptinww[1],
           pcar[1],ppt[1],
           gammacarorpt                              );

piw[1] = pow ((acariw1 * pow (pcar[1],(1 - gammacarorpt)) +
              aptiw1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[1] = (pcar[1]* xcarinww[1] + ppt[1] * xptinww[1])/piw[1];

two_system (&acariw2,"acariw2",&aptiw2,"aptiw2",
           xcarinww[2],xptinww[2],
           pcar[2],ppt[2],
           gammacarorpt                              );

piw[2] = pow ((acariw2 * pow (pcar[2],(1 - gammacarorpt)) +
              aptiw2 * pow (ppt[2],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[2] = (pcar[2]* xcarinww[2] + ppt[2] * xptinww[2])/piw[2];

two_system (&acariw3,"acariw3",&aptiw3,"aptiw3",
           xcarinww[3],xptinww[3],
           pcar[3],ppt[3],
           gammacarorpt                              );

piw[3] = pow ((acariw3 * pow (pcar[3],(1 - gammacarorpt)) +
              aptiw3 * pow (ppt[3],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[3] = (pcar[3]* xcarinww[3] + ppt[3] * xptinww[3])/piw[3];

two_system (&acariw15,"acariw15",&aptiw15,"aptiw15",
           xcarinww[15],xptinww[15],
           pcar[15],ppt[15],
           gammacarorpt                              );

piw[15] = pow ((acariw15 * pow (pcar[15],(1 - gammacarorpt)) +
              aptiw15 * pow (ppt[15],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[15] = (pcar[15]* xcarinww[15] + ppt[15] * xptinww[15])/piw[15];

two_system (&acariw16,"acariw16",&aptiw16,"aptiw16",
           xcarinww[16],xptinww[16],
           pcar[16],ppt[16],
           gammacarorpt                              );

piw[16] = pow ((acariw16 * pow (pcar[16],(1 - gammacarorpt)) +
              aptiw16 * pow (ppt[16],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinww[16] = (pcar[16]* xcarinww[16] + ppt[16] * xptinww[16])/piw[16];

/*inward other*/

two_system (&acario24,"acario24",&aptio24,"aptio24",
           xcarinwo[24],xptinwo[24],
           pcar[24],ppt[24],
           gammacarorpt                              );

pio[24] = pow ((acario24 * pow (pcar[24],(1 - gammacarorpt)) +
              aptio24 * pow (ppt[24],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[24] = (pcar[24]* xcarinwo[24] + ppt[24] * xptinwo[24])/pio[24];

two_system (&acario25,"acario25",&aptio25,"aptio25",
           xcarinwo[25],xptinwo[25],
           pcar[25],ppt[25],
           gammacarorpt                              );

pio[25] = pow ((acario25 * pow (pcar[25],(1 - gammacarorpt)) +
              aptio25 * pow (ppt[25],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));
```

```
xinwo[25] = (pcar[25]* xcarinwo[25] + ppt[25] * xptinwo[25])/pio[25];

two_system (&acario22,"acario22",&aptio22,"aptio22",
           xcarinwo[22],xptinwo[22],
           pcar[22],ppt[22],
           gammacarorpt                              );

pio[22] = pow ((acario22 * pow (pcar[22],(1 - gammacarorpt)) +
               aptio22 * pow (ppt[22],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[22] = (pcar[22]* xcarinwo[22] + ppt[22] * xptinwo[22])/pio[22];

two_system (&acario23,"acario23",&aptio23,"aptio23",
           xcarinwo[23],xptinwo[23],
           pcar[23],ppt[23],
           gammacarorpt                              );

pio[23] = pow ((acario23 * pow (pcar[23],(1 - gammacarorpt)) +
               aptio23 * pow (ppt[23],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[23] = (pcar[23]* xcarinwo[23] + ppt[23] * xptinwo[23])/pio[23];

two_system (&acario20,"acario20",&aptio20,"aptio20",
           xcarinwo[20],xptinwo[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

pio[20] = pow ((acario20 * pow (pcar[20],(1 - gammacarorpt)) +
               aptio20 * pow (ppt[20],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[20] = (pcar[20]* xcarinwo[20] + ppt[20] * xptinwo[20])/pio[20];

two_system (&acario21,"acario21",&aptio21,"aptio21",
           xcarinwo[21],xptinwo[21],
           pcar[21],ppt[21],
           gammacarorpt                              );

pio[21] = pow ((acario21 * pow (pcar[21],(1 - gammacarorpt)) +
               aptio21 * pow (ppt[21],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[21] = (pcar[21]* xcarinwo[21] + ppt[21] * xptinwo[21])/pio[21];

two_system (&acario26,"acario26",&aptio26,"aptio26",
           xcarinwo[26],xptinwo[26],
           pcar[26],ppt[26],
           gammacarorpt                              );

pio[26] = pow ((acario26 * pow (pcar[26],(1 - gammacarorpt)) +
               aptio26 * pow (ppt[26],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[26] = (pcar[26]* xcarinwo[26] + ppt[26] * xptinwo[26])/pio[26];

two_system (&acario19,"acario19",&aptio19,"aptio19",
           xcarinwo[19],xptinwo[19],
           pcar[19],ppt[19],
           gammacarorpt                              );

pio[19] = pow ((acario19 * pow (pcar[19],(1 - gammacarorpt)) +
               aptio19 * pow (ppt[19],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[19] = (pcar[19]* xcarinwo[19] + ppt[19] * xptinwo[19])/pio[19];

two_system (&acario17,"acario17",&aptio17,"aptio17",
           xcarinwo[17],xptinwo[17],
           pcar[17],ppt[17],
           gammacarorpt                              );

pio[17] = pow ((acario17 * pow (pcar[17],(1 - gammacarorpt)) +
               aptio17 * pow (ppt[17],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));
```

```
xinwo[17] = (pcar[17]* xcarinwo[17] + ppt[17] * xptinwo[17])/pio[17];

two_system (&acario18,"acario18",&aptio18,"aptio18",
           xcarinwo[18],xptinwo[18],
           pcar[18],ppt[18],
           gammacarorpt                              );

pio[18] = pow ((acario18 * pow (pcar[18],(1 - gammacarorpt)) +
               aptio18 * pow (ppt[18],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[18] = (pcar[18]* xcarinwo[18] + ppt[18] * xptinwo[18])/pio[18];

two_system (&acario12,"acario12",&aptio12,"aptio12",
           xcarinwo[12],xptinwo[12],
           pcar[12],ppt[12],
           gammacarorpt                              );

pio[12] = pow ((acario12 * pow (pcar[12],(1 - gammacarorpt)) +
               aptio12 * pow (ppt[12],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[12] = (pcar[12]* xcarinwo[12] + ppt[12] * xptinwo[12])/pio[12];

two_system (&acario13,"acario13",&aptio13,"aptio13",
           xcarinwo[13],xptinwo[13],
           pcar[13],ppt[13],
           gammacarorpt                              );

pio[13] = pow ((acario13 * pow (pcar[13],(1 - gammacarorpt)) +
               aptio13 * pow (ppt[13],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[13] = (pcar[13]* xcarinwo[13] + ppt[13]*xptinwo[13])/pio[13] ;

two_system (&acario14,"acario14",&aptio14,"aptio14",
           xcarinwo[14],xptinwo[14],
           pcar[14],ppt[14],
           gammacarorpt                              );

pio[14] = pow ((acario14 * pow (pcar[14],(1 - gammacarorpt)) +
               aptio14 * pow (ppt[14],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[14] = (pcar[14]* xcarinwo[14] + ppt[14]*xptinwo[14])/pio[14] ;

two_system (&acario6,"acario6",&aptio6,"aptio6",
           xcarinwo[6],xptinwo[6],
           pcar[6],ppt[6],
           gammacarorpt                              );

pio[6] = pow ((acario6 * pow (pcar[6],(1 - gammacarorpt)) +
               aptio6 * pow (ppt[6],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[6] = (pcar[6]* xcarinwo[6] + ppt[6]*xptinwo[6])/pio[6] ;

two_system (&acario7,"acario7",&aptio7,"aptio7",
           xcarinwo[7],xptinwo[7],
           pcar[7],ppt[7],
           gammacarorpt                              );

pio[7] = pow ((acario7 * pow (pcar[7],(1 - gammacarorpt)) +
               aptio7 * pow (ppt[7],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[7] = (pcar[7]* xcarinwo[7] + ppt[7]*xptinwo[7])/pio[7] ;

two_system (&acario8,"acario8",&aptio8,"aptio8",
           xcarinwo[8],xptinwo[8],
           pcar[8],ppt[8],
           gammacarorpt                              );

pio[8] = pow ((acario8 * pow (pcar[8],(1 - gammacarorpt)) +
               aptio8 * pow (ppt[8],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));
```

```
xinwo[8] = (pcar[8]* xcarinwo[8] + ppt[8]*xptinwo[8])/pio[8] ;

two_system (&acario9,"acario9",&aptio9,"aptio9",
           xcarinwo[9],xptinwo[9],
           pcar[9],ppt[9],
           gammacarorpt                              );

pio[9] = pow ((acario9 * pow (pcar[9],(1 - gammacarorpt)) +
              aptio9 * pow (ppt[9],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[9] = (pcar[9]* xcarinwo[9] + ppt[9]*xptinwo[9])/pio[9] ;

two_system (&acario10,"acario10",&aptio10,"aptio10",
           xcarinwo[10],xptinwo[10],
           pcar[10],ppt[10],
           gammacarorpt                              );

pio[10] = pow ((acario10 * pow (pcar[10],(1 - gammacarorpt)) +
              aptio10 * pow (ppt[10],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[10] = (pcar[10]* xcarinwo[10] + ppt[10]*xptinwo[10])/pio[10] ;

two_system (&acario11,"acario11",&aptio11,"aptio11",
           xcarinwo[11],xptinwo[11],
           pcar[11],ppt[11],
           gammacarorpt                              );

pio[11] = pow ((acario11 * pow (pcar[11],(1 - gammacarorpt)) +
              aptio11 * pow (ppt[11],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[11] = (pcar[11]* xcarinwo[11] + ppt[11]*xptinwo[11])/pio[11] ;

two_system (&acario4,"acario4",&aptio4,"aptio4",
           xcarinwo[4],xptinwo[4],
           pcar[4],ppt[4],
           gammacarorpt                              );

pio[4] = pow ((acario4 * pow (pcar[4],(1 - gammacarorpt)) +
              aptio4 * pow (ppt[4],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[4] = (pcar[4]* xcarinwo[4] + ppt[4]*xptinwo[4])/pio[4] ;

two_system (&acario5,"acario5",&aptio5,"aptio5",
           xcarinwo[5],xptinwo[5],
           pcar[5],ppt[5],
           gammacarorpt                              );

pio[5] = pow ((acario5 * pow (pcar[5],(1 - gammacarorpt)) +
              aptio5 * pow (ppt[5],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[5] = (pcar[5]* xcarinwo[5] + ppt[5]*xptinwo[5])/pio[5] ;

two_system (&acario1,"acario1",&aptio1,"aptio1",
           xcarinwo[1],xptinwo[1],
           pcar[1],ppt[1],
           gammacarorpt                              );

pio[1] = pow ((acario1 * pow (pcar[1],(1 - gammacarorpt)) +
              aptio1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwo[1] = (pcar[1]* xcarinwo[1] + ppt[1]*xptinwo[1])/pio[1] ;

two_system (&acario2,"acario2",&aptio2,"aptio2",
           xcarinwo[2],xptinwo[2],
           pcar[2],ppt[2],
           gammacarorpt                              );

pio[2] = pow ((acario2 * pow (pcar[2],(1 - gammacarorpt)) +
              aptio2 * pow (ppt[2],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));
```

```
xinwo[2] = (pcar[2]* xcarinwo[2] + ppt[2]*xptinwo[2])/pio[2] ;

two_system (&acario3,"acario3",&aptio3,"aptio3",
           xcarinwo[3],xptinwo[3],
           pcar[3],ppt[3],
           gammacarorpt                                );

pio[3] = pow ((acario3 * pow (pcar[3],(1 - gammacarorpt)) +
               aptio3 * pow (ppt[3],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[3] = (pcar[3]* xcarinwo[3] + ppt[3]*xptinwo[3])/pio[3] ;

two_system (&acario15,"acario15",&aptio15,"aptio15",
           xcarinwo[15],xptinwo[15],
           pcar[15],ppt[15],
           gammacarorpt                                );

pio[15] = pow ((acario15 * pow (pcar[15],(1 - gammacarorpt)) +
               aptio15 * pow (ppt[15],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[15] = (pcar[15]* xcarinwo[15] + ppt[15]*xptinwo[15])/pio[15] ;

two_system (&acario16,"acario16",&aptio16,"aptio16",
           xcarinwo[16],xptinwo[16],
           pcar[16],ppt[16],
           gammacarorpt                                );

pio[16] = pow ((acario16 * pow (pcar[16],(1 - gammacarorpt)) +
               aptio16 * pow (ppt[16],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwo[16] = (pcar[16]* xcarinwo[16] + ppt[16]*xptinwo[16])/pio[16] ;

/*inward school*/

two_system (&acaris3,"acaris3",&aptis3,"aptis3",
           xcarinws[3],xptinws[3],
           pcar[3],ppt[3],
           gammacarorpt                                );

pis[3] = pow ((acaris3 * pow (pcar[3],(1 - gammacarorpt)) +
               aptis3 * pow (ppt[3],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinws[3] = (pcar[3]* xcarinws[3] + ppt[3]*xptinws[3])/pis[3] ;

two_system (&acaris10,"acaris10",&aptis10,"aptis10",
           xcarinws[10],xptinws[10],
           pcar[10],ppt[10],
           gammacarorpt                                );

pis[10] = pow ((acaris10 * pow (pcar[10],(1 - gammacarorpt)) +
               aptis10 * pow (ppt[10],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinws[10] = (pcar[10]* xcarinws[10] + ppt[10]*xptinws[10])/pis[10] ;

two_system (&acaris9,"acaris9",&aptis9,"aptis9",
           xcarinws[9],xptinws[9],
           pcar[9],ppt[9],
           gammacarorpt                                );

pis[9] = pow ((acaris9 * pow (pcar[9],(1 - gammacarorpt)) +
               aptis9 * pow (ppt[9],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinws[9] = (pcar[9]* xcarinws[9] + ppt[9]*xptinws[9])/pis[9] ;

two_system (&acaris11,"acaris11",&aptis11,"aptis11",
           xcarinws[11],xptinws[11],
           pcar[11],ppt[11],
           gammacarorpt                                );

pis[11] = pow ((acaris11 * pow (pcar[11],(1 - gammacarorpt)) +
               aptis11 * pow (ppt[11],(1 - gammacarorpt))),
```

```
            (1.0 / (1 - gammacarorpt)));

xinws[11] = (pcar[11]* xcarinws[11] + ppt[11]*xptinws[11])/pis[11] ;

two_system (&acaris12,"acaris12",&aptis12,"aptis12",
           xcarinws[12],xptinws[12],
           pcar[12],ppt[12],
           gammacarorpt                              );

pis[12] = pow ((acaris12 * pow (pcar[12],(1 - gammacarorpt)) +
              aptis12 * pow (ppt[12],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinws[12] = (pcar[12]* xcarinws[12] + ppt[12]*xptinws[12])/pis[12] ;

two_system (&acaris13,"acaris13",&aptis13,"aptis13",
           xcarinws[13],xptinws[13],
           pcar[13],ppt[13],
           gammacarorpt                              );

pis[13] = pow ((acaris13 * pow (pcar[13],(1 - gammacarorpt)) +
              aptis13 * pow (ppt[13],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinws[13] = (pcar[13]* xcarinws[13] + ppt[13]*xptinws[13])/pis[13] ;

two_system (&acaris14,"acaris14",&aptis14,"aptis14",
           xcarinws[14],xptinws[14],
           pcar[14],ppt[14],
           gammacarorpt                              );

pis[14] = pow ((acaris14 * pow (pcar[14],(1 - gammacarorpt)) +
              aptis14 * pow (ppt[14],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinws[14] = (pcar[14]* xcarinws[14] + ppt[14]*xptinws[14])/pis[14] ;

two_system (&acaris20,"acaris20",&aptis20,"aptis20",
           xcarinws[20],xptinws[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

pis[20] = pow ((acaris20 * pow (pcar[20],(1 - gammacarorpt)) +
              aptis20 * pow (ppt[20],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinws[20] = (pcar[20]* xcarinws[20] + ppt[20]*xptinws[20])/pis[20] ;

two_system (&acaris26,"acaris26",&aptis26,"aptis26",
           xcarinws[26],xptinws[26],
           pcar[26],ppt[26],
           gammacarorpt                              );

pis[26] = pow ((acaris26 * pow (pcar[26],(1 - gammacarorpt)) +
              aptis26 * pow (ppt[26],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinws[26] = (pcar[26]* xcarinws[26] + ppt[26]*xptinws[26])/pis[26] ;

/*inward shopping*/

two_system (&acarish3,"acarish3",&aptish3,"aptish3",
           xcarinwsh[3],xptinwsh[3],
           pcar[3],ppt[3],
           gammacarorpt                              );

pish[3] = pow ((acarish3 * pow (pcar[3],(1 - gammacarorpt)) +
              aptish3 * pow (ppt[3],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xinwsh[3] = (pcar[3]* xcarinwsh[3] + ppt[3]*xptinwsh[3])/pish[3] ;

two_system (&acarish12,"acarish12",&aptish12,"aptish12",
           xcarinwsh[12],xptinwsh[12],
           pcar[12],ppt[12],
           gammacarorpt                              );
```

```
pish[12] = pow ((acarish12 * pow (pcar[12],(1 - gammacarorpt)) +
               aptish12 * pow (ppt[12],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwsh[12] = (pcar[12]* xcarinwsh[12] + ppt[12]*xptinwsh[12])/pish[12] ;

two_system (&acarish13,"acarish13",&aptish13,"aptish13",
           xcarinwsh[13],xptinwsh[13],
           pcar[13],ppt[13],
           gammacarorpt                              );

pish[13] = pow ((acarish13 * pow (pcar[13],(1 - gammacarorpt)) +
               aptish13 * pow (ppt[13],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwsh[13] = (pcar[13]* xcarinwsh[13] + ppt[13]*xptinwsh[13])/pish[13] ;

two_system (&acarish14,"acarish14",&aptish14,"aptish14",
           xcarinwsh[14],xptinwsh[14],
           pcar[14],ppt[14],
           gammacarorpt                              );

pish[14] = pow ((acarish14 * pow (pcar[14],(1 - gammacarorpt)) +
               aptish14 * pow (ppt[14],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwsh[14] = (pcar[14]* xcarinwsh[14] + ppt[14]*xptinwsh[14])/pish[14] ;

two_system (&acarish16,"acarish16",&aptish16,"aptish16",
           xcarinwsh[16],xptinwsh[16],
           pcar[16],ppt[16],
           gammacarorpt                              );

pish[16] = pow ((acarish16 * pow (pcar[16],(1 - gammacarorpt)) +
                aptish16 * pow (ppt[16],(1 - gammacarorpt))),
               (1.0 / (1 - gammacarorpt)));

xinwsh[16] = (pcar[16]* xcarinwsh[16] + ppt[16]*xptinwsh[16])/pish[16] ;

two_system (&acarish17,"acarish17",&aptish17,"aptish17",
           xcarinwsh[17],xptinwsh[17],
           pcar[17],ppt[17],
           gammacarorpt                              );

pish[17] = pow ((acarish17 * pow (pcar[17],(1 - gammacarorpt)) +
               aptish17 * pow (ppt[17],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwsh[17] = (pcar[17]* xcarinwsh[17] + ppt[17]*xptinwsh[17])/pish[17] ;

two_system (&acarish20,"acarish20",&aptish20,"aptish20",
           xcarinwsh[20],xptinwsh[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

pish[20] = pow ((acarish20 * pow (pcar[20],(1 - gammacarorpt)) +
               aptish20 * pow (ppt[20],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xinwsh[20] = (pcar[20]* xcarinwsh[20] + ppt[20]*xptinwsh[20])/pish[20] ;

/*through work*/

two_system (&acartw7,"acartw7",&apttw7,"apttw7",
           xcarthw[7],xptthw[7],
           pcar[7],ppt[7],
           gammacarorpt                              );

ptw[7] = pow ((acartw7 * pow (pcar[7],(1 - gammacarorpt)) +
              apttw7 * pow (ppt[7],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[7] = (pcar[7]* xcarthw[7] + ppt[7]*xptthw[7])/ptw[7] ;

two_system (&acartw8,"acartw8",&apttw8,"apttw8",
           xcarthw[8],xptthw[8],
           pcar[8],ppt[8],
```

```
           gammacarorpt                                );

ptw[8] = pow ((acartw8 * pow (pcar[8],(1 - gammacarorpt)) +
              apttw8 * pow (ppt[8],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[8] = (pcar[8]* xcarthw[8] + ppt[8]*xptthw[8])/ptw[8] ;

two_system (&acartw4,"acartw4",&apttw4,"apttw4",
           xcarthw[4],xptthw[4],
           pcar[4],ppt[4],
           gammacarorpt                                );

ptw[4] = pow ((acartw4 * pow (pcar[4],(1 - gammacarorpt)) +
              apttw4 * pow (ppt[4],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[4] = (pcar[4]* xcarthw[4] + ppt[4]*xptthw[4])/ptw[4] ;

two_system (&acartw19,"acartw19",&apttw19,"apttw19",
           xcarthw[19],xptthw[19],
           pcar[19],ppt[19],
           gammacarorpt                                );

ptw[19] = pow ((acartw19 * pow (pcar[19],(1 - gammacarorpt)) +
              apttw19 * pow (ppt[19],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[19] = (pcar[19]* xcarthw[19] + ppt[19]*xptthw[19])/ptw[19] ;

two_system (&acartw20,"acartw20",&apttw20,"apttw20",
           xcarthw[20],xptthw[20],
           pcar[20],ppt[20],
           gammacarorpt                                );

ptw[20] = pow ((acartw20 * pow (pcar[20],(1 - gammacarorpt)) +
              apttw20 * pow (ppt[20],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[20] = (pcar[20]* xcarthw[20] + ppt[20]*xptthw[20])/ptw[20] ;

two_system (&acartw22,"acartw22",&apttw22,"apttw22",
           xcarthw[22],xptthw[22],
           pcar[22],ppt[22],
           gammacarorpt                                );

ptw[22] = pow ((acartw22 * pow (pcar[22],(1 - gammacarorpt)) +
              apttw22 * pow (ppt[22],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[22] = (pcar[22]* xcarthw[22] + ppt[22]*xptthw[22])/ptw[22] ;

two_system (&acartw23,"acartw23",&apttw23,"apttw23",
           xcarthw[23],xptthw[23],
           pcar[23],ppt[23],
           gammacarorpt                                );

ptw[23] = pow ((acartw23 * pow (pcar[23],(1 - gammacarorpt)) +
              apttw23 * pow (ppt[23],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[23] = (pcar[23]* xcarthw[23] + ppt[23]*xptthw[23])/ptw[23] ;

two_system (&acartw1,"acartw1",&apttw1,"apttw1",
           xcarthw[1],xptthw[1],
           pcar[1],ppt[1],
           gammacarorpt                                );

ptw[1] = pow ((acartw1 * pow (pcar[1],(1 - gammacarorpt)) +
              apttw1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthw[1] = (pcar[1]* xcarthw[1] + ppt[1]*xptthw[1])/ptw[1] ;

two_system (&acartw24,"acartw24",&apttw24,"apttw24",
           xcarthw[24],xptthw[24],
           pcar[24],ppt[24],
```

```
          gammacarorpt                          );

ptw[24] = pow ((acartw24 * pow (pcar[24],(1 - gammacarorpt)) +
               apttw24 * pow (ppt[24],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xthw[24] = (pcar[24]* xcarthw[24] + ppt[24]*xptthw[24])/ptw[24] ;

/*through other*/

two_system (&acarto7,"acarto7",&aptto7,"aptto7",
          xcartho[7],xpttho[7],
          pcar[7],ppt[7],
          gammacarorpt                          );

pto[7] = pow ((acarto7 * pow (pcar[7],(1 - gammacarorpt)) +
               aptto7 * pow (ppt[7],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[7] = (pcar[7]* xcartho[7] + ppt[7]*xpttho[7])/pto[7] ;

two_system (&acarto8,"acarto8",&aptto8,"aptto8",
          xcartho[8],xpttho[8],
          pcar[8],ppt[8],
          gammacarorpt                          );

pto[8] = pow ((acarto8 * pow (pcar[8],(1 - gammacarorpt)) +
               aptto8 * pow (ppt[8],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[8] = (pcar[8]* xcartho[8] + ppt[8]*xpttho[8])/pto[8] ;

two_system (&acarto4,"acarto4",&aptto4,"aptto4",
          xcartho[4],xpttho[4],
          pcar[4],ppt[4],
          gammacarorpt                          );

pto[4] = pow ((acarto4 * pow (pcar[4],(1 - gammacarorpt)) +
               aptto4 * pow (ppt[4],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[4] = (pcar[4]* xcartho[4] + ppt[4]*xpttho[4])/pto[4] ;

two_system (&acarto19,"acarto19",&aptto19,"aptto19",
          xcartho[19],xpttho[19],
          pcar[19],ppt[19],
          gammacarorpt                          );

pto[19] = pow ((acarto19 * pow (pcar[19],(1 - gammacarorpt)) +
               aptto19 * pow (ppt[19],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[19] = (pcar[19]* xcartho[19] + ppt[19]*xpttho[19])/pto[19] ;

two_system (&acarto20,"acarto20",&aptto20,"aptto20",
          xcartho[20],xpttho[20],
          pcar[20],ppt[20],
          gammacarorpt                          );

pto[20] = pow ((acarto20 * pow (pcar[20],(1 - gammacarorpt)) +
               aptto20 * pow (ppt[20],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[20] = (pcar[20]* xcartho[20] + ppt[20]*xpttho[20])/pto[20] ;

two_system (&acarto22,"acarto22",&aptto22,"aptto22",
          xcartho[22],xpttho[22],
          pcar[22],ppt[22],
          gammacarorpt                          );

pto[22] = pow ((acarto22 * pow (pcar[22],(1 - gammacarorpt)) +
               aptto22 * pow (ppt[22],(1 - gammacarorpt))),
              (1.0 / (1 - gammacarorpt)));

xtho[22] = (pcar[22]* xcartho[22] + ppt[22]*xpttho[22])/pto[22] ;

two_system (&acarto23,"acarto23",&aptto23,"aptto23",
```

```
           xcartho[23],xpttho[23],
           pcar[23],ppt[23],
           gammacarorpt                              );

pto[23] = pow ((acarto23 * pow (pcar[23],(1 - gammacarorpt)) +
              aptto23 * pow (ppt[23],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xtho[23] = (pcar[23]* xcartho[23] + ppt[23]*xpttho[23])/pto[23] ;

two_system (&acarto1,"acarto1",&aptto1,"aptto1",
           xcartho[1],xpttho[1],
           pcar[1],ppt[1],
           gammacarorpt                              );

pto[1] = pow ((acarto1 * pow (pcar[1],(1 - gammacarorpt)) +
              aptto1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xtho[1] = (pcar[1]* xcartho[1] + ppt[1]*xpttho[1])/pto[1] ;

two_system (&acarto24,"acarto24",&aptto24,"aptto24",
           xcartho[24],xpttho[24],
           pcar[24],ppt[24],
           gammacarorpt                              );

pto[24] = pow ((acarto24 * pow (pcar[24],(1 - gammacarorpt)) +
              aptto24 * pow (ppt[24],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xtho[24] = (pcar[24]* xcartho[24] + ppt[24]*xpttho[24])/pto[24] ;

/*through school*/

two_system (&acarts7,"acarts7",&aptts7,"aptts7",
           xcarths[7],xptths[7],
           pcar[7],ppt[7],
           gammacarorpt                              );

pts[7] = pow ((acarts7 * pow (pcar[7],(1 - gammacarorpt)) +
              aptts7 * pow (ppt[7],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[7] = (pcar[7]* xcarths[7] + ppt[7]*xptths[7])/pts[7] ;

two_system (&acarts8,"acarts8",&aptts8,"aptts8",
           xcarths[8],xptths[8],
           pcar[8],ppt[8],
           gammacarorpt                              );

pts[8] = pow ((acarts8 * pow (pcar[8],(1 - gammacarorpt)) +
              aptts8 * pow (ppt[8],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[8] = (pcar[8]* xcarths[8] + ppt[8]*xptths[8])/pts[8] ;

two_system (&acarts4,"acarts4",&aptts4,"aptts4",
           xcarths[4],xptths[4],
           pcar[4],ppt[4],
           gammacarorpt                              );

pts[4] = pow ((acarts4 * pow (pcar[4],(1 - gammacarorpt)) +
              aptts4 * pow (ppt[4],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[4] = (pcar[4]* xcarths[4] + ppt[4]*xptths[4])/pts[4] ;

two_system (&acarts19,"acarts19",&aptts19,"aptts19",
           xcarths[19],xptths[19],
           pcar[19],ppt[19],
           gammacarorpt                              );

pts[19] = pow ((acarts19 * pow (pcar[19],(1 - gammacarorpt)) +
              aptts19 * pow (ppt[19],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[19] = (pcar[19]* xcarths[19] + ppt[19]*xptths[19])/pts[19] ;
```

```
two_system (&acarts20,"acarts20",&aptts20,"aptts20",
           xcarths[20],xptths[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

pts[20] = pow ((acarts20 * pow (pcar[20],(1 - gammacarorpt)) +
              aptts20 * pow (ppt[20],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[20] = (pcar[20]* xcarths[20] + ppt[20]*xptths[20])/pts[20] ;

two_system (&acarts22,"acarts22",&aptts22,"aptts22",
           xcarths[22],xptths[22],
           pcar[22],ppt[22],
           gammacarorpt                              );

pts[22] = pow ((acarts22 * pow (pcar[22],(1 - gammacarorpt)) +
              aptts22 * pow (ppt[22],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[22] = (pcar[22]* xcarths[22] + ppt[22]*xptths[22])/pts[22] ;

two_system (&acarts23,"acarts23",&aptts23,"aptts23",
           xcarths[23],xptths[23],
           pcar[23],ppt[23],
           gammacarorpt                              );

pts[23] = pow ((acarts23 * pow (pcar[23],(1 - gammacarorpt)) +
              aptts23 * pow (ppt[23],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[23] = (pcar[23]* xcarths[23] + ppt[23]*xptths[23])/pts[23] ;

two_system (&acarts1,"acarts1",&aptts1,"aptts1",
           xcarths[1],xptths[1],
           pcar[1],ppt[1],
           gammacarorpt                              );

pts[1] = pow ((acarts1 * pow (pcar[1],(1 - gammacarorpt)) +
              aptts1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[1] = (pcar[1]* xcarths[1] + ppt[1]*xptths[1])/pts[1] ;

two_system (&acarts24,"acarts24",&aptts24,"aptts24",
           xcarths[24],xptths[24],
           pcar[24],ppt[24],
           gammacarorpt                              );

pts[24] = pow ((acarts24 * pow (pcar[24],(1 - gammacarorpt)) +
              aptts24 * pow (ppt[24],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xths[24] = (pcar[24]* xcarths[24] + ppt[24]*xptths[24])/pts[24] ;

/*through shopping*/

two_system (&acartsh7,"acartsh7",&apttsh7,"apttsh7",
           xcarthsh[7],xptthsh[7],
           pcar[7],ppt[7],
           gammacarorpt                              );

ptsh[7] = pow ((acartsh7 * pow (pcar[7],(1 - gammacarorpt)) +
              apttsh7 * pow (ppt[7],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[7] = (pcar[7]* xcarthsh[7] + ppt[7]*xptthsh[7])/ptsh[7] ;

two_system (&acartsh8,"acartsh8",&apttsh8,"apttsh8",
           xcarthsh[8],xptthsh[8],
           pcar[8],ppt[8],
           gammacarorpt                              );

ptsh[8] = pow ((acartsh8 * pow (pcar[8],(1 - gammacarorpt)) +
              apttsh8 * pow (ppt[8],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));
```

```
xthsh[8] = (pcar[8]* xcarthsh[8] + ppt[8]*xptthsh[8])/ptsh[8] ;

two_system (&acartsh4,"acartsh4",&apttsh4,"apttsh4",
           xcarthsh[4],xptthsh[4],
           pcar[4],ppt[4],
           gammacarorpt                              );

ptsh[4] = pow ((acartsh4 * pow (pcar[4],(1 - gammacarorpt)) +
              apttsh4 * pow (ppt[4],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[4] = (pcar[4]* xcarthsh[4] + ppt[4]*xptthsh[4])/ptsh[4] ;

two_system (&acartsh19,"acartsh19",&apttsh19,"apttsh19",
           xcarthsh[19],xptthsh[19],
           pcar[19],ppt[19],
           gammacarorpt                              );

ptsh[19] = pow ((acartsh19 * pow (pcar[19],(1 - gammacarorpt)) +
              apttsh19 * pow (ppt[19],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[19] = (pcar[19]* xcarthsh[19] + ppt[19]*xptthsh[19])/ptsh[19] ;

two_system (&acartsh20,"acartsh20",&apttsh20,"apttsh20",
           xcarthsh[20],xptthsh[20],
           pcar[20],ppt[20],
           gammacarorpt                              );

ptsh[20] = pow ((acartsh20 * pow (pcar[20],(1 - gammacarorpt)) +
              apttsh20 * pow (ppt[20],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[20] = (pcar[20]* xcarthsh[20] + ppt[20]*xptthsh[20])/ptsh[20] ;

two_system (&acartsh22,"acartsh22",&apttsh22,"apttsh22",
           xcarthsh[22],xptthsh[22],
           pcar[22],ppt[22],
           gammacarorpt                              );

ptsh[22] = pow ((acartsh22 * pow (pcar[22],(1 - gammacarorpt)) +
              apttsh22 * pow (ppt[22],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[22] = (pcar[22]* xcarthsh[22] + ppt[22]*xptthsh[22])/ptsh[22] ;

two_system (&acartsh23,"acartsh23",&apttsh23,"apttsh23",
           xcarthsh[23],xptthsh[23],
           pcar[23],ppt[23],
           gammacarorpt                              );

ptsh[23] = pow ((acartsh23 * pow (pcar[23],(1 - gammacarorpt)) +
              apttsh23 * pow (ppt[23],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[23] = (pcar[23]* xcarthsh[23] + ppt[23]*xptthsh[23])/ptsh[23] ;

two_system (&acartsh1,"acartsh1",&apttsh1,"apttsh1",
           xcarthsh[1],xptthsh[1],
           pcar[1],ppt[1],
           gammacarorpt                              );

ptsh[1] = pow ((acartsh1 * pow (pcar[1],(1 - gammacarorpt)) +
              apttsh1 * pow (ppt[1],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));

xthsh[1] = (pcar[1]* xcarthsh[1] + ppt[1]*xptthsh[1])/ptsh[1] ;

two_system (&acartsh24,"acartsh24",&apttsh24,"apttsh24",
           xcarthsh[24],xptthsh[24],
           pcar[24],ppt[24],
           gammacarorpt                              );

ptsh[24] = pow ((acartsh24 * pow (pcar[24],(1 - gammacarorpt)) +
              apttsh24 * pow (ppt[24],(1 - gammacarorpt))),
             (1.0 / (1 - gammacarorpt)));
```

```
xthsh[24] = (pcar[24]* xcarthsh[24] + ppt[24]*xptthsh[24])/ptsh[24] ;
```

6.replic

```
/* Replication*/
xt = xmor * at *  pow((pmor/pt), lastgamma);
xi = xmor * ai *  pow((pmor/pi), lastgamma);

xtnm = xt * atnm *  pow((pt/ptnm), gammat  ) ;
xtm = xt * atm *  pow((pt/ptm), gammat );

xto = xtnm * ato *  pow((ptnm/ptho), gammatnm);
xtsh = xtnm * atsh *  pow((ptnm/pthsh), gammatnm) ;

xtw = xtm * atw *  pow((ptm/pthw), gammatm) ;
xts = xtm * ats *  pow((ptm/pths), gammatm) ;

xtoe = xto * atoe *  pow((ptho/ptoe), gammato) ;
xtho[4] = xto * ato4 * pow ((ptho/pto[4]), gammato);
xtow = xto * atow *  pow((ptho/ptow), gammato) ;

xtho[7] = xtoe * atoe7 *  pow((ptoe/pto[7]), gammatoe) ;
xtho[8] = xtoe * atoe8 *  pow((ptoe/pto[8]), gammatoe) ;

xtosw = xtow * atosw *  pow((ptow/ptosw), gammatow) ;
xtonw = xtow * atonw *  pow((ptow/ptonw), gammatow) ;

xtosw1920 = xtosw * atosw1920 *  pow((ptosw/ptosw1920), gammatosw2) ;
xtosw2223 = xtosw * atosw2223 *  pow((ptosw/ptosw2223), gammatosw2) ;

xtho[19] = xtosw1920 * atosw19 *  pow((ptosw1920/pto[19]), gammatosw);
xtho[20] = xtosw1920 * atosw20 *  pow((ptosw1920/pto[20]), gammatosw);

xtho[22] = xtosw2223 * atosw22 *  pow((ptosw2223/pto[22]), gammatosw);
xtho[23] = xtosw2223 * atosw23 *  pow((ptosw2223/pto[23]), gammatosw);

xtho[1] = xtonw * atonw1 *  pow((ptonw/pto[1]), gammatonw) ;
xtho[24] = xtonw * atonw24 *  pow((ptonw/pto[24]), gammatonw) ;

xtshe = xtsh * atshe *  pow((pthsh/ptshe), gammatsh) ;
xthsh[4] = xtsh * atsh4 * pow ((pthsh/ptsh[4]), gammatsh);
xtshw = xtsh * atshw *  pow((pthsh/ptshw), gammatsh) ;

xthsh[7] = xtshe * atshe7 *  pow((ptshe/ptsh[7]), gammatshe) ;
xthsh[8] = xtshe * atshe8 *  pow((ptshe/ptsh[8]), gammatshe) ;

xtshsw = xtshw * atshsw *  pow((ptshw/ptshsw), gammatshw) ;
xtshnw = xtshw * atshnw *  pow((ptshw/ptshnw), gammatshw) ;

xtshsw1920 = xtshsw * atshsw1920 *  pow((ptshsw/ptshsw1920), gammatshsw2) ;
xtshsw2223 = xtshsw * atshsw2223 *  pow((ptshsw/ptshsw2223), gammatshsw2) ;

xthsh[19] = xtshsw1920 * atshsw19 *  pow((ptshsw1920/ptsh[19]), gammatshsw);
xthsh[20] = xtshsw1920 * atshsw20 *  pow((ptshsw1920/ptsh[20]), gammatshsw);

xthsh[22] = xtshsw2223 * atshsw22 *  pow((ptshsw2223/ptsh[22]), gammatshsw);
xthsh[23] = xtshsw2223 * atshsw23 *  pow((ptshsw2223/ptsh[23]), gammatshsw);

xthsh[1] = xtshnw * atshnw1 *  pow((ptshnw/ptsh[1]), gammatshnw) ;
xthsh[24] = xtshnw * atshnw24 *  pow((ptshnw/ptsh[24]), gammatshnw) ;

xtwe = xtw * atwe *  pow((pthw/ptwe), gammatw) ;
xtww = xtw * atww *  pow((pthw/ptww), gammatw) ;

xthw[7] = xtwe * atwe7 *  pow((ptwe/ptw[7]), gammatwe) ;
xthw[8] = xtwe * atwe8 *  pow((ptwe/ptw[8]), gammatwe) ;
xthw[4] = xtw * atw4 * pow ((pthw/ptw[4]), gammatw) ;

xtwsw = xtww * atwsw *  pow((ptww/ptwsw), gammatww) ;
xtwnw = xtww * atwnw *  pow((ptww/ptwnw), gammatww) ;
```

```
xtwsw1920 = xtwsw * atwsw1920 *  pow((ptwsw/ptwsw1920), gammatwsw2) ;
xtwsw2223 = xtwsw * atwsw2223 *  pow((ptwsw/ptwsw2223), gammatwsw2) ;

xthw[19] = xtwsw1920 * atwsw19 *  pow((ptwsw1920/ptw[19]), gammatwsw);
xthw[20] = xtwsw1920 * atwsw20 *  pow((ptwsw1920/ptw[20]), gammatwsw);

xthw[22] = xtwsw2223 * atwsw22 *  pow((ptwsw2223/ptw[22]), gammatwsw);
xthw[23] = xtwsw2223 * atwsw23 *  pow((ptwsw2223/ptw[23]), gammatwsw);

xthw[1] = xtwnw * atwnw1 *  pow((ptwnw/ptw[1]), gammatwnw) ;
xthw[24] = xtwnw * atwnw24 *  pow((ptwnw/ptw[24]), gammatwnw) ;

xtse = xts * atse *  pow((pths/ptse), gammats) ;
xths[4] = xts * ats4 * pow ((pths/pts[4]), gammats) ;
xtsw = xts * atsw *  pow((pths/ptsw), gammats) ;

xths[7] = xtse * atse7 *  pow((ptse/pts[7]), gammatse) ;
xths[8] = xtse * atse8 *  pow((ptse/pts[8]), gammatse) ;

xtssw = xtsw * atssw *  pow((ptsw/ptssw), gammatsw) ;
xtsnw = xtsw * atsnw *  pow((ptsw/ptsnw), gammatsw) ;

xtssw1920 = xtssw * atssw1920 *  pow((ptssw/ptssw1920), gammatssw2) ;
xtssw2223 = xtssw * atssw2223 *  pow((ptssw/ptssw2223), gammatssw2) ;

xths[19] = xtssw1920 * atssw19 *  pow((ptssw1920/pts[19]), gammatssw);
xths[20] = xtssw1920 * atssw20 *  pow((ptssw1920/pts[20]), gammatssw);

xths[22] = xtssw2223 * atssw22 *  pow((ptssw2223/pts[22]), gammatssw);
xths[23] = xtssw2223 * atssw23 *  pow((ptssw2223/pts[23]), gammatssw);

xths[1] = xtsnw * atsnw1 *  pow((ptsnw/pts[1]), gammatsnw) ;
xths[24] = xtsnw * atsnw24 *  pow((ptsnw/pts[24]), gammatsnw) ;

xinm = xi * ainm *  pow((pi/pinm), gammai) ;
xim = xi * aim *  pow((pi/pim), gammai) ;

xio = xinm * aio *  pow((pinm/pino), gammainm);
xish = xinm * aish *  pow((pinm/pinsh), gammainm) ;

xioe = xio * aioe *  pow((pino/pioe) , gammaio) ;
xiosc = xio * aiosc *  pow((pino/piosc) , gammaio) ;
xionwc = xio * aionwc *  pow((pino/pionwc) , gammaio) ;

xioe2425 = xioe * aioe2425 *  pow((pioe/pioe2425) , gammaio3) ;
xioe2223 = xioe * aioe2223 *  pow((pioe/pioe2223) , gammaio3) ;
xioe202126 = xioe * aioe202126 *  pow((pioe/pioe202126) , gammaio3) ;
xinwo[19] = xioe * aioe19 *  pow((pioe/pio[19]) , gammaio3);

xinwo[24] = xioe2425 * aioe24 *  pow((pioe2425/pio[24]) , gammaioe);
xinwo[25] = xioe2425 * aioe25 *  pow((pioe2425/pio[25]) , gammaioe);

xinwo[22] = xioe2223 * aioe22 *  pow((pioe2223/pio[22]) , gammaioe);
xinwo[23] = xioe2223 * aioe23 *  pow((pioe2223/pio[23]) , gammaioe);

xinwo[20] = xioe202126 * aioe20 *  pow((pioe202126/pio[20]) , gammaioe);
xinwo[21] = xioe202126 * aioe21 *  pow((pioe202126/pio[21]) , gammaioe);
xinwo[26] = xioe202126 * aioe26 *  pow((pioe202126/pio[26]) , gammaioe);

xiosc1718 = xiosc * aiosc1718 *  pow((piosc/piosc1718) , gammaio3);
xiosc121314 = xiosc * aiosc121314 *  pow((piosc/piosc121314) , gammaio3);

xinwo[17] = xiosc1718 * aiosc17 *  pow((piosc1718/pio[17]) , gammaiosc) ;
xinwo[18] = xiosc1718 * aiosc18 *  pow((piosc1718/pio[18]) , gammaiosc) ;

xinwo[12] = xiosc121314 * aiosc12 *  pow((piosc121314/pio[12]) , gammaiosc) ;
xinwo[13] = xiosc121314 * aiosc13 *  pow((piosc121314/pio[13]) , gammaiosc) ;
xinwo[14] = xiosc121314 * aiosc14 *  pow((piosc121314/pio[14]) , gammaiosc) ;

xiow = xionwc * aiow *  pow((pionwc/piow) , gammaionwc);
xionc = xionwc * aionc *  pow((pionwc/pionc) , gammaionwc);

xiow678 = xiow * aiow678 *  pow((piow/piow678) , gammaio2) ;
xiow91011 = xiow * aiow91011 *  pow((piow/piow91011) , gammaio2) ;

xinwo[6] = xiow678 * aiow6 *  pow((piow678/pio[6]) , gammaiow) ;
```

```
xinwo[7] = xiow678 * aiow7 *  pow((piow678/pio[7]) , gammaiow) ;
xinwo[8] = xiow678 * aiow8 *  pow((piow678/pio[8]) , gammaiow) ;

xinwo[9] = xiow91011 * aiow9 *  pow((piow91011/pio[9]) , gammaiow) ;
xinwo[10] = xiow91011 * aiow10 *  pow((piow91011/pio[10]) , gammaiow) ;
xinwo[11] = xiow91011 * aiow11 *  pow((piow91011/pio[11]) , gammaiow) ;

xionc45 = xionc * aionc45 *  pow((pionc/pionc45) , gammaio2);
xionc1231516 = xionc * aionc1231516 *  pow((pionc/pionc1231516) , gammaio2);

xinwo[4] = xionc45 * aionc4 *  pow((pionc45/pio[4]) , gammaionc) ;
xinwo[5] = xionc45 * aionc5 *  pow((pionc45/pio[5]) , gammaionc) ;

xinwo[1] = xionc1231516 * aionc1 *  pow((pionc1231516/pio[1]) , gammaionc) ;
xinwo[2] = xionc1231516 * aionc2 *  pow((pionc1231516/pio[2]) , gammaionc) ;
xinwo[3] = xionc1231516 * aionc3 *  pow((pionc1231516/pio[3]) , gammaionc) ;
xinwo[15] = xionc1231516 * aionc15 *  pow((pionc1231516/pio[15]) , gammaionc) ;
xinwo[16] = xionc1231516 * aionc16 *  pow((pionc1231516/pio[16]) , gammaionc) ;

xishnc = xish * aishnc *  pow((pinsh/pishnc) , gammaish) ;
xinwsh[20] = xish * aishj *  pow((pinsh/pish[20]) , gammaish) ;

xinwsh[3] = xishnc * aishn3 *  pow((pishnc/pish[3]) , gammaishnc);
xinwsh[12] = xishnc * aishn12 *  pow((pishnc/pish[12]) , gammaishnc);
xinwsh[13] = xishnc * aishn13 *  pow((pishnc/pish[13]) , gammaishnc);
xinwsh[14] = xishnc * aishn14 *  pow((pishnc/pish[14]) , gammaishnc);
xinwsh[16] = xishnc * aishn16 *  pow((pishnc/pish[16]) , gammaishnc);
xinwsh[17] = xishnc * aishn17 *  pow((pishnc/pish[17]) , gammaishnc);

xiw = xim * aiw *  pow((pim/pinw), gammaim) ;
xis = xim * ais *  pow((pim/pins), gammaim) ;

xiwe = xiw * aiwe *  pow((pinw/piwe) , gammaiw) ;
xiwsc = xiw * aiwsc *  pow((pinw/piwsc) , gammaiw) ;
xiwnwc = xiw * aiwnwc *  pow((pinw/piwnwc) , gammaiw) ;

xiwe2425 = xiwe * aiwe2425 *  pow((piwe/piwe2425) , gammaiw3) ;
xiwe2223 = xiwe * aiwe2223 *  pow((piwe/piwe2223) , gammaiw3) ;
xiwe202126 = xiwe * aiwe202126 *  pow((piwe/piwe202126) , gammaiw3) ;
xinww[19] = xiwe * aiwe19 *  pow((piwe/piw[19]) , gammaiw3);

xinww[24] = xiwe2425 * aiwe24 *  pow((piwe2425/piw[24]) , gammaiwe);
xinww[25] = xiwe2425 * aiwe25 *  pow((piwe2425/piw[25]) , gammaiwe);

xinww[22] = xiwe2223 * aiwe22 *  pow((piwe2223/piw[22]) , gammaiwe);
xinww[23] = xiwe2223 * aiwe23 *  pow((piwe2223/piw[23]) , gammaiwe);

xinww[20] = xiwe202126 * aiwe20 *  pow((piwe202126/piw[20]) , gammaiwe);
xinww[21] = xiwe202126 * aiwe21 *  pow((piwe202126/piw[21]) , gammaiwe);
xinww[26] = xiwe202126 * aiwe26 *  pow((piwe202126/piw[26]) , gammaiwe);

xiwsc1718 = xiwsc * aiwsc1718 *  pow((piwsc/piwsc1718) , gammaiw3);
xiwsc121314 = xiwsc * aiwsc121314 *  pow((piwsc/piwsc121314) , gammaiw3);

xinww[17] = xiwsc1718 * aiwsc17 *  pow((piwsc1718/piw[17]) , gammaiwsc) ;
xinww[18] = xiwsc1718 * aiwsc18 *  pow((piwsc1718/piw[18]) , gammaiwsc) ;

xinww[12] = xiwsc121314 * aiwsc12 *  pow((piwsc121314/piw[12]) , gammaiwsc) ;
xinww[13] = xiwsc121314 * aiwsc13 *  pow((piwsc121314/piw[13]) , gammaiwsc) ;
xinww[14] = xiwsc121314 * aiwsc14 *  pow((piwsc121314/piw[14]) , gammaiwsc) ;

xiww = xiwnwc * aiww *  pow((piwnwc/piww) , gammaiwnwc);
xiwnc = xiwnwc * aiwnc *  pow((piwnwc/piwnc) , gammaiwnwc);

xiww678 = xiww * aiww678 *  pow((piww/piww678) , gammaiw2) ;
xiww91011 = xiww * aiww91011 *  pow((piww/piww91011) , gammaiw2) ;

xinww[6] = xiww678 * aiww6 *  pow((piww678/piw[6]) , gammaiww) ;
xinww[7] = xiww678 * aiww7 *  pow((piww678/piw[7]) , gammaiww) ;
xinww[8] = xiww678 * aiww8 *  pow((piww678/piw[8]) , gammaiww) ;

xinww[9] = xiww91011 * aiww9 *  pow((piww91011/piw[9]) , gammaiww) ;
xinww[10] = xiww91011 * aiww10 *  pow((piww91011/piw[10]) , gammaiww) ;
xinww[11] = xiww91011 * aiww11 *  pow((piww91011/piw[11]) , gammaiww) ;

xiwnc45 = xiwnc * aiwnc45 *  pow((piwnc/piwnc45) , gammaiw2);
xiwnc1231516 = xiwnc * aiwnc1231516 *  pow((piwnc/piwnc1231516) , gammaiw2);
```

```
xinww[4] = xiwnc45 * aiwnc4 *  pow((piwnc45/piw[4]) , gammaiwnc) ;
xinww[5] = xiwnc45 * aiwnc5 *  pow((piwnc45/piw[5]) , gammaiwnc) ;

xinww[1] = xiwnc1231516 * aiwnc1 *  pow((piwnc1231516/piw[1]) , gammaiwnc) ;
xinww[2] = xiwnc1231516 * aiwnc2 *  pow((piwnc1231516/piw[2]) , gammaiwnc) ;
xinww[3] = xiwnc1231516 * aiwnc3 *  pow((piwnc1231516/piw[3]) , gammaiwnc) ;
xinww[15] = xiwnc1231516 * aiwnc15 *  pow((piwnc1231516/piw[15]) , gammaiwnc) ;
xinww[16] = xiwnc1231516 * aiwnc16 *  pow((piwnc1231516/piw[16]) , gammaiwnc) ;

xis310 = xis * ais310 *  pow((pins/pis310) , gammais2) ;
xinws[9] = xis * ais9 *  pow((pins/pis[9]) , gammais2) ;
xinws[11] = xis * ais11 *  pow((pins/pis[11]) , gammais2) ;
xinws[12] = xis * ais12 *  pow((pins/pis[12]) , gammais2) ;
xinws[13] = xis * ais13 *  pow((pins/pis[13]) , gammais2) ;
xinws[14] = xis * ais14 *  pow((pins/pis[14]) , gammais2) ;
xinws[20] = xis * ais20 *  pow((pins/pis[20]) , gammais2) ;
xinws[26] = xis * ais26 *  pow((pins/pis[26]) , gammais2) ;

xinws[3] = xis310 * ais3 *  pow((pis310/pis[3]) , gammais) ;
xinws[10] = xis310 * ais10 *  pow((pis310/pis[10]) , gammais) ;

#include"ptreplic.c"
```

8.pt_replic

```
xcarinww[24] = xinww[24] * acariw24 * pow ((piw[24]/pcar[24]),gammacarorpt);
xptinww[24] = xinww[24] * aptiw24 * pow ((piw[24]/ppt[24]),gammacarorpt);

xcarinww[25] = xinww[25] * acariw25 * pow ((piw[25]/pcar[25]),gammacarorpt);
xptinww[25] = xinww[25] * aptiw25 * pow ((piw[25]/ppt[25]),gammacarorpt);

xcarinww[22] = xinww[22] * acariw22 * pow ((piw[22]/pcar[22]),gammacarorpt);
xptinww[22] = xinww[22] * aptiw22 * pow ((piw[22]/ppt[22]),gammacarorpt);

xcarinww[23] = xinww[23] * acariw23 * pow ((piw[23]/pcar[23]),gammacarorpt);
xptinww[23] = xinww[23] * aptiw23 * pow ((piw[23]/ppt[23]),gammacarorpt);

xcarinww[20] = xinww[20] * acariw20 * pow ((piw[20]/pcar[20]),gammacarorpt);
xptinww[20] = xinww[20] * aptiw20 * pow ((piw[20]/ppt[20]),gammacarorpt);

xcarinww[21] = xinww[21] * acariw21 * pow ((piw[21]/pcar[21]),gammacarorpt);
xptinww[21] = xinww[21] * aptiw21 * pow ((piw[21]/ppt[21]),gammacarorpt);

xcarinww[26] = xinww[26] * acariw26 * pow ((piw[26]/pcar[26]),gammacarorpt);
xptinww[26] = xinww[26] * aptiw26 * pow ((piw[26]/ppt[26]),gammacarorpt);

xcarinww[17] = xinww[17] * acariw17 * pow ((piw[17]/pcar[17]),gammacarorpt);
xptinww[17] = xinww[17] * aptiw17 * pow ((piw[17]/ppt[17]),gammacarorpt);

xcarinww[18] = xinww[18] * acariw18 * pow ((piw[18]/pcar[18]),gammacarorpt);
xptinww[18] = xinww[18] * aptiw18 * pow ((piw[18]/ppt[18]),gammacarorpt);

xcarinww[12] = xinww[12] * acariw12 * pow ((piw[12]/pcar[12]),gammacarorpt);
xptinww[12] = xinww[12] * aptiw12 * pow ((piw[12]/ppt[12]),gammacarorpt);

xcarinww[13] = xinww[13] * acariw13 * pow ((piw[13]/pcar[13]),gammacarorpt);
xptinww[13] = xinww[13] * aptiw13 * pow ((piw[13]/ppt[13]),gammacarorpt);

xcarinww[14] = xinww[14] * acariw14 * pow ((piw[14]/pcar[14]),gammacarorpt);
xptinww[14] = xinww[14] * aptiw14 * pow ((piw[14]/ppt[14]),gammacarorpt);

xcarinww[6] = xinww[6] * acariw6 * pow ((piw[6]/pcar[6]),gammacarorpt);
xptinww[6] = xinww[6] * aptiw6 * pow ((piw[6]/ppt[6]),gammacarorpt);

xcarinww[7] = xinww[7] * acariw7 * pow ((piw[7]/pcar[7]),gammacarorpt);
xptinww[7] = xinww[7] * aptiw7 * pow ((piw[7]/ppt[7]),gammacarorpt);

xcarinww[8] = xinww[8] * acariw8 * pow ((piw[8]/pcar[8]),gammacarorpt);
xptinww[8] = xinww[8] * aptiw8 * pow ((piw[8]/ppt[8]),gammacarorpt);

xcarinww[9] = xinww[9] * acariw9 * pow ((piw[9]/pcar[9]),gammacarorpt);
xptinww[9] = xinww[9] * aptiw9 * pow ((piw[9]/ppt[9]),gammacarorpt);
```

```
xcarinww[10] = xinww[10] * acariw10 * pow ((piw[10]/pcar[10]),gammacarorpt);
xptinww[10] = xinww[10] * aptiw10 * pow ((piw[10]/ppt[10]),gammacarorpt);

xcarinww[11] = xinww[11] * acariw11 * pow ((piw[11]/pcar[11]),gammacarorpt);
xptinww[11] = xinww[11] * aptiw11 * pow ((piw[11]/ppt[11]),gammacarorpt);

xcarinww[4] = xinww[4] * acariw4 * pow ((piw[4]/pcar[4]),gammacarorpt);
xptinww[4] = xinww[4] * aptiw4 * pow ((piw[4]/ppt[4]),gammacarorpt);

xcarinww[5] = xinww[5] * acariw5 * pow ((piw[5]/pcar[5]),gammacarorpt);
xptinww[5] = xinww[5] * aptiw5 * pow ((piw[5]/ppt[5]),gammacarorpt);

xcarinww[1] = xinww[1] * acariw1 * pow ((piw[1]/pcar[1]),gammacarorpt);
xptinww[1] = xinww[1] * aptiw1 * pow ((piw[1]/ppt[1]),gammacarorpt);

xcarinww[2] = xinww[2] * acariw2 * pow ((piw[2]/pcar[2]),gammacarorpt);
xptinww[2] = xinww[2] * aptiw2 * pow ((piw[2]/ppt[2]),gammacarorpt);

xcarinww[3] = xinww[3] * acariw3 * pow ((piw[3]/pcar[3]),gammacarorpt);
xptinww[3] = xinww[3] * aptiw3 * pow ((piw[3]/ppt[3]),gammacarorpt);

xcarinww[15] = xinww[15] * acariw15 * pow ((piw[15]/pcar[15]),gammacarorpt);
xptinww[15] = xinww[15] * aptiw15 * pow ((piw[15]/ppt[15]),gammacarorpt);

xcarinww[16] = xinww[16] * acariw16 * pow ((piw[16]/pcar[16]),gammacarorpt);
xptinww[16] = xinww[16] * aptiw16 * pow ((piw[16]/ppt[16]),gammacarorpt);

xcarinwo[24] = xinwo[24] * acario24 * pow ((pio[24]/pcar[24]),gammacarorpt);
xptinwo[24] = xinwo[24] * aptio24 * pow ((pio[24]/ppt[24]),gammacarorpt);

xcarinwo[25] = xinwo[25] * acario25 * pow ((pio[25]/pcar[25]),gammacarorpt);
xptinwo[25] = xinwo[25] * aptio25 * pow ((pio[25]/ppt[25]),gammacarorpt);

xcarinwo[22] = xinwo[22] * acario22 * pow ((pio[22]/pcar[22]),gammacarorpt);
xptinwo[22] = xinwo[22] * aptio22 * pow ((pio[22]/ppt[22]),gammacarorpt);

xcarinwo[23] = xinwo[23] * acario23 * pow ((pio[23]/pcar[23]),gammacarorpt);
xptinwo[23] = xinwo[23] * aptio23 * pow ((pio[23]/ppt[23]),gammacarorpt);

xcarinwo[20] = xinwo[20] * acario20 * pow ((pio[20]/pcar[20]),gammacarorpt);
xptinwo[20] = xinwo[20] * aptio20 * pow ((pio[20]/ppt[20]),gammacarorpt);

xcarinwo[21] = xinwo[21] * acario21 * pow ((pio[21]/pcar[21]),gammacarorpt);
xptinwo[21] = xinwo[21] * aptio21 * pow ((pio[21]/ppt[21]),gammacarorpt);

xcarinwo[26] = xinwo[26] * acario26 * pow ((pio[26]/pcar[26]),gammacarorpt);
xptinwo[26] = xinwo[26] * aptio26 * pow ((pio[26]/ppt[26]),gammacarorpt);

xcarinwo[17] = xinwo[17] * acario17 * pow ((pio[17]/pcar[17]),gammacarorpt);
xptinwo[17] = xinwo[17] * aptio17 * pow ((pio[17]/ppt[17]),gammacarorpt);

xcarinwo[18] = xinwo[18] * acario18 * pow ((pio[18]/pcar[18]),gammacarorpt);
xptinwo[18] = xinwo[18] * aptio18 * pow ((pio[18]/ppt[18]),gammacarorpt);

xcarinwo[12] = xinwo[12] * acario12 * pow ((pio[12]/pcar[12]),gammacarorpt);
xptinwo[12] = xinwo[12] * aptio12 * pow ((pio[12]/ppt[12]),gammacarorpt);

xcarinwo[13] = xinwo[13] * acario13 * pow ((pio[13]/pcar[13]),gammacarorpt);
xptinwo[13] = xinwo[13] * aptio13 * pow ((pio[13]/ppt[13]),gammacarorpt);

xcarinwo[14] = xinwo[14] * acario14 * pow ((pio[14]/pcar[14]),gammacarorpt);
xptinwo[14] = xinwo[14] * aptio14 * pow ((pio[14]/ppt[14]),gammacarorpt);

xcarinwo[6] = xinwo[6] * acario6 * pow ((pio[6]/pcar[6]),gammacarorpt);
xptinwo[6] = xinwo[6] * aptio6 * pow ((pio[6]/ppt[6]),gammacarorpt);

xcarinwo[7] = xinwo[7] * acario7 * pow ((pio[7]/pcar[7]),gammacarorpt);
xptinwo[7] = xinwo[7] * aptio7 * pow ((pio[7]/ppt[7]),gammacarorpt);

xcarinwo[8] = xinwo[8] * acario8 * pow ((pio[8]/pcar[8]),gammacarorpt);
xptinwo[8] = xinwo[8] * aptio8 * pow ((pio[8]/ppt[8]),gammacarorpt);

xcarinwo[9] = xinwo[9] * acario9 * pow ((pio[9]/pcar[9]),gammacarorpt);
xptinwo[9] = xinwo[9] * aptio9 * pow ((pio[9]/ppt[9]),gammacarorpt);

xcarinwo[10] = xinwo[10] * acario10 * pow ((pio[10]/pcar[10]),gammacarorpt);
```

```
xptinwo[10] = xinwo[10] * aptio10 * pow ((pio[10]/ppt[10]),gammacarorpt);

xcarinwo[11] = xinwo[11] * acario11 * pow ((pio[11]/pcar[11]),gammacarorpt);
xptinwo[11] = xinwo[11] * aptio11 * pow ((pio[11]/ppt[11]),gammacarorpt);

xcarinwo[4] = xinwo[4] * acario4 * pow ((pio[4]/pcar[4]),gammacarorpt);
xptinwo[4] = xinwo[4] * aptio4 * pow ((pio[4]/ppt[4]),gammacarorpt);

xcarinwo[5] = xinwo[5] * acario5 * pow ((pio[5]/pcar[5]),gammacarorpt);
xptinwo[5] = xinwo[5] * aptio5 * pow ((pio[5]/ppt[5]),gammacarorpt);

xcarinwo[1] = xinwo[1] * acario1 * pow ((pio[1]/pcar[1]),gammacarorpt);
xptinwo[1] = xinwo[1] * aptio1 * pow ((pio[1]/ppt[1]),gammacarorpt);

xcarinwo[2] = xinwo[2] * acario2 * pow ((pio[2]/pcar[2]),gammacarorpt);
xptinwo[2] = xinwo[2] * aptio2 * pow ((pio[2]/ppt[2]),gammacarorpt);

xcarinwo[3] = xinwo[3] * acario3 * pow ((pio[3]/pcar[3]),gammacarorpt);
xptinwo[3] = xinwo[3] * aptio3 * pow ((pio[3]/ppt[3]),gammacarorpt);

xcarinwo[15] = xinwo[15] * acario15 * pow ((pio[15]/pcar[15]),gammacarorpt);
xptinwo[15] = xinwo[15] * aptio15 * pow ((pio[15]/ppt[15]),gammacarorpt);

xcarinwo[16] = xinwo[16] * acario16 * pow ((pio[16]/pcar[16]),gammacarorpt);
xptinwo[16] = xinwo[16] * aptio16 * pow ((pio[16]/ppt[16]),gammacarorpt);

xcarinws[3] = xinws[3] * acaris3 * pow ((pis[3]/pcar[3]),gammacarorpt);
xptinws[3] = xinws[3] * aptis3 * pow ((pis[3]/ppt[3]),gammacarorpt);

xcarinws[10] = xinws[10] * acaris10 * pow ((pis[10]/pcar[10]),gammacarorpt);
xptinws[10] = xinws[10] * aptis10 * pow ((pis[10]/ppt[10]),gammacarorpt);

xcarinws[9] = xinws[9] * acaris9 * pow ((pis[9]/pcar[9]),gammacarorpt);
xptinws[9] = xinws[9] * aptis9 * pow ((pis[9]/ppt[9]),gammacarorpt);

xcarinws[11] = xinws[11] * acaris11 * pow ((pis[11]/pcar[11]),gammacarorpt);
xptinws[11] = xinws[11] * aptis11 * pow ((pis[11]/ppt[11]),gammacarorpt);

xcarinws[12] = xinws[12] * acaris12 * pow ((pis[12]/pcar[12]),gammacarorpt);
xptinws[12] = xinws[12] * aptis12 * pow ((pis[12]/ppt[12]),gammacarorpt);

xcarinws[13] = xinws[13] * acaris13 * pow ((pis[13]/pcar[13]),gammacarorpt);
xptinws[13] = xinws[13] * aptis13 * pow ((pis[13]/ppt[13]),gammacarorpt);

xcarinws[14] = xinws[14] * acaris14 * pow ((pis[14]/pcar[14]),gammacarorpt);
xptinws[14] = xinws[14] * aptis14 * pow ((pis[14]/ppt[14]),gammacarorpt);

xcarinws[20] = xinws[20] * acaris20 * pow ((pis[20]/pcar[20]),gammacarorpt);
xptinws[20] = xinws[20] * aptis20 * pow ((pis[20]/ppt[20]),gammacarorpt);

xcarinws[26] = xinws[26] * acaris26 * pow ((pis[26]/pcar[26]),gammacarorpt);
xptinws[26] = xinws[26] * aptis26 * pow ((pis[26]/ppt[26]),gammacarorpt);

xcarinwsh[3] = xinwsh[3] * acarish3 * pow ((pish[3]/pcar[3]),gammacarorpt);
xptinwsh[3] = xinwsh[3] * aptish3 * pow ((pish[3]/ppt[3]),gammacarorpt);

xcarinwsh[12] = xinwsh[12] * acarish12 * pow ((pish[12]/pcar[12]),gammacarorpt);
xptinwsh[12] = xinwsh[12] * aptish12 * pow ((pish[12]/ppt[12]),gammacarorpt);

xcarinwsh[13] = xinwsh[13] * acarish13 * pow ((pish[13]/pcar[13]),gammacarorpt);
xptinwsh[13] = xinwsh[13] * aptish13 * pow ((pish[13]/ppt[13]),gammacarorpt);

xcarinwsh[14] = xinwsh[14] * acarish14 * pow ((pish[14]/pcar[14]),gammacarorpt);
xptinwsh[14] = xinwsh[14] * aptish14 * pow ((pish[14]/ppt[14]),gammacarorpt);

xcarinwsh[16] = xinwsh[16] * acarish16 * pow ((pish[16]/pcar[16]),gammacarorpt);
xptinwsh[16] = xinwsh[16] * aptish16 * pow ((pish[16]/ppt[16]),gammacarorpt);

xcarinwsh[17] = xinwsh[17] * acarish17 * pow ((pish[17]/pcar[17]),gammacarorpt);
xptinwsh[17] = xinwsh[17] * aptish17 * pow ((pish[17]/ppt[17]),gammacarorpt);

xcarinwsh[20] = xinwsh[20] * acarish20 * pow ((pish[20]/pcar[20]),gammacarorpt);
xptinwsh[20] = xinwsh[20] * aptish20 * pow ((pish[20]/ppt[20]),gammacarorpt);

xcarthw[7] = xthw[7] * acartw7 * pow ((ptw[7]/pcar[7]),gammacarorpt);
xptthw[7] = xthw[7] * apttw7 * pow ((ptw[7]/ppt[7]),gammacarorpt);
```

```
xcarthw[8] = xthw[8] * acartw8 * pow ((ptw[8]/pcar[8]),gammacarorpt);
xptthw[8] = xthw[8] * apttw8 * pow ((ptw[8]/ppt[8]),gammacarorpt);

xcarthw[4] = xthw[4] * acartw4 * pow ((ptw[4]/pcar[4]),gammacarorpt);
xptthw[4] = xthw[4] * apttw4 * pow ((ptw[4]/ppt[4]),gammacarorpt);

xcarthw[19] = xthw[19] * acartw19 * pow ((ptw[19]/pcar[19]),gammacarorpt);
xptthw[19] = xthw[19] * apttw19 * pow ((ptw[19]/ppt[19]),gammacarorpt);

xcarthw[20] = xthw[20] * acartw20 * pow ((ptw[20]/pcar[20]),gammacarorpt);
xptthw[20] = xthw[20] * apttw20 * pow ((ptw[20]/ppt[20]),gammacarorpt);

xcarthw[22] = xthw[22] * acartw22 * pow ((ptw[22]/pcar[22]),gammacarorpt);
xptthw[22] = xthw[22] * apttw22 * pow ((ptw[22]/ppt[22]),gammacarorpt);

xcarthw[23] = xthw[23] * acartw23 * pow ((ptw[23]/pcar[23]),gammacarorpt);
xptthw[23] = xthw[23] * apttw23 * pow ((ptw[23]/ppt[23]),gammacarorpt);

xcarthw[1] = xthw[1] * acartw1 * pow ((ptw[1]/pcar[1]),gammacarorpt);
xptthw[1] = xthw[1] * apttw1 * pow ((ptw[1]/ppt[1]),gammacarorpt);

xcarthw[24] = xthw[24] * acartw24 * pow ((ptw[24]/pcar[24]),gammacarorpt);
xptthw[24] = xthw[24] * apttw24 * pow ((ptw[24]/ppt[24]),gammacarorpt);

xcartho[7] = xtho[7] * acarto7 * pow ((pto[7]/pcar[7]),gammacarorpt);
xpttho[7] = xtho[7] * aptto7 * pow ((pto[7]/ppt[7]),gammacarorpt);

xcartho[8] = xtho[8] * acarto8 * pow ((pto[8]/pcar[8]),gammacarorpt);
xpttho[8] = xtho[8] * aptto8 * pow ((pto[8]/ppt[8]),gammacarorpt);

xcartho[4] = xtho[4] * acarto4 * pow ((pto[4]/pcar[4]),gammacarorpt);
xpttho[4] = xtho[4] * aptto4 * pow ((pto[4]/ppt[4]),gammacarorpt);

xcartho[19] = xtho[19] * acarto19 * pow ((pto[19]/pcar[19]),gammacarorpt);
xpttho[19] = xtho[19] * aptto19 * pow ((pto[19]/ppt[19]),gammacarorpt);

xcartho[20] = xtho[20] * acarto20 * pow ((pto[20]/pcar[20]),gammacarorpt);
xpttho[20] = xtho[20] * aptto20 * pow ((pto[20]/ppt[20]),gammacarorpt);

xcartho[22] = xtho[22] * acarto22 * pow ((pto[22]/pcar[22]),gammacarorpt);
xpttho[22] = xtho[22] * aptto22 * pow ((pto[22]/ppt[22]),gammacarorpt);

xcartho[23] = xtho[23] * acarto23 * pow ((pto[23]/pcar[23]),gammacarorpt);
xpttho[23] = xtho[23] * aptto23 * pow ((pto[23]/ppt[23]),gammacarorpt);

xcartho[1] = xtho[1] * acarto1 * pow ((pto[1]/pcar[1]),gammacarorpt);
xpttho[1] = xtho[1] * aptto1 * pow ((pto[1]/ppt[1]),gammacarorpt);

xcartho[24] = xtho[24] * acarto24 * pow ((pto[24]/pcar[24]),gammacarorpt);
xpttho[24] = xtho[24] * aptto24 * pow ((pto[24]/ppt[24]),gammacarorpt);

xcarths[7] = xths[7] * acarts7 * pow ((pts[7]/pcar[7]),gammacarorpt);
xptths[7] = xths[7] * aptts7 * pow ((pts[7]/ppt[7]),gammacarorpt);

xcarths[8] = xths[8] * acarts8 * pow ((pts[8]/pcar[8]),gammacarorpt);
xptths[8] = xths[8] * aptts8 * pow ((pts[8]/ppt[8]),gammacarorpt);

xcarths[4] = xths[4] * acarts4 * pow ((pts[4]/pcar[4]),gammacarorpt);
xptths[4] = xths[4] * aptts4 * pow ((pts[4]/ppt[4]),gammacarorpt);

xcarths[19] = xths[19] * acarts19 * pow ((pts[19]/pcar[19]),gammacarorpt);
xptths[19] = xths[19] * aptts19 * pow ((pts[19]/ppt[19]),gammacarorpt);

xcarths[20] = xths[20] * acarts20 * pow ((pts[20]/pcar[20]),gammacarorpt);
xptths[20] = xths[20] * aptts20 * pow ((pts[20]/ppt[20]),gammacarorpt);

xcarths[22] = xths[22] * acarts22 * pow ((pts[22]/pcar[22]),gammacarorpt);
xptths[22] = xths[22] * aptts22 * pow ((pts[22]/ppt[22]),gammacarorpt);

xcarths[23] = xths[23] * acarts23 * pow ((pts[23]/pcar[23]),gammacarorpt);
xptths[23] = xths[23] * aptts23 * pow ((pts[23]/ppt[23]),gammacarorpt);

xcarths[1] = xths[1] * acarts1 * pow ((pts[1]/pcar[1]),gammacarorpt);
xptths[1] = xths[1] * aptts1 * pow ((pts[1]/ppt[1]),gammacarorpt);

xcarths[24] = xths[24] * acarts24 * pow ((pts[24]/pcar[24]),gammacarorpt);
xptths[24] = xths[24] * aptts24 * pow ((pts[24]/ppt[24]),gammacarorpt);
```

```
xcarthsh[7] = xthsh[7] * acartsh7 * pow ((ptsh[7]/pcar[7]),gammacarorpt);
xptthsh[7] = xthsh[7] * apttsh7 * pow ((ptsh[7]/ppt[7]),gammacarorpt);

xcarthsh[8] = xthsh[8] * acartsh8 * pow ((ptsh[8]/pcar[8]),gammacarorpt);
xptthsh[8] = xthsh[8] * apttsh8 * pow ((ptsh[8]/ppt[8]),gammacarorpt);

xcarthsh[4] = xthsh[4] * acartsh4 * pow ((ptsh[4]/pcar[4]),gammacarorpt);
xptthsh[4] = xthsh[4] * apttsh4 * pow ((ptsh[4]/ppt[4]),gammacarorpt);

xcarthsh[19] = xthsh[19] * acartsh19 * pow ((ptsh[19]/pcar[19]),gammacarorpt);
xptthsh[19] = xthsh[19] * apttsh19 * pow ((ptsh[19]/ppt[19]),gammacarorpt);

xcarthsh[20] = xthsh[20] * acartsh20 * pow ((ptsh[20]/pcar[20]),gammacarorpt);
xptthsh[20] = xthsh[20] * apttsh20 * pow ((ptsh[20]/ppt[20]),gammacarorpt);

xcarthsh[22] = xthsh[22] * acartsh22 * pow ((ptsh[22]/pcar[22]),gammacarorpt);
xptthsh[22] = xthsh[22] * apttsh22 * pow ((ptsh[22]/ppt[22]),gammacarorpt);

xcarthsh[23] = xthsh[23] * acartsh23 * pow ((ptsh[23]/pcar[23]),gammacarorpt);
xptthsh[23] = xthsh[23] * apttsh23 * pow ((ptsh[23]/ppt[23]),gammacarorpt);

xcarthsh[1] = xthsh[1] * acartsh1 * pow ((ptsh[1]/pcar[1]),gammacarorpt);
xptthsh[1] = xthsh[1] * apttsh1 * pow ((ptsh[1]/ppt[1]),gammacarorpt);

xcarthsh[24] = xthsh[24] * acartsh24 * pow ((ptsh[24]/pcar[24]),gammacarorpt);
xptthsh[24] = xthsh[24] * apttsh24 * pow ((ptsh[24]/ppt[24]),gammacarorpt);
```

9.routines

```
void printxerror (x1, x2, a, p1,p2, gam, xstring)

double x1,x2,a,p1,p2,gam;
char *xstring;

{
  printf ("For %d \n",origin);
  printf ("          %s  x * a * pow (p1/p2),gamma)\n",xstring);
  printf ("          %f = %f * %f * pow (%f/%f),%f)\n",x1,x2,a,p1,p2,gam);
}


void two_system (a1,s1,a2,s2,
                x1,x2,
                p1,p2,
                gam    )

double *a1,*a2,x1,x2,p1,p2,gam ;
char *s1,*s2;

{

  if ((x1 == 0) && (x2==0))
    {
      *a1 = *a2 = 0.5;
      fprintf (filelog, "%f\n%f \n",*a1,*a2);
      return;
    }

   *a1 = p1*x1*pow(p1,(gam-1))*pow(p2,(1-gam))/(p2*x2+p1*x1*pow(p1,(gam-1))*pow(p2,(1-
gam)));
   *a2 = p2*x2*pow(p2,(gam-1))*pow(p1,(1-gam))/(p1*x1+p2*x2*pow(p2,(gam-1))*pow(p1,(1-
gam)));

  if ((fabs(  *a1 +  *a2 - 1.0))>0.001)
    {
      *a2=  *a1 = 0.5;
    }

  if ((p1 >= 999.0) && (p2 >= 999.0))
    {
      *a2=  *a1 = 0.5;
    }
```

```
  if (fabs (p1-p2) < 0.10)
    {
      *a2=  *a1 = 0.5;
    }

  if (*a1 < -0.000001)
    {
      printf ("%s (a1) NEGATIVE : %f for origin %d \n", s1,*a1,origin);
      printf ("   other a %f x1 %f x2 %f p1 %f p2 %f gamma %f \n",
              *a2,x1,x2,p1,p2,gam                               );
    }
  if (*a2 < -0.000001)
    {
      printf ("%s (a2) NEGATIVE : %f for origin %d \n", s2,*a2,origin);
      printf ("   other a %f x1 %f x2 %f p1 %f p2 %f gamma %f \n",
              *a1,x1,x2,p1,p2,gam                               );
    }

  fprintf (filelog, "%f\n%f \n",*a1,*a2);
}

void three_system (a1,s1,a2,s2,a3,s3,
                  x1,x2,x3,
                  p1,p2,p3,
                  gam      )

double *a1,*a2,*a3,x1,x2,x3,p1,p2,p3,gam ;
char *s1,*s2,*s3;

{
  double m[3][3];
  double fact[3];

  double b[3];
  int ipiv[3];
  int info;
  int n,nrhs,lda,ldb;
  double verif;

  int dgesv_ ();

  if ((x1 == 0) && (x2==0) && (x3==0))
    {
      *a1 = *a2 = *a3 = 1.0/3.0;
      fprintf (filelog,"%f \n %f \n  %f \n",*a1,*a2,*a3);
      return;
    }


  fact[0] = (p1*x1)/(p1*x1+p2*x2+p3*x3)*pow(p1,(gam-1));
  m[0][0] = fact[0]*pow(p1,(1-gam)) - 1.0 ;
  m[1][0] = fact[0]*pow(p2,(1-gam)) ;
  m[2][0] = fact[0]*pow(p3,(1-gam)) ;

  fact[1] = (p2*x2)/(p1*x1+p2*x2+p3*x3)*pow(p2,(gam-1));
  m[0][1] = fact[1]*pow(p1,(1-gam)) ;
  m[1][1] = fact[1]*pow(p2,(1-gam))  - 1.0 ;
  m[2][1] = fact[1]*pow(p3,(1-gam)) ;

  m[0][2] = 1.0;
  m[1][2] = 1.0;
  m[2][2] = 1.0;

  b[0] = b[1] = 0.0;
  b[2] = 1.0;

  n = 3;
  nrhs = 1;
  lda = 3 ;
  ldb = 3 ;

  dgesv_ (&n, &nrhs, &m, &lda, &ipiv, &b, &ldb, &info ) ;

  *a1 = b[0];
  *a2 = b[1];
  *a3 = b[2];
```

```
  fact[2] = (p3*x3)/(p1*x1+p2*x2+p3*x3)*pow(p3,(gam-1));
  verif = fact[2]*pow(p1,(1-gam)) * *a1 +
    fact[2]*pow(p2,(1-gam)) * *a2 +
    (fact[2]*pow(p3,(1-gam))  - 1.0) * *a3 ;


  if (fabs(verif) > 0.00001)
    {
      printf ("Error in three-system : a1 = %f,a2= %f,a3= %f, verif = %f, info = %d
\n",*a1,*a2,*a3,verif,info);
    };

  if (*a1 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s1,*a1,origin);
    }
  if (*a2 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s2,*a2,origin);
    }
  if (*a3 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s3,*a3,origin);
    }
  fprintf (filelog,"%f \n %f \n  %f \n",*a1,*a2,*a3);
}

void four_system (a1,s1,a2,s2,a3,s3,a4,s4,
                  x1,x2,x3,x4,
                  p1,p2,p3,p4,
                  gam      )

double *a1,*a2,*a3,*a4,x1,x2,x3,x4,p1,p2,p3,p4,gam ;
char *s1,*s2,*s3,*s4;

{
  double m[4][4];

  double fact[4];

  double b[4];
  int ipiv[4];
  int info;
  int n,nrhs,lda,ldb;

  double verif;

  int dgesv_ ();

  if ((x1 == 0) && (x2==0) && (x3 == 0) && (x4 == 0))
    {
      *a1 = *a2 = *a3 = *a4 = 0.25;
      fprintf (filelog,"%f \n  %f  \n %f  \n %f \n",*a1,*a2,*a3,*a4);
      return;
    }

  fact[0] = (p1*x1)/(p1*x1+p2*x2+p3*x3+p4*x4)*pow(p1,(gam-1));
  m[0][0] = fact[0]*pow(p1,(1-gam)) - 1.0 ;
  m[1][0] = fact[0]*pow(p2,(1-gam)) ;
  m[2][0] = fact[0]*pow(p3,(1-gam)) ;
  m[3][0] = fact[0]*pow(p4,(1-gam)) ;

  fact[1] = (p2*x2)/(p1*x1+p2*x2+p3*x3+p4*x4)*pow(p2,(gam-1));
  m[0][1] = fact[1]*pow(p1,(1-gam)) ;
  m[1][1] = fact[1]*pow(p2,(1-gam))  - 1.0 ;
  m[2][1] = fact[1]*pow(p3,(1-gam)) ;
  m[3][1] = fact[1]*pow(p4,(1-gam)) ;

  fact[2] = (p3*x3)/(p1*x1+p2*x2+p3*x3+p4*x4)*pow(p3,(gam-1));
  m[0][2] = fact[2]*pow(p1,(1-gam)) ;
  m[1][2] = fact[2]*pow(p2,(1-gam)) ;
  m[2][2] = fact[2]*pow(p3,(1-gam))  - 1.0 ;
  m[3][2] = fact[2]*pow(p4,(1-gam)) ;

  m[0][3] = 1.0 ;
```

```
  m[1][3] = 1.0 ;
  m[2][3] = 1.0 ;
  m[3][3] = 1.0 ;

  b[0] = b[1] = b[2] = 0.0;
  b[3] = 1.0;

  n = 4;
  nrhs = 1;
  lda = 4;
  ldb = 4;

  dgesv_ (&n, &nrhs, &m, &lda, &ipiv, &b, &ldb, &info ) ;

  *a1 = b[0];
  *a2 = b[1];
  *a3 = b[2];
  *a4 = b[3];

  fact[3] = (p4*x4)/(p1*x1+p2*x2+p3*x3+p4*x4)*pow(p4,(gam-1));
  verif = fact[3]*pow(p1,(1-gam)) * *a1 +
    fact[3]*pow(p2,(1-gam)) * *a2 +
    fact[3]*pow(p3,(1-gam)) * *a3 +
    (fact[3]*pow(p4,(1-gam))  - 1.0)* *a4 ;

  if (fabs(verif) > 0.00001)
    {
      printf ("Error in four-system : a1 = %f,a2= %f,a3= %f, a4 = %f, verif = %f, info
= %d\n",*a1,*a2,*a3,*a4,verif,info);
    };

  if (*a1 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s1,*a1,origin);
    }
  if (*a2 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s2,*a2,origin);
    }
  if (*a3 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s3,*a3,origin);
    }
  if (*a4 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s4,*a4,origin);
    }
  fprintf (filelog,"%f \n  %f  \n %f  \n %f \n",*a1,*a2,*a3,*a4);
}

void five_system (a1,s1,a2,s2,a3,s3,a4,s4,a5,s5,
                  x1,x2,x3,x4,x5,
                  p1,p2,p3,p4,p5,
                  gam      )

double *a1,*a2,*a3,*a4,*a5,x1,x2,x3,x4,x5,p1,p2,p3,p4,p5,gam ;
char *s1,*s2,*s3,*s4,*s5;

{
  double m[5][5];

  double fact[5];

  double b[5];
  int ipiv[5];
  int info;
  int n,nrhs,lda,ldb;

  double verif;

  int dgesv_ ();

  if ((x1 == 0) && (x2==0) && (x3 == 0) && (x4 == 0) && (x5 == 0))
    {
      *a1 = *a2 = *a3 = *a4 = *a5 = 0.2;
      fprintf (filelog, "%f  \n %f  \n %f  \n %f  \n %f \n",*a1,*a2,*a3,*a4,*a5);
      return;
```

```
    }

  fact[0] = (p1*x1)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5)*pow(p1,(gam-1));
  m[0][0] = fact[0]*pow(p1,(1-gam)) - 1.0 ;
  m[1][0] = fact[0]*pow(p2,(1-gam)) ;
  m[2][0] = fact[0]*pow(p3,(1-gam)) ;
  m[3][0] = fact[0]*pow(p4,(1-gam)) ;
  m[4][0] = fact[0]*pow(p5,(1-gam)) ;

  fact[1] = (p2*x2)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5)*pow(p2,(gam-1));
  m[0][1] = fact[1]*pow(p1,(1-gam)) ;
  m[1][1] = fact[1]*pow(p2,(1-gam))  - 1.0 ;
  m[2][1] = fact[1]*pow(p3,(1-gam)) ;
  m[3][1] = fact[1]*pow(p4,(1-gam)) ;
  m[4][1] = fact[1]*pow(p5,(1-gam)) ;

  fact[2] = (p3*x3)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5)*pow(p3,(gam-1));
  m[0][2] = fact[2]*pow(p1,(1-gam)) ;
  m[1][2] = fact[2]*pow(p2,(1-gam)) ;
  m[2][2] = fact[2]*pow(p3,(1-gam))  - 1.0 ;
  m[3][2] = fact[2]*pow(p4,(1-gam)) ;
  m[4][2] = fact[2]*pow(p5,(1-gam)) ;

  fact[3] = (p4*x4)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5)*pow(p4,(gam-1));
  m[0][3] = fact[3]*pow(p1,(1-gam)) ;
  m[1][3] = fact[3]*pow(p2,(1-gam)) ;
  m[2][3] = fact[3]*pow(p3,(1-gam)) ;
  m[3][3] = fact[3]*pow(p4,(1-gam))  - 1.0 ;
  m[4][3] = fact[3]*pow(p5,(1-gam)) ;

  m[0][4] = 1.0 ;
  m[1][4] = 1.0 ;
  m[2][4] = 1.0 ;
  m[3][4] = 1.0 ;
  m[4][4] = 1.0 ;

  b[0] = b[1] = b[2] = b[3] = 0.0;
  b[4] = 1.0;

  n = 5;
  nrhs = 1;
  lda = 5;
  ldb = 5;

  dgesv_ (&n, &nrhs, &m, &lda, &ipiv, &b, &ldb, &info ) ;

  *a1 = b[0];
  *a2 = b[1];
  *a3 = b[2];
  *a4 = b[3];
  *a5 = b[4];

  fact[4] = (p5*x5)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5)*pow(p5,(gam-1));
  verif = fact[4]*pow(p1,(1-gam)) * *a1 +
    fact[4]*pow(p2,(1-gam)) * *a2 +
    fact[4]*pow(p3,(1-gam)) * *a3 +
    fact[4]*pow(p4,(1-gam)) * *a4 +
    (fact[4]*pow(p5,(1-gam))  - 1.0)* *a5 ;

  if (fabs(verif) > 0.00001)
    {
      printf ("Error in five-system : a1 = %f,a2= %f,a3= %f, a4 = %f, a5 = %f, verif =
%f, info = %d\n",*a1,*a2,*a3,*a4,*a5,verif,info);
    };

  if (*a1 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s1,*a1,origin);
    }
  if (*a2 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s2,*a2,origin);
    }
  if (*a3 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s3,*a3,origin);
    }
```

```
  if (*a4 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s4,*a4,origin);
    }
  if (*a5 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s5,*a5,origin);
    }
    fprintf (filelog, "%f  \n %f  \n %f  \n %f  \n %f \n",*a1,*a2,*a3,*a4,*a5);
}

void six_system (a1,s1,a2,s2,a3,s3,a4,s4,a5,s5,a6,s6,
                  x1,x2,x3,x4,x5,x6,
                  p1,p2,p3,p4,p5,p6,
                  gam      )

double *a1,*a2,*a3,*a4,*a5,*a6,x1,x2,x3,x4,x5,x6,p1,p2,p3,p4,p5,p6,gam ;
char *s1,*s2,*s3,*s4,*s5,*s6;

{
  double m[6][6];

  double fact[6];

  double b[6];
  int ipiv[6];
  int info;
  int n,nrhs,lda,ldb;

  double verif;

  int dgesv_ ();

  if ((x1 == 0) && (x2==0) && ( x3 == 0 ) && ( x4 == 0 ) && ( x5 == 0 ) && ( x6 == 0
))
    {
      *a1 = *a2 = *a3 = *a4 = *a5 = *a6 = 1.0/6.0;
      fprintf (filelog,"%f  \n %f  \n %f  \n %f  \n %f  \n %f\n",
          *a1,*a2,*a3,*a4,*a5,*a6);
      return;
    }

  fact[0] = (p1*x1)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p1,(gam-1));
  m[0][0] = fact[0]*pow(p1,(1-gam)) - 1.0 ;
  m[1][0] = fact[0]*pow(p2,(1-gam)) ;
  m[2][0] = fact[0]*pow(p3,(1-gam)) ;
  m[3][0] = fact[0]*pow(p4,(1-gam)) ;
  m[4][0] = fact[0]*pow(p5,(1-gam)) ;
  m[5][0] = fact[0]*pow(p6,(1-gam)) ;

  fact[1] = (p2*x2)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p2,(gam-1));
  m[0][1] = fact[1]*pow(p1,(1-gam)) ;
  m[1][1] = fact[1]*pow(p2,(1-gam))  - 1.0 ;
  m[2][1] = fact[1]*pow(p3,(1-gam)) ;
  m[3][1] = fact[1]*pow(p4,(1-gam)) ;
  m[4][1] = fact[1]*pow(p5,(1-gam)) ;
  m[5][1] = fact[1]*pow(p6,(1-gam)) ;

  fact[2] = (p3*x3)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p3,(gam-1));
  m[0][2] = fact[2]*pow(p1,(1-gam)) ;
  m[1][2] = fact[2]*pow(p2,(1-gam)) ;
  m[2][2] = fact[2]*pow(p3,(1-gam))  - 1.0 ;
  m[3][2] = fact[2]*pow(p4,(1-gam)) ;
  m[4][2] = fact[2]*pow(p5,(1-gam)) ;
  m[5][2] = fact[2]*pow(p6,(1-gam)) ;

  fact[3] = (p4*x4)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p4,(gam-1));
  m[0][3] = fact[3]*pow(p1,(1-gam)) ;
  m[1][3] = fact[3]*pow(p2,(1-gam)) ;
  m[2][3] = fact[3]*pow(p3,(1-gam)) ;
  m[3][3] = fact[3]*pow(p4,(1-gam))  - 1.0 ;
  m[4][3] = fact[3]*pow(p5,(1-gam)) ;
  m[5][3] = fact[3]*pow(p6,(1-gam)) ;

  fact[4] = (p5*x5)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p5,(gam-1));
  m[0][4] = fact[4]*pow(p1,(1-gam)) ;
  m[1][4] = fact[4]*pow(p2,(1-gam)) ;
```

```
  m[2][4] = fact[4]*pow(p3,(1-gam)) ;
  m[3][4] = fact[4]*pow(p4,(1-gam)) ;
  m[4][4] = fact[4]*pow(p5,(1-gam)) - 1.0 ;
  m[5][4] = fact[4]*pow(p6,(1-gam)) ;

  m[0][5] = 1.0 ;
  m[1][5] = 1.0 ;
  m[2][5] = 1.0 ;
  m[3][5] = 1.0 ;
  m[4][5] = 1.0 ;
  m[5][5] = 1.0 ;

  b[0] = b[1] = b[2] = b[3] = b[4]  = 0.0;
  b[5] = 1.0;

  n = 6;
  nrhs = 1;
  lda = 6;
  ldb = 6;

  dgesv_ (&n, &nrhs, &m, &lda, &ipiv, &b, &ldb, &info ) ;

  *a1 = b[0];
  *a2 = b[1];
  *a3 = b[2];
  *a4 = b[3];
  *a5 = b[4];
  *a6 = b[5];

  fact[5] = (p6*x6)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6)*pow(p6,(gam-1));
  verif = fact[5]*pow(p1,(1-gam)) * *a1 +
    fact[5]*pow(p2,(1-gam)) * *a2 +
    fact[5]*pow(p3,(1-gam)) * *a3 +
    fact[5]*pow(p4,(1-gam)) * *a4 +
    fact[5]*pow(p5,(1-gam)) * *a5 +
    (fact[5]*pow(p6,(1-gam)) - 1.0) * *a6 ;

  if (fabs(verif) > 0.00001)
    {
      printf ("Error in six-system : a1 = %f,a2= %f,a3= %f, a4 = %f, a5 = %f, a6 = %f,
verif = %f, info = %d\n",*a1,*a2,*a3,*a4,*a5,*a6,verif,info);
    };

  if (*a1 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s1,*a1,origin);
    }
  if (*a2 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s2,*a2,origin);
    }
  if (*a3 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s3,*a3,origin);
    }
  if (*a4 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s4,*a4,origin);
    }
  if (*a5 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s5,*a5,origin);
    }
  if (*a6 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s6,*a6,origin);
    }
      fprintf (filelog,"%f  \n %f  \n %f  \n %f  \n %f  \n %f\n",
          *a1,*a2,*a3,*a4,*a5,*a6);
}

void eight_system (a1,s1,a2,s2,a3,s3,a4,s4,a5,s5,a6,s6,a7,s7,a8,s8,
                  x1,x2,x3,x4,x5,x6,x7,x8,
                  p1,p2,p3,p4,p5,p6,p7,p8,
                  gam      )
```

```
double
*a1,*a2,*a3,*a4,*a5,*a6,*a7,*a8,x1,x2,x3,x4,x5,x6,x7,x8,p1,p2,p3,p4,p5,p6,p7,p8,gam ;

{
  double m[8][8];

  double fact[8];

  double b[8];
  int ipiv[8];
  int info;
  int n,nrhs,lda,ldb;

  double verif ;

  int dgesv_ ();

  if ((x1 == 0) && (x2==0) && (x3 == 0) && (x4==0) && (x5 == 0) && (x6==0) && (x7 ==
0) && (x8==0))
    {
      *a1 = *a2 = *a3 = *a4 = *a5 = *a6 = *a7 = *a8 = 0.125;
      fprintf (filelog,"%f  \n %f  \n %f \n  %f \n  %f  \n %f  \n %f  \n %f\n",
         *a1,*a2,*a3,*a4,*a5,*a6,*a7,*a8);
      return;
    }

  fact[0] = (p1*x1)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p1,(gam-1));
  m[0][0] = fact[0]*pow(p1,(1-gam)) - 1.0 ;
  m[1][0] = fact[0]*pow(p2,(1-gam)) ;
  m[2][0] = fact[0]*pow(p3,(1-gam)) ;
  m[3][0] = fact[0]*pow(p4,(1-gam)) ;
  m[4][0] = fact[0]*pow(p5,(1-gam)) ;
  m[5][0] = fact[0]*pow(p6,(1-gam)) ;
  m[6][0] = fact[0]*pow(p7,(1-gam)) ;
  m[7][0] = fact[0]*pow(p8,(1-gam)) ;

  fact[1] = (p2*x2)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p2,(gam-1));
  m[0][1] = fact[1]*pow(p1,(1-gam)) ;
  m[1][1] = fact[1]*pow(p2,(1-gam))  - 1.0 ;
  m[2][1] = fact[1]*pow(p3,(1-gam)) ;
  m[3][1] = fact[1]*pow(p4,(1-gam)) ;
  m[4][1] = fact[1]*pow(p5,(1-gam)) ;
  m[5][1] = fact[1]*pow(p6,(1-gam)) ;
  m[6][1] = fact[1]*pow(p7,(1-gam)) ;
  m[7][1] = fact[1]*pow(p8,(1-gam)) ;

  fact[2] = (p3*x3)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p3,(gam-1));
  m[0][2] = fact[2]*pow(p1,(1-gam)) ;
  m[1][2] = fact[2]*pow(p2,(1-gam)) ;
  m[2][2] = fact[2]*pow(p3,(1-gam))  - 1.0 ;
  m[3][2] = fact[2]*pow(p4,(1-gam)) ;
  m[4][2] = fact[2]*pow(p5,(1-gam)) ;
  m[5][2] = fact[2]*pow(p6,(1-gam)) ;
  m[6][2] = fact[2]*pow(p7,(1-gam)) ;
  m[7][2] = fact[2]*pow(p8,(1-gam)) ;

  fact[3] = (p4*x4)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p4,(gam-1));
  m[0][3] = fact[3]*pow(p1,(1-gam)) ;
  m[1][3] = fact[3]*pow(p2,(1-gam)) ;
  m[2][3] = fact[3]*pow(p3,(1-gam)) ;
  m[3][3] = fact[3]*pow(p4,(1-gam))  - 1.0 ;
  m[4][3] = fact[3]*pow(p5,(1-gam)) ;
  m[5][3] = fact[3]*pow(p6,(1-gam)) ;
  m[6][3] = fact[3]*pow(p7,(1-gam)) ;
  m[7][3] = fact[3]*pow(p8,(1-gam)) ;

  fact[4] = (p5*x5)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p5,(gam-1));
  m[0][4] = fact[4]*pow(p1,(1-gam)) ;
  m[1][4] = fact[4]*pow(p2,(1-gam)) ;
  m[2][4] = fact[4]*pow(p3,(1-gam)) ;
  m[3][4] = fact[4]*pow(p4,(1-gam)) ;
  m[4][4] = fact[4]*pow(p5,(1-gam)) - 1.0 ;
  m[5][4] = fact[4]*pow(p6,(1-gam)) ;
  m[6][4] = fact[4]*pow(p7,(1-gam)) ;
  m[7][4] = fact[4]*pow(p8,(1-gam)) ;

  fact[5] = (p6*x6)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p6,(gam-1));
```

```
  m[0][5] = fact[5]*pow(p1,(1-gam)) ;
  m[1][5] = fact[5]*pow(p2,(1-gam)) ;
  m[2][5] = fact[5]*pow(p3,(1-gam)) ;
  m[3][5] = fact[5]*pow(p4,(1-gam)) ;
  m[4][5] = fact[5]*pow(p5,(1-gam)) ;
  m[5][5] = fact[5]*pow(p6,(1-gam)) - 1.0 ;
  m[6][5] = fact[5]*pow(p7,(1-gam)) ;
  m[7][5] = fact[5]*pow(p8,(1-gam)) ;


  fact[6] = (p7*x7)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p7,(gam-1));
  m[0][6] = fact[6]*pow(p1,(1-gam)) ;
  m[1][6] = fact[6]*pow(p2,(1-gam)) ;
  m[2][6] = fact[6]*pow(p3,(1-gam)) ;
  m[3][6] = fact[6]*pow(p4,(1-gam)) ;
  m[4][6] = fact[6]*pow(p5,(1-gam)) ;
  m[5][6] = fact[6]*pow(p6,(1-gam)) ;
  m[6][6] = fact[6]*pow(p7,(1-gam)) - 1.0 ;
  m[7][6] = fact[6]*pow(p8,(1-gam)) ;

  m[0][7] = 1.0 ;
  m[1][7] = 1.0 ;
  m[2][7] = 1.0 ;
  m[3][7] = 1.0 ;
  m[4][7] = 1.0 ;
  m[5][7] = 1.0 ;
  m[6][7] = 1.0 ;
  m[7][7] = 1.0 ;

  b[0] = b[1] = b[2] = b[3] = b[4] = b[5] = b[6] = 0.0;
  b[7] = 1.0;

  n = 8;
  nrhs = 1;
  lda = 8;
  ldb = 8;

  dgesv_ (&n, &nrhs, &m, &lda, &ipiv, &b, &ldb, &info ) ;

  *a1 = b[0];
  *a2 = b[1];
  *a3 = b[2];
  *a4 = b[3];
  *a5 = b[4];
  *a6 = b[5];
  *a7 = b[6];
  *a8 = b[7];

  fact[7] = (p8*x8)/(p1*x1+p2*x2+p3*x3+p4*x4+p5*x5+p6*x6+p7*x7+p8*x8)*pow(p8,(gam-1));
  verif = fact[7]*pow(p1,(1-gam)) * *a1 +
    fact[7]*pow(p2,(1-gam)) * *a2 +
    fact[7]*pow(p3,(1-gam)) * *a3 +
    fact[7]*pow(p4,(1-gam)) * *a4 +
    fact[7]*pow(p5,(1-gam)) * *a5 +
    fact[7]*pow(p6,(1-gam)) * *a6 +
    fact[7]*pow(p7,(1-gam)) * *a7 +
    (fact[7]*pow(p8,(1-gam)) - 1.0) * *a8;


  if (fabs(verif) > 0.00001)
    {
      printf ("Error in eight-system : a1 = %f,a2= %f,a3= %f, a4 = %f, a5 = %f, a6 =
%f, a7 = %f, a8 = %f,verif = %f, info =
%d\n",*a1,*a2,*a3,*a4,*a5,*a6,*a7,*a8,verif,info);
    };

  if (*a1 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s1,*a1,origin);
    }
  if (*a2 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s2,*a2,origin);
    }
  if (*a3 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s3,*a3,origin);
```

```
    }
  if (*a4 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s4,*a4,origin);
    }
  if (*a5 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s5,*a5,origin);
    }
  if (*a6 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s6,*a6,origin);
    }
  if (*a7 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s7,*a7,origin);
    }
  if (*a8 < -0.000001)
    {
      printf ("%s NEGATIVE : %f for origin %d \n", s8,*a8,origin);
    }
      fprintf (filelog,"%f  \n %f  \n %f \n  %f \n  %f  \n %f  \n %f  \n %f\n",
          *a1,*a2,*a3,*a4,*a5,*a6,*a7,*a8);

}
```

Bijlage 3

An experimental case study for Namur

1. Introduction

This paper presents and applies a simulation model for the assessment of congestion pricing mechanisms in transport networks. It combines a demand model with an equilibrium model of a transport network. Iterative implementation of both components leads to a simultaneous consumer equilibrium and network equilibrium, for a given set of link taxes. The degree of efficiency of the combined equilibrium is assessed by standard economic measures. The model can be used to assess any given congestion pricing scheme, and it allows to search for second-best pricing mechanisms. The motivation for constructing this model is twofold:

- Available ‘elastic demand’ network models lack economic consistency in the specification of demand. The basic goal of network assignment models is to find a user equilibrium on the transport network for a given origin-destination matrix. Extensions of this framework to allow for price sensitivity of demand, have usually relied on the definition of a demand curve for each origin-destination pair (e.g. Hall et al., 1992). These demand curves are defined over the price of the origin-destination-pair only. The price elasticity implicity represents the dimensions of choice of departure time and travel mode, as well as the decision to travel or not. Usually the demand functions do not allow for substitution between origins and/or destinations. The reason for the fairly partial nature of the demand specification is that the demand function is modelled through ‘virtual links’ within the network model. This approach allows the application of (highly efficient) algorithms for a fixed demand network model, in the elastic demand setting.

  In this paper, we will use a more detailed and consistent demand structure, which allows to take account of tax revenue use and which allows for welfare assessment. Since such a demand model can not be represented within the network structure, it is constructed as a separate module. It is described in section 2.2.

- When not all links in a transport network can be taxed (partial network pricing), it is interesting to find out what the properties of second-best partial network pricing schemes are. However, the construction of second-best tax optimisation models runs into substantial computational difficulties. This is related to the combinatorial nature of the optimisation problem (a combinatorial optimisation problem is an optimisation problem in which the space of possible solutions is discrete). This issue is further explored in section 2.3. Here we note that a simulation approach sidesteps the difficulties by treating the network equilibrium problem and the consumer optimisation problem separately.

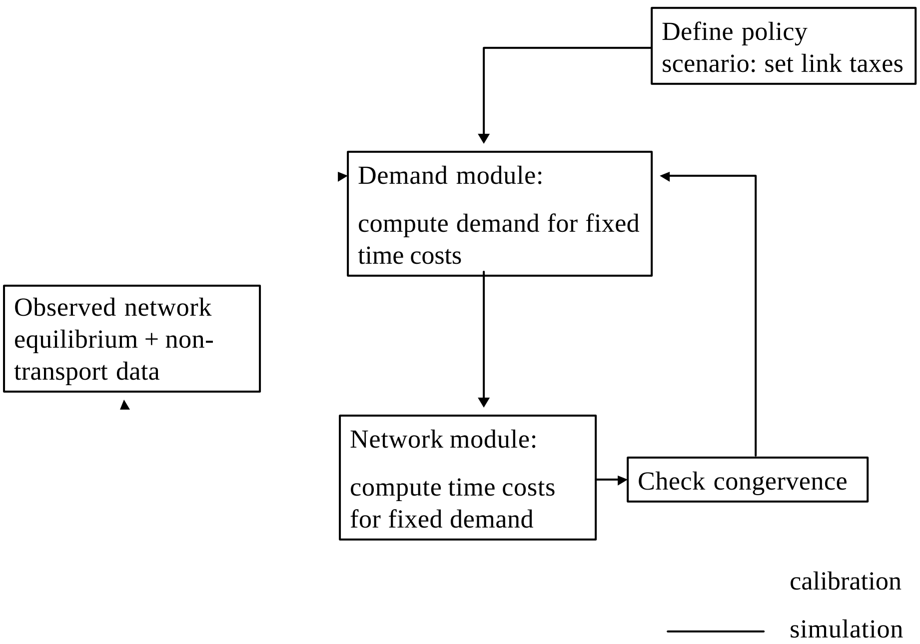
The model is applied to a thirty-link network representation of the city of Namur (Belgium), using demand data for the morning peak of an average workday in 2000 (see section 3). By using the simulation model in a limited grid search, we find that partial pricing schemes in which only a small subset of links is optimally taxed, are potentially very effective in dealing with network congestion. A four-link congestion pricing scheme is identified which yields ¾ of the first-best efficiency gain.

Although the network model is small compared to those used in transportation science (which may contain thousands of links), it captures the essential features of a network assignment process. In particular, for some origin-destination pairs multiple paths are used, and the set of used paths is seen to depend on the link tax configuration.

2. Model structure

The simulation model iterates between a demand module and a network module, untill equilibrium is reached in both modules, given the exogenously specified link taxes (figure 1). The network module specifies the topology of the network and the link cost functions. In principle, any standard network equilibrium model can be used. The demand module specifies the demand system as a nested CES-utility tree. The parameterised version of the demand system is obtained through calibration for an observed network equilibrium. We discuss both modules in sections 2.1 and 2.2. In section 2.3, the difficulties of constructing an integrated optimisation model are discussed.

Figure 1 Structure of the simulation model



2.1 The network module

In the network module, traffic flows are assigned to the network so as to reach a fixed demand user equilibrium. In this section, we define the elements of the network model, and discuss the properties of the user equilibrium.

Car trips are produced in a road traffic network G consisting of nodes $n \in N$ which are connected by links $(m,n) = a \in A$; $m, n \in N$. Demand for origin-destination pairs $d_{i,j}$, $i \in O \subseteq N$, $j \in D \subseteq N$ is fixed. A trip origin i is connected to a destination j by at least one path $r \in P_{i,j}$, the set of paths connecting origin i to destination j. A path is defined as a sequence of links. Links are congestible, so that the travel time cost depends on link flow: $c_a = c_a(f_a)$, with f_a the flow on link a. Congestion taxes t_a can be introduced on the link level, not on the path level. Equations (3.1) define the link cost, the link flow, and the path flow $q^r_{i,j}$. Link flows and path flows are restricted to be non-negative.

$$linkuser\ price\ p_a = c_a\left(f_a\right) + t_a, \frac{\partial c_a}{\partial f_a} = c'_a \geq 0$$
$$where\ f_a = \sum_{i \in O} \sum_{j \in D} \sum_{r \in P_{i,j}} d_r^a q_{i,j}^r = f_a \text{ for } d_r^a = 1 \Leftrightarrow a \in r,\ zero\ otherwise. \quad (3.1)$$

The user equilibrium on the network is the situation in which no cost-reducing path switching is possible for any origin-destination pair, from the user point of view. Consequently, the user cost on used paths is minimal and equal, and not higher than costs on unused paths. However, as user costs do not coincide with social costs because of the congestion externality, the user equilibrium is socially inefficient. The specific type of network equilibrium which we compute is the Wardrop equilibrium. The equilibrium is the result of unilateral decisions of network users, who take the decisions of other users as given, under complete information (see e.g. Nagurney, 1993). The Wardrop equilibrium on the network must also satisfy the conservation of flow condition, which stipulates that no traffic flows are lost on the network.

The combination of the conservation of flow condition and the equilibrium requirement directly leads to the complementarity formulation of the network user equilibrium problem. The conditions can be expressed in two ways, by using paths or through a multicommodity approach. The path-flow format maybe is the most intuitive one. It is given in equations (3.2) and (3.3). Equations (3.4) and (3.5) define the multicommodity format. The latter approach is used in the network module, for reasons set out below.

<u>Complementarity – path flow format</u>:

- Conservation of flow:

$$\sum_{r \in P_{i,j}} q_{i,j}^r = d_{i,j} \text{ and } \sum_{i \in O} \sum_{j \in D} \sum_{r \in P_{i,j}} d_r^a q_{i,j}^r = f_a \quad (3.2)$$

  The conditions say that the sum of all path-flows for an origin-destination-pair equals the demand for that origin-destination-pair, and that the link flow consists of flows on all paths using the link.

- Network equilibrium:

$$\forall r \in P_{i,j} : 0 \leq (p_r - s_r) \perp q_{i,j}^r \geq 0,\ \forall i \in O, j \in D$$

where $p_r = \sum_{a \in A} d_r^a p_a$ is the user price of path r,

and s_r is its minimal value (3.3)

The $\perp$ symbol indicates that at least one of the adjacent inequalities must hold as an equality.

  This condition requires user costs on all paths carrying positive flow for a given origin-destination-pair, to be lower than or equal to the costs on zero-flow paths. The set of used paths is determined during the solution. If it is exogenously given, the complementarity problem reduces to a system of equations (Ferris and Munson, 2000).

The computational implementation of the path-flow format requires implicit or explicit enumeration of all paths for all origin-destination-pairs. Since this is not computationally straightforward, even in networks of moderate size, the multi-commodity format, described next, is more practical (Ferris et al., 1999).

Mixed complementarity – multicommodity format:

- Conservation of flow:

$$\sum_{n:(m,n)} q^j_{(m,n)} - \sum_{n:(n,m)} q^j_{(n,m)} = d^j_m,\ \forall (m,n) \in A,\ j \in D \tag{3.4}$$

  The conservation of flow condition stipulates that the difference between the flow leaving a node for a given destination and the flow coming into the node for that destination, is the demand in that node for the destination.

- Network equilibrium:

$$\left[c_{(m,n)}\left(f_{(m,n)}\right) + s^j_n\right] \geq s^j_m \perp q^j_{(m,n)} \geq 0,\ \forall (m,n) \in A,\ j \in D \tag{3.5}$$

where s^j_n is the minimal cost to reach j from node n

  The Wardropian equilibrium condition ensures that, if there is positive flow for destination *j* along link *(m,n)*, this flow is realised at minimal cost.

Other approaches

The literature contains alternatives to the complementarity formulation, for the representation of the Wardropian network equilibrium problem. The most important ones are the optimisation approach (e.g. Sheffi, 1985) and the variational inequality approach. In terms of the level of generality, Nagurney (1993) shows that the variational inequality approach encompasses the complementarity approach, and the latter encompasses the optimisation method. We briefly discuss the optimisation approach.

If the link cost functions are integrable, and when the interaction between network links is symmetric ($\frac{\partial c_a}{\partial f_b} = \frac{\partial c_b}{\partial f_a},\ \forall a \neq b \in A$)[16], an equivalent minimisation problem exists for the complementarity format. This is known as the Beckmann transformation, on which earlier network equilibrium models are based. The first-order conditions of this programme produce the Wardropian network equilibrium condition. The equivalent minimisation problem in the multicommodity format reads as follows:

$$\begin{aligned} &\min_{q^r_{i,j}} \sum_{a \in A} \int_0^{f_a} c_a(w)\,dw \\ &\text{subject to} \\ &\sum_{j:(i,j)} q^k_{(i,j)} - \sum_{j:(j,i)} q^k_{(j,i)} = d^k_i,\ \forall (i,j) \in A,\ k \in D \end{aligned} \tag{3.6}$$

[16] Integrability is possible for the link cost functions in the present network. Also, the symmetry condition is satisfied, as the cross-derivatives are zero. So, the equivalent minimisation approach can in principle be applied.

The network equilibrium is unique in link flows when the link costs are increasing in flow. There is no uniqueness with respect to path flows.

The network module of the simulation model is based on the mixed complementarity – multicommodity format (equations (3.4) and (3.5)). The reasons are that:

- Solution of larger networks is more conveniently done in the multicommodity format, as this does not require explicit enumeration of all possible paths for all origin-destination-pairs. The downside is that the paths used for each origin-destination-pair are not part of the model output, as only the link flows to each destination is determined, without specification of the origin.[17] Reconstructing the paths on the basis of this information is not trivial.
- The complementarity format directly produces costs of origin-destination-pairs as part of the output, while this is not the case in the optimisation framework. The origin-destination-cost variable is required as an input in the demand module. We note that, as a side-benefit, the mixed complementarity algorithm is more robust than the nonlinear programming algorithm, for finding the network equilibrium.[18] The nonlinear programming formulation requires good initial values, and it usually fails when path switching (i.e. when paths which are set at zero usage in the initial values, should be used in the final equilibrium) occurs.

2.2 The demand module

The consumers' allocation process is represented as the maximisation of a nested CES utility function subject to a budget constraint. The budget is adapted for the particular redistribution rule of the congestion tax revenues. The application in the next section assumes lump sum redistribution. A basic modelling choice is whether a one consumer structure is used, or whether a consumer is associated with every trip origin in the network. The structure of the simulation model accomodates both choices. In this paper, we rely on the one-consumer formulation. The reason is that this allows us to find the first best pricing solution by iteration, since first best implies marginal social cost pricing –or Pigouvian taxation- on all network links. This is not necessarily the case in a multiple consumer context (Van Dender, 2001b).

The structure of the utility tree depends on the origin-destination matrix at hand (see section 3 for an example). Commodities at the lowest level of the tree are origin-destination pairs, possibly with further distinctions between trip purposes (e.g. work, shopping) and time periods (e.g. peak, offpeak). The consumer optimum is computed through a non-linear programming algorithm.

2.3 Simulation versus optimisation

The simulation model allows the assessment of any given link tax scheme, in terms of economic efficiency or welfare. A grid search approach can be used to identify second-best partial pricing schemes. The implementation of these grid search methods is feasible within reasonable amounts of computer time. Nevertheless, a more efficient approach would be to construct a second-best

[17] Alternatively, the model can be re-formulated so that link flows from each origin are computed, without knowing the final destination.

[18] However, both the mixed complementarity algorithm and the nonlinear programming algorithm are much less efficient than specific traffic network equilibrium algorithms.

optimisation model, which finds the optimal taxes in the optimal subset of links, where the size of the subset is a fixed parameter.

In such a second-best optimisation model, the (social welfare) objective needs to be maximised subject to the network equilibrium condition. This type of problem is known in the operations research literature as a mathematical programme with an equilibrium constraint (MPEC, e.g. Luo et al., 1996). The equilibrium constraint here refers to the network equilibrium requirement. It may also refer to, e.g., a consumer or a market equilibrium, as in optimal taxation problems. If an equivalent optimisation programme is used to find the (network) equilibrium (cfr (3.6)), the MPEC becomes a bilevel optimisation programme.

In the MPEC, the link tolls are design variables and the path-flows are state variables. The equilibrium constraint is parameterised in the design variables. This clarifies that the MPEC is an instance of a Stackelberg model, where government is the leader and sets the link taxes, and network users are the followers who take the link taxes as given. In case the user equilibrium problem has a unique solution (in link flows), the MPEC can be rewritten as a one-level implicit programme (Dirkse and Ferris, 1997). However, in general the objective of the implicit programme is non-smooth, because of the complementarity condition which characterises the network equilibrium problem. This non-smoothness is the principal computational difficulty for MPEC-type of problems. It makes the problem combinatorial, as the space of possible solutions consists of possibly disjoint subsets. The solution space is therefore possibly discrete, and certainly non-convex. Algorithms for fixed demand versions of the MPEC in (7) have been proposed and tested (Dirkse and Ferris, 1998; Larsson and Patriksson, 1998), but algorithms for more general variable demand problems are not available to our knowledge.

The mathematical programme for the MPEC can be seen as the primal or dual form of an optimal taxation programme, but only when the complementarity condition is left out of the equilibrium constraint. The dual form is used in Van Dender (2001b). Using a single representative consumer model, and taking the path-flow format of the network equilibrium problem, the Lagrangian for the primal form is as in (3.7). The utility structure allows for substitution between origin-destination pairs, and for imperfect substitution between paths (so it contains the Wardrop equilibrium as a special case).

$$\begin{aligned}
\Im = {} & U\left[x, q_{i,j}^{r}; i \in O, j \in D, r \in P_{i,j} \right] \\
& + g\left[Y - x - \sum_{i \in O} \sum_{j \in D} \sum_{r \in P_{i,j}} \left(\sum_{a} d_{r}^{a} c_{a}(f_{a}) q_{i,j}^{r} \right) \right] \\
& + \sum_{i \in O} \sum_{j \in D} \sum_{r \in P_{i,j}} y_{i,j}^{r} \left[\frac{\partial U}{\partial q_{i,j}^{r}} - l \left(\sum_{a} d_{r}^{a} (c_{a} + k_{a} t_{a}) \right) \right]
\end{aligned} \tag{3.7}$$

Utility of the representative consumer is maximised subject to:

- Societies' budget constraint (multiplier γ). Within the structure of the model, this constraint is interpretable as the production feasibility constraint, where Y is the generalised budget and x is the numéraire composite commodity.

- The consumer network equilibrium constraints for each path for each origin-destination pair (multipliers $y^r_{i,j}$). The instruments are link taxes, but these are not available for all links. Parameter κ_a indicates whether a link can be taxed or not *(k_a=1 or k_a=0)*. The taxes appear in the consumer equilibrium constraint, since they affect the consumer's decisions. They do not appear in societies' budget constraint, because of the assumption of lump sum redistribution.

The consumer equilibrium constraint on the market for other commodities says that $U_x=\lambda$, where λ is the private marginal utility of income. This constraint does not appear in equation (3.7), because it is automatically satisfied when the budget constraint and the equilibrium constraints on the transport markets are satisfied. Note that this implies $\lambda = \gamma$.

The primal form can be used to develop link selection mechanisms for second-best network pricing schemes. In particular, the multipliers of the network constraints reflects the shadow cost of the network requirement. Link taxes should probably be introduced where these shadow costs are large (see appendix 1 for a more detailed explanation; see also Verhoef, 2000). However, this approach is only valid when the set of used links is taken to be fixed. This assumption may be reasonable for policies of marginal network pricing reform, but not for first- or second-best optimisations.

To summarise, if the complementarity condition is unimportant in the network equilibrium problem, standard nonlinear programming algorithms can be used to solve the problem. Essentially, the complementarity condition becomes unimportant when the set of used paths is exogenously given.[19] This does not exclude that path-flows become zero in a counterfactual equilibrium. It is not possible, however, that previously unused paths start to carry positive flow. If the complementarity condition is important, no standard tools are available for optimisation. As will be illustrated in the next section, absence of the complementarity problem is not a generally valid assumption in larger networks.

3. Model implementation: a simplified network for Namur

3.1 Network and reference equilibrium

Figure 2 represents the transport network. It is a stylised representation of the transport network of a small regional centre town (Namur, Belgium), spanning a region of approximately 4.5km by 3km. The network topology and link cost functions are derived from a more detailed network for this area (see appendix 2 for more detail). The network consists of 11 nodes and 30 real links. The links shown in the figure are bi-directional, links CX and XC are virtual links.[20] Nodes B and E are in the city centre. The centre is surrounded by a (quasi-)ringroad, which gives access to the regional road network.

Figure 3 shows the shape of the link cost functions. This is a standard form in network assignment models, exhibiting low sensitivity of travel time to link flow at low volumes, and a very high sensitivity at volumes close to link capacity.

[19] This assumption is used in the application in Van Dender (2001b), since there all paths are always used.

[20] Node X is a virtual node, introduced in order to allow the representation of two parallel links between C and J, by the virtual links CX and XC.

Table 1 is the reference demand matrix, specifying the number of passenger car trips between all origin-destination-pairs (42 in total) during the morning peak period, for an average working day in 2000. It is derived from the 1990 demand matrix for the larger Namur network (see appendix 2 for more detail). The trip origins during the morning peak are the network entry nodes A, C, D, F, G and H. All network nodes except I, J and X,[21] serve as trip destinations. Other time periods are neglected in the application.

Figure 2 Network topology

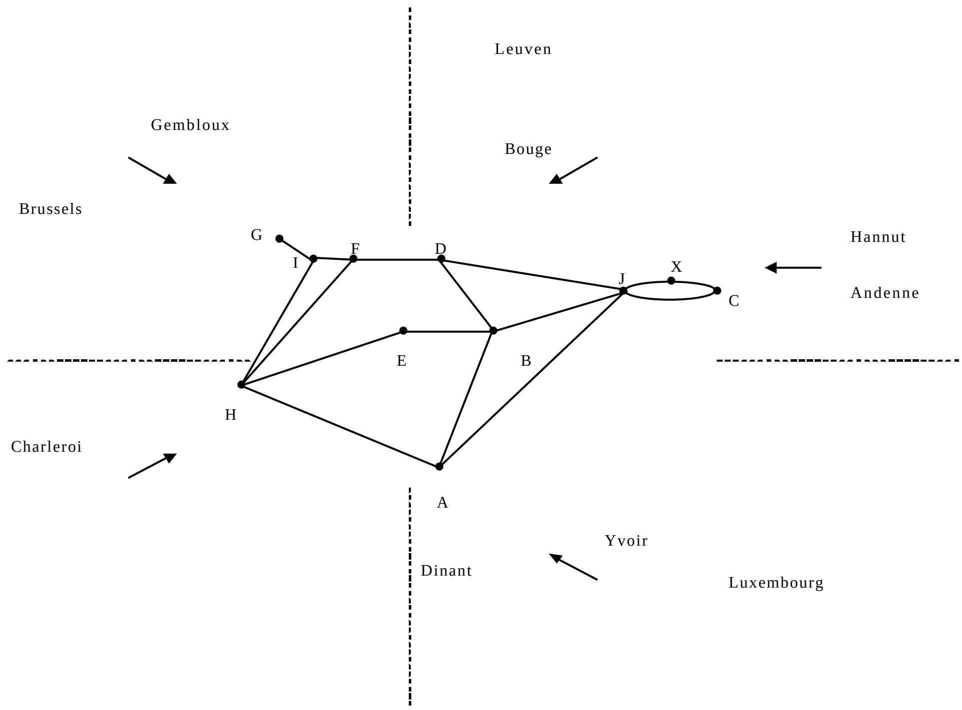
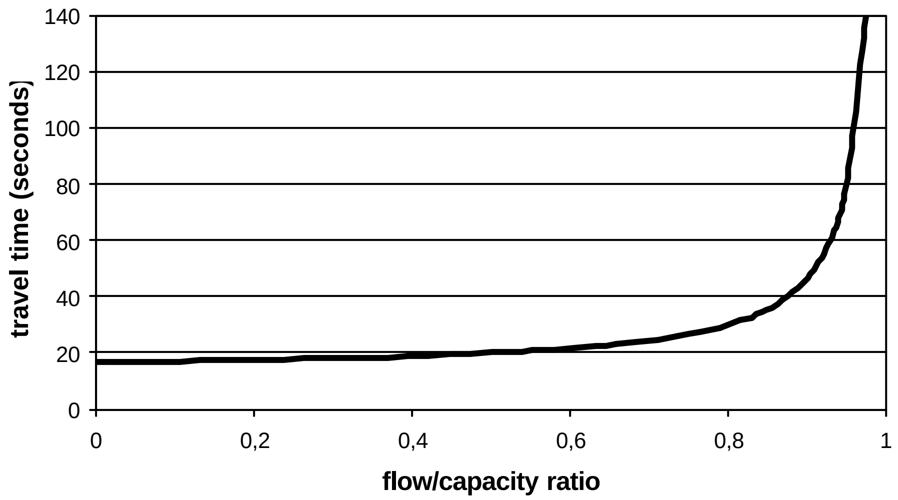


Figure 3 Representative link cost function



[21] Nodes I and J are pure connectors. X is a virtual node.

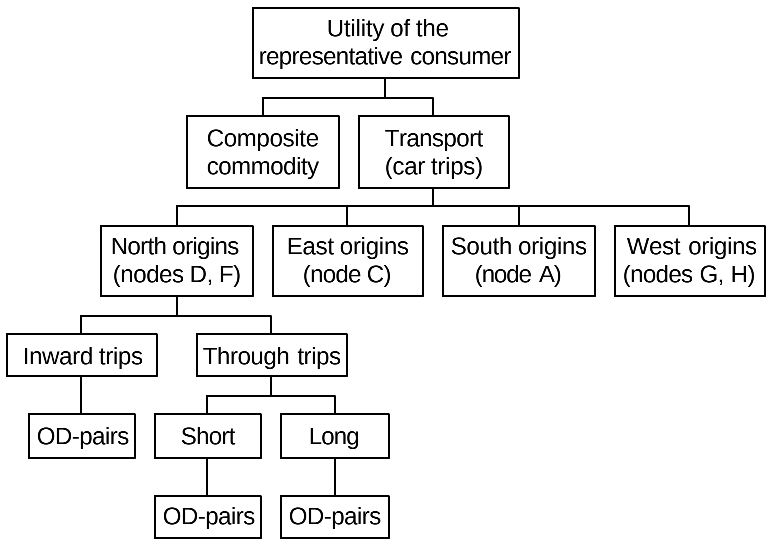
Table 1   Reference demand matrix, morning peak period

O\D	A	B	C	D	E	F	G	H	Total
A		1,771	1,245	558	454	302	616	1,409	6,354
B									
C	1,213		1,383	968	655	541	816	893	6,470
D	698	1,013	533		458	467	693	638	4,499
E									
F	177	404	121	164	159		168	199	1,391
G	664	883	389	529	403	298		653	3,818
H	1,020	1,110	493	627	1,110	421	719		5,500
Total	3,773	6,564	2,781	2,846	3,238	2,028	3,012	3,792	28,033

Table 2   Spatial distribution of trip origins and destinations

	Share of origins	Share of destinations
South	23.1%	13.5%
West	21.0%	24.3%
North	22.8%	17.3%
East	33.1%	9.9%
Central	0	35.0%

Figure 4 Structure of the utility tree



In the present application, the spatial distribution of trip origins and destinations forms the only basis for the structure of the utility tree. Figure 4 develops part of the structure. Analogous breakdowns to those of the north origins are made for the other origins. Node A is the south node, node C the east node, nodes D and F are the north nodes, nodes G and H the west nodes, and nodes B and E the central nodes. Trips with a central destination are inward trips, and trips with other destinations are through trips. A further distinction is made between 'short' and 'long' trips for through trips originating in the north and the west nodes. A short trip has its origin and its destination in the north or west respectively. Long trips have their destinations in other quarters. The spatial distribution of trips is summarised in table 2. City centre trips (towards B and E) represent only 35% of all trips, reflecting the observation that the Namur network is used intensively for through trips. Except for the dominance of east origins, trip origins are distributed rather uniformly,. Through trip destinations are mainly in the west quarter.

The reference network equilibrium is characterised by moderate and unevenly distributed congestion. For 61% of all links, the marginal external congestion cost is below 0.1 Euro.[22] The marginal external cost on the remaining links is in table 3. The two most congested links are DF and FD, on the north part of the ringroad (this is consistent with the larger network model for Namur). That DF is subject to the strongest congestion is in line with the spatial distribution of trips, as it is an east-west oriented link[23]. The other congested links are in different parts of the network, bearing no specifically strong relation to the reference demand configuration.

Table 3 Marginal external congestion costs in the reference equilibrium, Euro

Link	Marginal external congestion cost
DF	1.18
FD	0.77
HE	0.52
XJ	0.40
IG	0.37
JC	0.32
GI	0.31
DB	0.24
AH	0.17
JD	0.16
HA	0.11
JB	0.10

3.2 First best pricing

Given the assumption of a single representative consumer, first-best pricing is characterised by marginal social cost pricing on all network links (Van Dender, 2001). We assume that congestion tax revenues are redistributed to the consumer in a lump sum way. Using a calibrated demand module with low subsitutability between origin-destination-pairs (see section 2.4 for a discussion of the role of elasticities of substitution), the first-best is found by iterative implementation of the simulation model. As starting values for the optimal link taxes, the marginal external congestion costs of a system optimum[24] at 95% of the reference demand levels were used.[25]

Total demand for trips decreases by 6.4%. City-centre bound trips decrease a bit more (6.8%) than through trips (6.1%). This is related to the average length of both trip types, which is higher for through trips. The spatial distribution of the reduction is uniform for centre-bound trips, while the reduction is larger for through trips originating in the north and west, and smaller for through trips

[22] A value of time of 7.7 Euro/hour was used (Proost and Van Dender, 2001).

[23] This is a loose statement, as spatial distributions of origins and destinations say nothing about relative link capacities. The capacities in the network are relatively uniform however (see appendix).

[24] The system optimum computes the aggregate cost minimum for fixed demand levels, through nonlinear optimisation. It is equal to the first-best optimum in terms of network assignment, at least when the correct demand levels are specified.

[25] This reduces the risk that an equilibrium is found where prices equal marginal social costs, but which is not the first-best optimum. Starting from system optimum values is especially important when path switching occurs.

originating in the south. On the origin-destination-level, demand decreases range between 3.4% and 8.8%. No demand increases are observed.[26] Non-transport consumption increases by 0.77%. First-best pricing results in an efficiency improvement of 0.21% with respect to the reference equilibrium.

Table 4 compares the congestion pattern in the reference equilibrium and in the first-best. It is clear that the first-best pricing mechanism reduces the large initial congestion costs most. The result is a much more homogenous spread of congestion over the network than in the reference equilibrium. Increases in congestion on some links is not excluded, but it is limited to links with relatively low initial congestion.

The set of used paths changes between the reference equilibrium and the first-best solution, as link BD carries a positive flow in first-best, while it is not used in the reference equilibrium.[27] This is an illustration of the fact that searching for first-best (or second-best) solutions within the reference set of paths is not sufficient. For reasons set out in section 2.1, it is hard to say for which origin-destination pairs the set of used paths changes. Anticipating on the results of the partial pricing analysis (section 3.3), it can be inferred that previously unused paths become active for eastern origins (node C) and mainly western destinations (nodes G and H).

Table 4 Reference and first-best marginal external congestion costs, Euro

Link	Marginal external congestion cost Reference	Marginal external congestion cost First best	
DF	1.18	0.17	-85%
FD	0.77	0.19	-76%
HE	0.52	0.32	-39%
XJ	0.40	0.17	-56%
IG	0.37	0.17	-55%
JC	0.32	0.21	-33%
GI	0.31	0.15	-53%
DB	0.24	0.14	-42%
AH	0.17	0.07	-12%
JD	0.16	0.12	-53%
HA	0.11	0.12	+7%
JB	0.10	0.13	+12%

3.3 Partial network pricing

The simulation model can be used to assess the efficiency impact of any link or node pricing scheme. A more interesting question regards its performance in finding second-best partial pricing mechanisms. A second-best partial pricing mechanism sets optimal taxes on the optimal subset of links of any given size. One way of finding the second-best n-link pricing scheme is to perform a grid search on all possible n-link schemes. This clearly is a demanding task, even in the present 30 link network. Alternative methods have been proposed, based on the out-of-equilibrium shadow costs of

[26] This is not a general result. Exercises with smaller networks have shown that demand increases for some origin-destination pairs are possible in first-best.

[27] As was mentioned in section 2.1, the network module gives no direct information on which paths are actually used for any given origin-destination pair. However, when the flow on a link switches from zero to positive, the set of paths clearly changes.

network equilibrium constraints (see appendix 1). As argued in section 2.4, this approach is valid only for marginal tax reform, as it assumes a fixed set of used paths.

Here a much simpler heuristic (or: common sense) method is presented, based on a relatively narrow grid search. First, we compute the welfare gains of one-link pricing schemes, by stepwise increases of the link taxes, with a step size of 0.05Euro. Ranking the one-link schemes according to their performance produces a subset of links, in which two-link combinations are assessed. The starting values of the two-link schemes are taken from the one-link optimal taxes. Next, a subset of three-link schemes is evaluated, where the subset is constructed on the basis of highly performant one-link and two-link schemes, and on the basis of the size of link interactions. Choosing larger subsets proceeds analogously. There is no absolute guarantee that this procedure leads to the optimal subset of *n* links, but we are relatively confident with respect to this issue (see below). At any rate, the heuristic provides a quick way of selecting highly effective *n*-link pricing schemes.

The procedure works best in networks with heterogenous congestion in the reference case. In case congestion is homogenous to start with, the procedure will have difficulties in discriminating between different pricing schemes. However, when congestion is homogenous, the gains which can be expected from partial pricing mechanisms are small.[28] In case congestion is homogenous and excessive, aggregate demand reduction is the preferable policy option. This policy goal is not served well by partial pricing schemes. Other measures (e.g. parking charges or area licensing) become relatively attractive (Van Dender, 2001b).

3.2.1 One-link pricing

In a homogenous 30-link network with little network interaction and low subsitutability in the demand module, a tax on one link will generate approximately 3.3% of the first best efficiency gain. The performance of optimal one-link schemes in our network is quite different, as they produce between 0 and 33% of the first-best efficiency gain. Table 5 shows the best one-link schemes, and the share of the first-best gain which is achieved.

Table 5 Performance of optimal one-link pricing, relative to first-best efficiency gain

Link	Optimal tax (approximation), Euro	Marginal external congestion cost, Euro	Tax / marginal external congestion cost	% first best efficiency gain	Reference marg. external congestion cost, Euro
CJ	1.10	0.058	19.0	32.7	0.094
DF	0.15	0.187	0.8	29.3	1.180
JD	0.10	0.053	1.9	23.8	0.161
GI	0.70	0.117	6.0	23.7	0.311
FD	0.20	0.156	1.3	19.7	0.768
IG	0.70	0.125	5.6	19.6	0.373
AH	0.15	0.094	1.6	15.8	0.167
JC	1.00	0.173	5.8	15.8	0.318
DJ	0.10	0.016	6.1	11.2	0.021
FI	0.20	0.033	6.0	8.6	0.037
DB	0.15	0.163	0.9	7.5	0.241

28 The degree of heterogeneity can be checked by comparing the user equilibrium and the system optimum in a network model.

Optimal one-link pricing schemes are potentially very effective in terms of efficiency gains, as they generate up to 1/3 of the maximal efficiency improvement. The process generating the efficiency gain differs across links, however. This is explained for the three best one-link schemes, CJ, DF and JD.

The optimal tax on link CJ is much larger than the marginal external congestion cost on link CJ. Also, the link does not appear in the list of most congested links in the reference equilibrium (table 2). This indicates that the tax is mainly used to achieve an aggregate *demand reduction*. Demand for city centre trips originating in node C decreases by 7.2% (6.4% in first-best), while demand for through trips starting in node C drops by 3.5% (6.1% in first-best). Demand reductions for other destinations are limited (-0.8%, as compared to ca. –6% in first-best). Taxing link CJ only achieves a demand reduction for the dominant trip origin. Trips originating in C use the strongly congested part of the network, so that the demand reduction generates a substantial efficiency improvement.

The optimal tax on link DF is close to the marginal external congestion cost. Link DF is the link with the highest marginal external congestion cost in the reference equilibrium. The single link tax on DF is mainly used to selectively decrease marginal external congestion costs in the network, or to *improve network assignment*. The aggregate demand reductions are small (between 0 and 1.1%). The reduction of flow on DF is large (21%). From the link flow changes, it can be inferred that trips in the east-west direction are redirected from paths in the north (strongly congested, and using link DF) to paths through the centre (strong link flow increases on links JB, BE, EH) and the south (strong link flow increases on links JA and AH). This type of reassignment is also found in the first-best pricing system, but to a smaller extent.

The optimal tax on link JD also generates a more efficient network assignment, while demand reductions are limited to approximately 0.3%. The type of re-assignment is similar to the first-best case, as in the previous case. This could be expected, as link JD belongs to the north path connecting the eastern origins and western destinations. However, in contrast to the first-best and the single tax on DF, optimally taxing link JD leads to *a change in the set of used paths*. This can be inferred from the fact that link BD carries a positive (and substantial) flow in the user equilibrium given the optimal tax on JD, while it is not used in the reference user equilibrium. The reason is that some trips from the east to the west avoid the taxed link JD, and then use link BD to return to the northern path leading to western destinations. This phenomenon explains why taxing link JD performs less well than taxing link DF. This type of discrete path-switching again illustrates what makes the second-best optimisation problem combinatorial. An integrated second-best optimisation model will have difficulties in finding the optimal tax on link JD, as the search process needs to consider previously unused paths. This is a complex problem, as the change in the set of used paths occurs for some tax levels on some links, but not for others. Moreover, the path change does not lead to a discontinuity in the resulting efficiency gain. The problem is more likely to occur in partial pricing schemes than in first-best pricing, as partial pricing tends to cause relatively large relative price changes across paths.

3.2.2 Two-link pricing

The efficiency gain of optimally taxing two links can be expected to be smaller than or equal to the gain from taxing both links separately (subadditivity in efficiency improvement).[29] This implies that a

[29] This claim does not hold in very small networks, e.g. where two single-link paths connect one origin-destination pair. In that case, the two links are perfect substitutes and simultaneously taxing

reasonable subset of two-link pricing schemes can be constructed using information on the gains from one-link pricing systems, as two-link schemes consisting of poorly performing one-link taxes can be neglected. Furthermore, the optimal one-link taxes provide a good starting point for searching the optimal taxes in the two-link system.[30] Specifically, the optimal taxes in the two-link system can in most cases be expected to be lower then or equal to the taxes in the one-link systems. On the basis of these observations, we perform a limited grid search. The efficiency gains of two-link systems consisting of effective one-link systems are computed, for tax levels in the neighbourhood of the optimal one-link taxes. The grid search then proceeds in the direction of the highest efficiency gain, untill the optimal link tax levels are found.

Table 5a presents the efficiency gains from effective two-link pricing schemes, in terms of the share of the first-best gain which is achieved. The diagonal contains the performance of the one-link schemes. Table 5b shows the degree of subadditivity, by the ratio of the gain from optimal combined taxation of both links and the sum of the optimal single link taxation gains. A lower ratio indicates a stronger degree of subadditivity. A ratio equal to 50% implies that the two-link system performs in the same way as a tax on either link, while a ratio equal to 100% indicates that the link taxes operate in a completely separable way.

Table 5a Performance of two-link pricing schemes, as share of the first-best efficiency gain

First link		Second link					
		CJ	DF	JD	GI	FD	IG
	CJ	32.7	54.2	49.4	43.3	47.5	36.7
	DF		29.3	29.2	49.7	46.5	40.7
	JD			23.7	44.9	39.4	39.2
	GI				23.7	34.3	35.5
	FD					19.7	38.2
	IG						19.6

Table 5b Subadditivity of two-link pricing schemes

First link		Second link					
		CJ	DF	JD	GI	FD	IG
	CJ	100	87.4	87.4	76.7	90.6	70.2
	DF		100	54.9	93.7	95.0	83.1
	JD			100	94.5	90.8	90.4
	GI				100	79.2	82.0
	FD					100	97.3
	IG						100

The best two-link scheme (CJ, DF) generates 54.2% of the first-best efficiency gain. It consists of links CJ and DF, the two links which perform best among the one-link pricing schemes. As noted in section 2.2.1, taxing link CJ mainly works to reduce demand originating in the eastern quarter, while taxing link DF mainly improves the efficiency of the assignment. It is therefore not surprising that taxes on both links are complementary. The degree of subadditivity is relatively limited, as the welfare gain of combined taxes equals 87.4% of the gain of separate taxes. The optimal link tax on CJ

both may perform in a super-additive way, compared to separate taxes on either link. Such cases are of little importance in the present network, however.

[30] The validity of this statement was checked by an extensive grid search for a number of two-link combinations. The results show that efficiency monotonically increases from all directions, towards the second-best optimum. This is the case independent of the degree of link interaction.

in the two link scheme is 1Euro (compared to 1.1Euro in the one-link scheme). The optimal tax on DF is 0.1Euro in both cases.

The two-link scheme consisting of CJ and JD compares to CJ and DF in the same way as the one-link schemes DF and JD. That is, a tax on JD is a relatively good substitute for addressing the congestion problem on link DF. Taxing DF and JD simultaneously is subject to strong subaddivity, which could be expected on the basis of the results for the one-link schemes: both taxes essentially tackle the same problem of excessive congestion in the northern paths for east-west trips. Most other two-link schemes perform quite well, which is explained by the limited amount of subaddivity. The taxes deal with congestion problems which are separable to a large degree, so that each tax basically acts as a one-link tax.

3.2.3 Three- and four-link pricing

As the number of taxed links increases, it can be expected that the degree of subaddivity rises, or in other words that there are decreasing returns to expanding the set of taxed links. The selection of a reasonable subset should therefore take account of the information on subadditivity contained in the one-link and two-link analyses. The expected efficiency gain of the three-link systems (a,b,c) was computed on the basis of the two-link schemes (a,b), (b,c) and (a,c), making a correction for the sub-additivity of each link with each pair of other links.

The optimal taxes were computed for the three-link schemes with an expected gain above that of the best available two-link system. As starting points for the optimal taxes, the optimal values for the two-link systems were used. The first two columns of table 6 contain the three-link systems which effectively produce as much, or more, welfare gains than the best available two-link system.

Table 6 Gain from optimal three- and four- link schemes

Three-link systems		Four-link systems	
Links	% of first best efficiency gain	Links	% of first best efficiency gain
CJ, DF, FD	68	CJ, DF, FD, GI	76
CJ, DF, GI	65	DJ, DF, DJ, GI	74
CJ, FD, JD	62	CJ, DF, FD, IG	71
DF, FD, GI	60	CJ, FD, GI, JD	69
CJ, GI, JD	60	CJ, DF, FD, FI	69
DF, FD, IG	57	CJ, DF, DJ, FD	69
CJ, DF, IG	57	CJ, DJ, GI, JD	67
DF, GI, IG	56		
DF, FD, GI	56		
CJ, FD, GI	55		
CJ, DF, JD	54		
CJ, IG, JD	53		

The best three-link system is (CJ, FD, FD). It contains the two links of which the best two-link system consists (CJ, DF), and link FD. The first two links tackle the congestion problem in the east-west direction. Link FD is directed towards the congestion problem in the opposite direction. This illustrates that a two-link system is sufficient (in a second-best sense) for correcting both the demand levels and assignment decisions for the most problematic trip direction (east-west). Larger partial pricing systems start to tackle separable congestion problems within the network. This large degree of separability is illustrated by the ratio of the relative welfare gains, which indicates a limited amount of subadditivity. Let G be the efficiency gain of the partial pricing scheme in subscript:

$$G_{(CJ,\, DF,\, FD)} / [G_{(CJ,\, DF)} + G_{FD}] = 0.92.$$

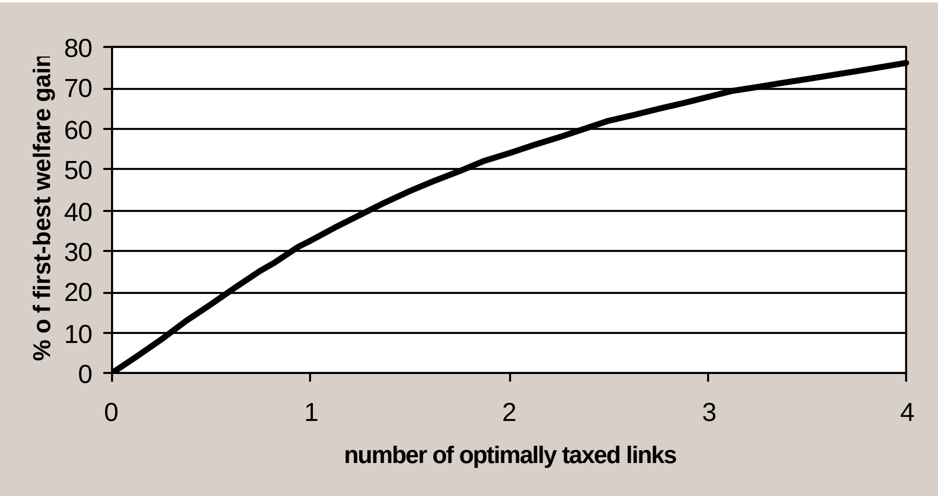
A similar procedure produced an estimate of the welfare gains from the optimal four-link pricing scheme, which consists of links (CJ, DF, FD, GI) and which produces 76% of the first-best welfare gain. Again, the four-link scheme contains the links of the optimal three-link scheme, plus an additional link (GI). In the four-link scheme, GI and FD address the west-east congestion problem, while CJ and DF take care of the east-west direction. Adding link GI brings a relatively small extra welfare gain as compared to the three-link system, because of the substantial degree of subadditivity with link FD (see table 2b). Other four-link schemes that perform at least as well as the best three-link system are in the last two columns of table 6.

3.2.4 Summary: the benefits of optimally taxing more links

Figure 5 shows the share of the first-best efficiency gain which is attained by partial pricing schemes of various sizes. The link taxes are optimal for the selected subset of taxes, and the selected subset is optimal among those analysed.

Two points are of interest. First, the share of the first-best gain is large for all partial pricing schemes. This of course is related to the fact that congestion is concentrated in a small part of the network, in the reference situation. Second, the incremental gains of expanding the set of taxed links rapidly decrease. This result is related to the size of the network. In a larger network, the decrease can be expected to be slower, as a larger number of (separable) sub-problems can occur. Nevertheless, it is interesting to note that a three-link system, in the thirty link network, is sufficient to correct the main congestion problem.

Figure 5 Share of first-best efficiency gain from optimally taxing 1 to 4 links



3.3 Partial pricing with high elasticities of substitution

The application in the previous section has assumed low elasticities of substitution between origin-destination-pairs, as well as between transport and non-transport commodities. This is consistent with standard practice in network modelling. This section briefly discusses the effect of allowing for substantial substitutability of origin-destination-pairs.

First, a larger degree of substitution between transport and non-transport consumption leads to larger demand reductions for transport (ca. 13% instead of 6%) and to larger first-best efficiency gains (0.30% instead of 0.20%). The larger demand reductions imply larger reductions in marginal external congestion costs. Second, partial pricing schemes aimed at demand reductions become more effective, by about 10%. The performance of partial schemes which mainly affect network assignment does not change by a lot. The optimal taxes tend to decrease. As the number of links in a partial scheme increases, the share of the first-best gain which is achieved converges between the low- and the high-elasticity case. Finally, we note that the convergence of the model is considerably slower when elasticities in the demand module increase.

4. Conclusion

The simulation model presented in this paper is a relatively simple, but effective tool for efficiency or welfare assessment of exogenously defined link pricing mechanisms in transport networks. It can be used to design effective partial pricing mechanisms, through a limited grid search approach. Although the second-best property of a partial pricing scheme can only be assessed with absolute certainty through a complete grid search, limited grid search methods work well when small subsets of links are considered.

Application of the model for a mid-size representation of the city of Namur (Belgium) shows that large proportions of the first-best efficiency gain can be attained through optimal taxes on a limited number of links. Specifically, taxing four links (out of thirty) produces ¾ of the maximal gain. The gain is achieved by a mixture of transport demand reductions and of improved network usage. It can safely be assumed that a network pricing system becomes more expensive when more links are taxed. Using a simple simulation tool like the one presented here can therefore contribute to the cost-benefit analysis of congestion pricing schemes.

Appendix 1 Network pricing reform: the primal optimal tax problem

This appendix repeats the problem of equation (3.7), and uses the first order conditions in (A.3.1) to show how the multiplier of the network equilibrium constraint contains information for the identification of welfare improving directions of network taxation reform. This is a generalisation of the theoretical analysis in Verhoef (2000). We refer to the same paper for a numerical application of the procedure.

From section 2.1, it is clear that the method only applies when the set of used paths is fixed. More precisely, it is not allowed that paths which are not used in the reference equilibrium, will be used after the tax reform. It is however possible that paths which are used in the reference equilibrium become unused after the reform.

$$\begin{aligned}\Im = {} & U\left[x, q^r_{i,j}; i\in O, j\in D, r\in P_{i,j}\right] \\ & + g\left[Y - x - \sum_{i,j}\sum_{r\in P_{i,j}}\left(\sum_a d^a_r c_a(f_a) q^r_{i,j}\right)\right] \\ & + \sum_{i,j}\sum_{r\in P_{i,j}} y^r_{i,j}\left[U_{q^r_{i,j}} - l\left(\sum_a d^a_r (c_a + k_a t_a)\right)\right]\end{aligned}$$

(A.3.1)

$$\frac{\partial\Im}{\partial x} = U_x - g = 0$$

$$\frac{\partial\Im}{\partial q^r_{i,j}} = U_{q^r_{i,j}} - g\left[\sum_a d^a_r (c_a + c'_a f_a)\right] + \sum_{k,l}\sum_{s\in P_{k,l}} y^s_{k,l}\left[\frac{dU^s_{k,l}}{dq^r_{i,j}} - l\sum_a d^a_r d^a_s c'_a\right] = 0, \forall r, i, j$$

$$\frac{\partial\Im}{\partial t_a} = \sum_{i,j}\sum_{r\in P_{i,j}} y^r_{i,j}\, l k_a = 0$$

Using the first order condition for the path flows $q^r_{i,j}$ and the consumer equilibrium condition (second constraint in the Lagrangian) allows to solve for any $y^r_{i,j}$, as in equation (A.3.2). As can be seen, the values of the multipliers are interdependent. They represent the marginal social cost of having to satisfy the consumer equilibrium constraint. By first order condition for the link taxes t_a, the sum of $y^r_{i,j}$ is zero, over those links on which a tax is possible (k_a=1). This means that, whatever the pricing restrictions are, the aggregate social cost of the consumer equilibrium constraints is minimised. Therefore, any positive $y^r_{i,j}$ is compensated by one or more negative $y^r_{i,j}$. These deviations imply that some unpriced congestion remains, which implies inefficient allocations over origin-destination pairs and/or over paths per origin-destination pair. Only in 1st best are all $y^r_{i,j}$ equal to zero. Note that the possibility of negative multipliers requires a Lagrangian formulation of the problem, instead of a Kuhn-Tucker form.

$$y^{r}_{i,j} = \frac{g\left[\sum_a d^a_r\left(c_a + c^{'}_a f_a\right)\right] - l\left[\sum_a d^a_r\left(c_a + k_a t_a\right)\right] + l\left[\sum_{k,l}\sum_{\substack{s\in P_{k,l}\\ \neq r\in P_{i,j}}} y^{s}_{k,l}\left(\sum_a d^a_r d^a_s c^{'}_a - \frac{\partial U_{q^s_{k,l}}}{\partial q^r_{i,j}}\right)\right]}{\frac{\partial U_{q^r_{i,j}}}{\partial q^r_{i,j}} - l\sum_a d^a_r c^{'}_a} \qquad \text{(A.3.2)}$$

It is clear from the first two terms in the numerator, that the shadow cost of the equilibrium constraint is increasing in the difference between the marginal external congestion costs and the paid tolls on the path. As is indicated by the third term of the numerator, the shadow cost decreases as flows from other paths (for the same or for other origin-destination pairs) are discouraged to use links in the path under consideration. The denominator says that the shadow cost becomes smaller as the price sensitivity of demand and/or the slope of the link cost functions in the considered path rises.

Next, and finally, combination of (A.3.2), and third first order condition in (A.3.1) and the requirement from the consumer equilibrium that $l = g$, leads to the expression (A.3.3) for an optimal link tax. Note that this expression describes the same tax as the one presented in Van Dender (2001b), equation (2.13) for the case where lump sum transfers are available. The present expression is somewhat less transparent, but clarifies the interaction between consumer preferences and network properties.

$$t_a = \frac{\sum_{i,j}\sum_{r\in P_{i,j}} \frac{\left[\sum_a d^a_r\left(c^{'}_a f_a\right)\right] - \left[\sum_{b\neq a} d^b_r k_b t_b\right] + \left[\sum_{k,l}\sum_{\substack{s\in P_{k,l}\\ \neq r\in P_{i,j}}} y^{s}_{k,l}\left(\sum_a d^a_r d^a_s c^{'}_a - \frac{\partial U_{q^s_{k,l}}}{\partial q^r_{i,j}}\right)\right]}{\frac{dU_{q^r_{i,j}}}{d\,q^r_{i,j}} - l\sum_a d^a_r c^{'}_a}}{\sum_{i,j}\sum_{r\in P_{i,j}}\left[\frac{d^a_r}{\frac{\partial U_{q^r_{i,j}}}{\partial q^r_{i,j}} - l\sum_a d^a_r c^{'}_a}\right]} \qquad \text{(A.3.3)}$$

The tax is a weighted average of 'components' (numerators' numerator) which belong to each path for each origin-destination pair in which the link is used. The weight is the difference between the change in marginal utility of an extra trip on the path belonging to the origin-destination pair, and the change in unit cost caused by that extra unit. This term is negative. Each 'component' consists of:

- The deviation between the marginal social cost of the links belonging to the path (valued at the private marginal utility of income), and the user costs for that path *except* the tax for which we are solving.
- The sum over all origin-destination pairs which share associated links, of the deviation between cost changes and marginal utility reactions caused by changes in the use of the current path for the current origin-destination pair. This deviation is valued by the multiplier belonging to each path and origin-destination pair. The sign of the third term can therefore not be determined theoretically.

The optimal tax will increase when less other links in associated paths are taxed or when they are taxed too low. It is also increasing in the slope of all link cost functions in the related paths. Interactions with other origin-destination pairs may to push the tax component for a particular path of an origin-destination pair up or down however.

Appendix 2 Construction of the network and the reference origin-destination-matrix

The network and the reference origin-destination-matrix used in this paper are based on more detailed network and demand data for Namur in 1990 (Cornélis and Van Dender, 2001). This appendix describes the connection between both data-sets.

A.1 The origin-destination-matrix

The original dataset distinguishes 26 network nodes, leading to 650 potential origin-destination-pairs. The origin-destination -matrix contains 502 positive entries, for a total demand of 26,053 trips during the 1990 morning peak.

First, the original origin-destination -matrix was collapsed into a (8x8) matrix, by neglecting origin-destination-pairs for which the reference demand is lower than 40 trips. This (8x8) origin-destination-matrix contains 20,415 trips, or 78% of the total in the (26x26) matrix. Removing the diagonal entries, which represent trips 'within' a node, leads to a total of 19,435 trips (75% of the original total). After this procedure, only trips from outside the city centre to the city centre (inward trips), or from the outside to the outside (through trips) are left over in the reduced origin-destination-matrix.

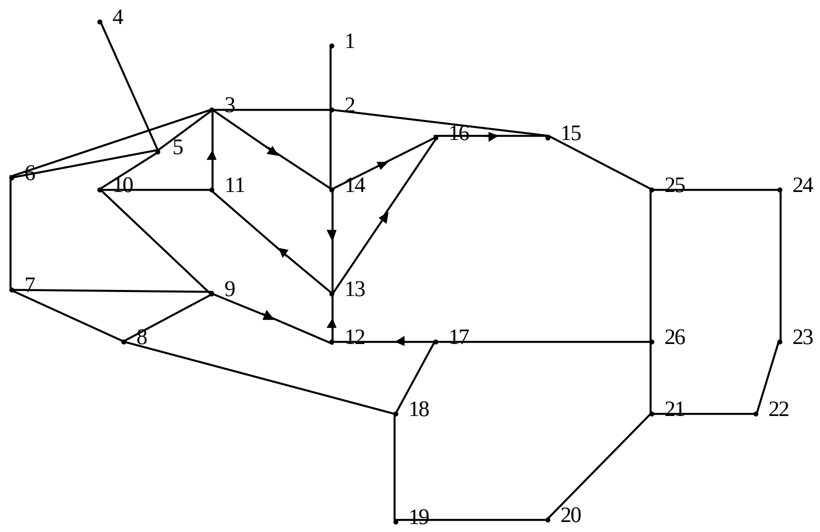
Next, the resulting (8x8) matrix is corrected, in order to re-introduce important parts of demand which have been deleted in the first step. This is done on the basis of spatial proximity between nodes in the original and the new matrix. In particular, demands for origin-destination-pairs (A,B), (A,H), (A,C) and (H,E) is doubled. The resulting (8x8) matrix now contains 21.564 trips (83% of the original total). The remaining 17% of the original trip demand mainly consists of trips from the city centre to non-centre destinations, and trips within the city centre. As these trips mainly use paths in the opposite directions from those retained in the reduced demand matrix, the interaction between those and the remaining trips can be expected to be small.

Finally, the 1990 demand matrix is converted to data for 2000, by applying a yearly growth rate of 2.6%, or a 30% traffic growth over the decade. The yearly passenger car transport growth rate as given in Eurostat-data is 2.2% for 1990 to 1998 (Eurostat, 2000). We used a slighly higher rate, as urban traffic grows faster than non-urban traffic. The resulting matrix is given in table 1.

A.2 The network graph

Figure A.1 displays the original network graph. Figure A.2 repeats the reduced network graph used in this paper. Table A.1 shows the incidence between links in both networks, and the resulting link time cost function for the reduced network. The latter are obtained from the original network by summing the link lengths and taking weighted averages of capacity parameters. The choice of the link incidence between both networks is based on the set of used paths for origin-destination pairs in the reference user equilibrium for the original network. The link incidence and the resulting link cost functions are in table A.1.

Figure A.1 Original network graph



links are bi-directional unless otherwise indicated

Figure A.2 Reduced network graph

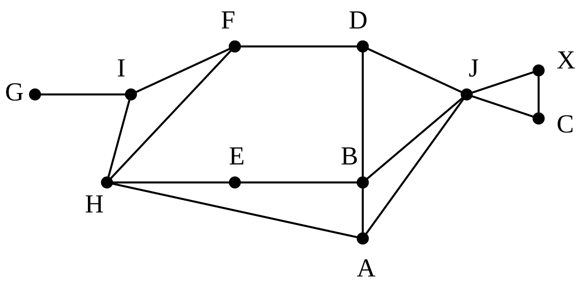


Table A.1 Connection between original and reduced network – cost function for the reduced network

Link in reduced network		Corresponding links in original network										
from	**To**	**1**	**2**	**3**	**4**	**5**	**6**	**Length (m)**	**Max speed (km/h)**	**Capacity (vehicles in morning peak)**	**Sensitivity to congestion**	**# of links in original network**
D	F	1,2	2,3					490	60	4250	0.15	2
F	D	2,1	3,2					630	60	4000	0.15	2
D	J	2,15	15,25					775	70	5000	0.2	2
J	D	15,2	25,15					775	70	5000	0.2	2
J	C	25,24	24,23					1860	75	3750	0.15	2
J	C	25,26	26,21	21,22				1330	66.7	5666.7	0.2	3
C	J	24,25	23,24					1860	75	3750	0.15	2
C	J	26,25	21,26	22,21				1330	66.7	5666.7	0.2	3
B	J	13,16	16,15	15,25				1330	63.3	3500	0.2	3
J	B	15,2	2,14	14,13				1015	63.3	3666.7	0.2	3
A	J	19,18	18,17	17,26				909	60	4666.7	0.2	3
J	A	18,19	17,18	26,17				909	60	4666.7	0.2	3
A	B	19,18	18,17	17,12	12,13			1251	50	5250	0.25	4
B	A	14,16	16,15	15,25	25,26	26,21	21,20	2265	65	3583.3	0.2	6
A	H	19,18	18,8	8,7				1867	60	3833.3	0.2	3
H	A	18,19	8,18	7,8				1867	60	3833.3	0.2	3
B	E	14,13	13,11	11,10				759	56.7	3500	0.2	3
E	B	10,11	11,3	3,14				874	60	3000	0.2	3
E	H	10,9	9,7					1092	55	3750	0.25	2
H	E	9,10	7,9					1092	55	3750	0.25	2
B	D	16,15	15,2	2,1				745	63.3	4333.3	0.17	3
D	B	1,2	2,14	14,13				795	60	3666.7	0.17	3
H	I	7,6	6,5					673	60	3250	0.2	2
I	H	6,7	5,6					673	60	3250	0.2	2
G	I	4,5						328	60	3500	0.2	1
I	G	5,4						328	60	3500	0.2	1
H	F	7,6	6,3					1267	60	3500	0.2	2
F	H	3,5	5,6	6,7				1178	60	3166.7	0.2	3
I	F	5,3						414	60	4200	0.2	1
F	I	3,5						505	60	3000	0.2	1

Bijlage 4

Economies of density and optimal transport pricing

1. Introduction

This paper looks into the role of economies of density in transit supply for the determination of efficient urban transport prices and service levels. We consider an urban transport system in which a private and a public transport mode are available in peak and off-peak hours. Government controls private transport taxes and public transport fares, as well as the supply of public transport. Increases in public transport demand can be met by increased occupancy rates and/or by increased service frequency. This is called service level optimisation. We analyse the properties of optimal prices and service levels, taking account of tax distortions outside the transport sector through the use of a marginal cost of public funds parameter.

The empirical analysis is based on a stylised model of the urban transport market, calibrated to datasets for Brussels and for London. When all transport prices are optimised, the impact of economies of density on optimal transit price levels and on welfare levels is found to be substantial. The application also suggests that, when car prices in the peak period are too low but cannot be raised, it can be desirable to have zero public transport prices during peak hours. The welfare improvement that can be reached with this second-best policy is quite limited however. First-best pricing is shown to lead to a budget surplus for the public transport operator, also when account is taken of economies of density. Optimising public transport prices when car prices are fixed leads to substantial deficits.

Section 2 reviews the literature, section 3 contains the theoretical analysis and section 4 discusses the applications. Concluding comments are in section 5.

2. Literature

Analyses of urban transport pricing find that currently important deviations from marginal social cost pricing exist. The transport pricing literature provides insights on both the causes and the possible justifications of such deviatons. First, they are partly due to non-internalised externalities of mainly congestion, air pollution, accidents and noise (e.g. Glaister and Lewis, 1978; Small, 1983; Proost and Van Dender, 2001). A second reason is the prevalence of explicit or implicit subsidies. Subsidies for transit are explicit; an example of implicit subsidies is the provision of free parking (Calthrop et al., 2000). Transit subsidies may be justified as second best pricing measures when there are restrictions on passenger car pricing. Inefficiencies related to labour taxation possibly make a case for provision of free parking (Calthrop, 2001). Third, economy-wide tax efficiency considerations and distributional issues may cause deviations from marginal social cost pricing (Mayeres and Proost, 2001). Despite the potential justifications for deviations from marginal social cost pricing, a general finding from most studies is that a transport pricing reform consisting of internalisation of externalities and the abolishment of subsidies, will yield substantial efficiency and welfare gains, if revenues are used appropriately.

A second strand of the literature focusses on the relation between public transport demand volumes, social costs, and optimal public transport prices. The social costs of public transport include the cost of the inputs supplied by the public transit operator and by the transit user (i.e. her time). This literature usually abstracts from the presence of competing modes (passenger cars). Increasing demand levels are found to imply decreasing average social costs. This is the case even when the transit operators' cost as such are subject to constant returns to scale. The reason is that higher demand levels allow for a higher frequency of service, which in turn leads to lower average waiting times at transit stops. As the returns to scale -on the social cost level- are largely dependent on the density of transit demand, the term economies of density (of demand) is often used. Using very detailed models of steady state and feeder route urban bus transport, Mohring (1972) finds considerable economies of density. Optimal supply conditions would combine increased frequency of bus service in both peak and off-peak hours with decreased fares. Turvey and Mohring (1975) discuss optimal pricing for transit, and show that marginal social costs are below average social costs so that cost recovery is not achieved. Nash (1988) constructs a transit cost function starting from the assumption of homogenously distributed demand for city centre trips, to be served by a number of radial bus routes. He shows that optimal service supply implies increasing bus sizes, increased network density and increased service frequency, as demand grows. All service characteristics adapt to demand according to a square root rule. Increased network density implies decreased walking and waiting times. Jansson (1997) constructs a stylised spatial model of transit costs within a circular city, showing that density of demand is the basic determinant of the returns to scale and that along an optimal expansion path any input may be chosen to exploit the returns to scale.

Three previous analyses are particularly relevant to our analysis, as they attempt to determine optimal price and service characteristics for multimodal urban transport systems. First, Viton (1983) combines a stylised spatial model of urban transport costs with a random utility demand model, to analyse the impact of efficient pricing. In case studies for the Bay Area and for Pittsburgh, he finds that optimal transit fares are well below current fares, that waiting times decrease and that the optimal modal share of transit is much larger than at present (up to 100% in some cases). Second, De Borger and Wouters (1998) use a model similar to ours, but containing a more explicit representation of the relation between transit supply in vehicle-kilometres, the number of vehicles used, and occupancy rates. The model contains a simpler representation of transport markets and no marginal cost of public funds. Moreover, it can only simulate first-best optima. In an application for Belgium, it finds that transit prices decrease by 61% (peak) to 84% (off-peak) in 1st best. Supply increases by 13% (peak) and 54% (off-peak). The reasons for these findings are the returns to scale in transit and the low off-peak marginal costs per passenger-kilometre. The increased attractiveness of transit also implies that optimal car prices increase less than in a situation without returns to scale. Third, Winston and Shirley (1998) look for an efficient urban transport system by setting prices equal to marginal social costs and optimising public transit service frequency, in a number of American urban areas. Their results indicate strong decreases in service frequency, combined with sharp price increases for all transport modes. Transit subsidies are virtually eliminated. The optimal modal share of transit is even lower than the reference share.

In comparison to these three studies, the present analysis emphasises the interaction between modal split, congestion externalities, economies of density and the cost of public funds. It will be shown that economies of density have a substantial effect on optimal transit prices and on modal split. The effect on optimal car prices is small, because the impact of economies of density is small compared to that of the internalisation of congestion externalities. Furthermore, both the presence of congestion and of a positive cost of funds have an effect on the characteristics of the optimal transit service. In particular, those factors tend to increase the attractiveness of high occupancy rates for transit as compared to a high frequency of service. Finally, optimal urban transport prices are shown to lead to increases in tax revenues from the transport sector, for the Brussels case study.

3. Theoretical analysis[31]

3.1 Model components

The model highlights the impact of economies of density on efficient transport prices and transit supply levels, and it clarifies the interaction between congestion, economies of density and revenue use. Government maximises welfare using private transport (car) taxes and public transport (transit) fares and supply levels as instruments. As the focus is on the conditions for efficiency, we use a single consumer with utility function $U = U\left(q_0, q_1, q_2, q_3, q_4\right)$ where q_0 is a composite commodity (numéraire). We use index $i=1,\ldots,4$ to designate transport commodities: q_1 is peak period car transport, q_2 is off-peak car transport, q_3 is peak period transit transport and q_4 is off-peak period transit transport. The unit for transport demand is passenger-kilometre. For car transport, occupancy rates are fixed such that changes in demand directly translate into changes in car flow ($X_i, i = 1,2$). For convenience we set car occupancy rates equal to one, such that $q_i \equiv X_i, i = 1,2$. The direct relation between passenger-kilometre and vehicle-kilometre does not hold in the case of transit, since demand increases may be met by increased occupancy rates (within technical limits). This is one form of economies of density, since increased occupancy rates lead to reduced average costs per passenger-kilometre. Abstraction is made from boarding and alighting costs and from discomfort externalities due to crowding (cfr. Kraus, 1991).

Transport prices are generalised prices, denoted p_i, $i=1,\ldots,4$. For car transport ($i=1,2$), the money cost to the consumer is the sum of the resource cost (c_i) and a tax or a subsidy (t_i). In the case of transit ($i=3,4$), the money price, or fare, is given by the tax. Time is converted to time cost by using a constant marginal value of time savings, which may in general differ between the transport commodities but which is normalised to one for convenience. The time cost consists of in-vehicle travel time cost (a_i) plus, for transit, waiting time cost at the bus stop (z_i).[32] The presence of congestion means that an additional vehicle-kilometre (i.e. increased traffic flow) by car or transit within a period (peak or off-peak) leads to increased travel times: $\frac{\partial a_i}{\partial X_j} > 0,\ i, j = 1,3$ and $\frac{\partial a_i}{\partial X_j} > 0,\ i, j = 2,4$. Note the assumption that both transit and cars contribute to congestion. This means that both modes share the road network.

---

[31] The general structure of the model is analogous to Proost and Van Dender (2001). The discussion therefore is brief, except for the transit pricing and supply conditions.

[32] It is standard practice to also include the walking time to the bus stop. As this cost is constant in our model, we loose no insight by abstracting from it.

Transit is therefore best understood as buses. Metro services using off-road infrastructure would be a special case with $\frac{\partial a_i}{\partial X_j} = 0,\ i, j = 1,3$. In general, cars and transit may contribute to congestion in a different way. To summarise, the generalised price is given by $p_i = c_i + t_i + a_i$ *(i=1,2)* and $p_i = t_i + a_i + z_i$ *(i=3,4).*

Tax revenues net of transit supply costs are valued at the economy-wide marginal cost of public funds $(1+\mu)$. The cost function of transit supply is linear: there is a fixed cost F and a fixed unit cost per vehicle-kilometre (which may differ between periods; c_i, i=3,4) times the number of vehicle-kilometres per period.[33] The per kilometre cost contains the leasing cost of the bus stock. As the peak period determines the capacity needs, the lease cost is included in the peak period resource cost.

We use a simple and general representation of transit supply. Confronted with a change in transit demand in a given period, and assuming that the demand change displays the same spatial distribution as existing demand, the transit supplier can increase the supply of vehicle-kilometres $X_i, i = 3,4$ and/or increase the occupancy rate per vehicle. The first response leads to increased service frequency, and hence to savings in average waiting time: $z_i = z_i\left(X_i\right)$, where $\frac{\partial z_i}{\partial X_i} < 0$, i=3,4 . The conditions under which a transit demand increase will lead to an increased supply of vehicle-kilometres are analysed below.

The supply conditions are subject to a technological constraint. It specifies a lower threshold to the supply of vehicle-kilometre required to meet a given demand of passenger-kilometre. This constraint captures maximal vehicle carrying capacity as well as transit network characteristics. For reasons of analytical simplicity we assume that the threshold does not depend on network congestion. So we get that $X_i \geq k_i q_i$, i=3,4, where k_i is the marginal capacity requirement in each period.

3.2 Welfare programme and optimal pricing conditions

Government has full control over car taxes, transit fares and transit supply. Tax revenues are returned to the consumer in the form of labour tax reductions. This is modelled by introducing the tax revenues into consumer income, and weighting tax revenues by the marginal cost of public funds minus one (μ). Indirect utility is converted to money units by dividing by the marginal utility of income in the reference equilibrium.[34] Under these assumptions, the following lagrangian is to be maximised, where Y is exogenous generalised consumer income and l_r is the private marginal utility of income in the reference equilibrium.

---

[33] The potential effect of congestion on the unit cost is ignored.

[34] The marginal utility of income in the reference equilibrium is known in a calibrated model.

$$
\begin{aligned}
\Im &= \frac{1}{l_r} V\left(p_0, p_i\ , i=1,\ldots,4, Y+\ R\right) + mR + \sum_{i=3}^{4} g_i\ \left(X_i - k_i q_i\right) \\
&\text{where } R = \sum_{i=1}^{4} t_i q_i - F - \sum_{i=3}^{4} c_i X_i
\end{aligned}
\tag{5.1}
$$

We first maximise with respect to t_j , j=1,…,4 while keeping the supply of transit fixed. Using the properties of the indirect utility function, denoting the private marginal utility of income by l, and defining $\tilde{l} = l / l_r$, the first order conditions for taxes on car transport and transit are in (5.2).

$$
-\tilde{l} q_j + \left(m + \tilde{l}\right)\left(q_j + \sum_{i=1}^{4} t_i \frac{\partial q_i}{\partial p_j}\right) - \tilde{l}\left(\sum_{i=1}^{4}\sum_{k=1}^{2} \frac{\partial a_i}{\partial q_k}\frac{\partial q_k}{\partial p_j}\right) - \sum_{i=3}^{4} g_i \left(k_i \frac{\partial q_i}{\partial p_j}\right) = 0, \text{ j=1,...,4} \tag{5.2}
$$

When cross-price elasticities are negligeable and when transport tax revenues do not receive a premium ($\mu=0$), equation (5.2) simplifies to (5.3) for car taxes. This amounts to a tax equal to marginal external congestion cost, or Pigouvian taxation. Neglecting cross-price elasticities with a positive tax revenue weight produces (5.4), again only for car taxes. Here the tax consists of a Pigouvian component and a Ramsey component. The first is equal to a share of marginal external cost, where the share is decreasing in the marginal cost of public funds weight μ. The second is decreasing in the (absolute value of the) own price elasticity, and increasing in the marginal cost of public funds weight.

$$
t_1 = \sum_{i=1,3} \frac{\partial a_i}{\partial q_1} q_i \text{ and } t_2 = \sum_{i=2,4} \frac{\partial a_i}{\partial q_2} q_i \tag{5.3}
$$

$$
t_1 = \frac{\tilde{l}}{\tilde{l}+m}\left(\sum_{i=1,3} \frac{\partial a_i}{\partial q_1} q_i\right) + \frac{m}{\tilde{l}+m}\frac{p_1}{\left|e_{11}\right|} \text{ and } t_2 = \frac{\tilde{l}}{\tilde{l}+m}\left(\sum_{i=2,4} \frac{\partial a_i}{\partial q_2} q_i\right) + \frac{m}{\tilde{l}+m}\frac{p_2}{\left|e_{22}\right|} \tag{5.4}
$$

The equivalent simplifications for transit read:

$$
t_j = \frac{g_j}{\tilde{l}} k_j,\ j = 3,4 \tag{5.5}
$$

$$
t_j = \frac{g_j}{\tilde{l}+m} k_j + \frac{m}{\tilde{l}+m}\frac{p_j}{\left|e_{jj}\right|},\ j = 3,4 \tag{5.6}
$$

In (5.5) the fare is zero when the capacity constraint is not binding (γ_j=0). It is indeed obvious that when a marginal demand increase is met by constant capacity, and there are no boarding and alighting externalities, the marginal resource cost of an extra passenger is zero. When capacity is not sufficient, the fare is raised in order to restrict demand. The fare then reflects the social cost of the capacity constraint. Introducing a positive weight for tax revenues from the transport sector leads to the addition of a Ramsey term to the optimal price condition, see (5.6). Even with sufficient capacity, positive fares may be justified on revenue raising grounds. Weighting the marginal capacity requirement by the marginal cost of public funds factor, implies that fares are used to decrease government expenditures on capacity.

We now look at the decision on the supply of vehicle-kilometres. First, assuming that a larger supply of vehicle-kilometre has no impact on demand, we obtain (5.7).

$$\begin{aligned} g_3 &\geq \left(\tilde{I}+m\right)c_3+\tilde{I}\left(\frac{\partial a_1}{\partial X_3}q_1+\left(\frac{\partial a_3}{\partial X_3}+\frac{\partial z_3}{\partial X_3}\right)q_3\right) \\ g_4 &\geq \left(\tilde{I}+m\right)c_4+\tilde{I}\left(\frac{\partial a_2}{\partial X_4}q_2+\left(\frac{\partial a_4}{\partial X_4}+\frac{\partial z_4}{\partial X_4}\right)q_4\right) \end{aligned} \tag{5.7}$$

The lagrange multiplier γ_i represents the marginal social benefit of supply expansion. Capacity is expanded up to the point where the marginal benefit is set equal to the marginal social cost of supply expansion. The multiplier is positive when the sum of marginal resource and congestion costs exceeds marginal waiting time cost savings. It would be negative if the benefits of reduced waiting time outweigh resource and congestion costs; In that case, it is desirable to expand vehicle-kilometre supply beyond the technical minimum up to the point where marginal costs equal marginal benefits, such that $g_{3,4}=0$.

$$\begin{aligned} g_3 &= \left(\tilde{I}+m\right)\left(c_3+\sum_{i=1}^{4}t_i\left(\frac{\partial q_i}{\partial t_i}\frac{\partial a_i}{\partial X_3}+\frac{\partial q_i}{\partial t_i}\frac{\partial z_3}{\partial X_3}\right)\right)+\tilde{I}\left(\frac{\partial a_1}{\partial X_3}q_1+\left(\frac{\partial a_3}{\partial X_3}+\frac{\partial z_3}{\partial X_3}\right)q_3\right) \\ g_4 &= \left(\tilde{I}+m\right)\left(c_4+\sum_{i=1}^{4}t_i\left(\frac{\partial q_i}{\partial t_i}\frac{\partial a_i}{\partial X_4}+\frac{\partial q_i}{\partial t_i}\frac{\partial z_4}{\partial X_4}\right)\right)+\tilde{I}\left(\frac{\partial a_2}{\partial X_4}q_2+\left(\frac{\partial a_4}{\partial X_4}+\frac{\partial z_4}{\partial X_4}\right)q_4\right) \end{aligned} \tag{5.8}$$

Next, equation (5.8) takes account of demand effects of increased transit supply. Since supply expansion reduces waiting times and waiting times are part of generalised cost, these demand effects on welfare are analogous to the case of a price change. Note that the effect of increased frequency and/or capacity on demand only needs to be taken into account in as far tax revenues change. Unless waiting time cost savings outweigh marginal resource and congestion costs at the technical minimum supply, transit supply is determined by technological conditions. Under these circumstances, occupancy rates are constant and equal to the technical maximum at all demand levels, as the minimal service level constraint binds. Furthermore, the resource costs and the tax revenues are valued at the "private plus social utility of income" because they are financed through taxes. The welfare cost of supply expansion is thus increasing in μ, implying that service levels and consequently public transport demand levels are decreasing in μ.

As noted in section 2, transit supply models often abstract from congestion and from costs of public funds. In that case, it does not matter whether economies of density are exploited through reduced waiting costs or increased occupancy rates, when marginal deviations from an optimal supply equilibrium are considered. Our model indicates that inclusion of both congestion and a positive cost of funds tends to favour higher occupancy rates as the instrument to exploit economies of density (leading to supply at the minimal level which is technically feasible). Moreover, a positive cost of funds decreases the relative importance of externalities and of economies of density in setting transit prices.

4. Empirical implementation

4.1 The link between the theory and the applications

The numerical model used for the case studies contains a more detailed representation of urban transport systems than the theoretical model of section 3. It is more detailed in that (a) it contains environmental, accident and noise externalities, in addition to congestion, (b) more transport markets are taken into account, and (c) multiple consumer groups are represented, which receive an equal welfare weight. The objective function is analogous to (5.1), implying analogous pricing rules (except that cross-price effects are taken into account). With respect to public transport supply, the standard assumption is that the technical constraint is always binding, so that the occupancy rate is fixed. The implicit assumption is that economies of density do not outweigh the marginal social costs of supply expansion. Nevertheless, supply expansion has an effect on average waiting times or on average occupancy rates (which in turn has a feedback effect on demand levels and/or public transport financing requirements). We will compare optimal pricing reforms in case supply expansions do and do not affect waiting times and/or occupancy rates (the latter case hence deviates from the standard assumption).

Section 4.1 briefly describes the model. A more extensive treatment of the model is in Proost and Van Dender (2001). Results for Brussels and for London are discussed in section 4.2.

4.1 An overview of the TRENEN model

Key features

The TRENEN model is a tool for assessment of different types of transport policies. The basic idea is to study the relevant transport markets simultaneously and to search for optimal prices and regulation on these markets, taking into account different types of external costs (congestion, air pollution, accidents and noise).

The demand side is an aggregated representation of the choices of transport users. Demand for passenger transport is generated by assuming that a representative individual optimally allocates her income between passenger transport and other goods. Many passenger transport services are available: the individual can choose between motorised and non-motorised transport, between peak and off-peak travel, and has the option to use her car or one of the available public transport modes. For Brussels, metro and tram or bus are distinguished. The London case study distinguishes bus, metro and rail transit. In the presentation of results we will focus on bus and metro. For passenger car transport, different types and sizes of vehicles are available. These vehicles differ in terms of user costs and environmental impacts. Finally, the individual has the explicit options of driving solo or car-pooling. The structure contains 20 alternative transport markets in total, all of which vary in terms of resource costs, external costs, taxes and substitutability. Demand for each type of transport service in a given geographical zone is a function of the generalised price of that type of transport service (this is the sum of money price and time cost), of the generalised prices of the other transport services and of other factors (like income and preferences).

In the equilibrium price module, generalised prices are computed for the different types of transport services. The generalised price is the sum of three elements (a) the producer price for different types of vehicle km, (b) the transportation time cost consisting of in-vehicle time, walking and waiting time, and (c) a tax (or subsidy) that has two policy functions: to raise tax revenue or subsidise certain modes of transportation, and to correct for external costs of air pollution, marginal congestion costs, noise and accidents. In the case studies of this paper, we will assume values of the marginal cost of public funds ($1+\mu$) parameter of 1.0 and 1.066 for Brussels, and 1.0 and 1.035 for London. The positive values reflect optimal revenue use, i.e. decreasing labour taxes.[35] Using a value equal to zero is not meant to reflect a particular assumption on revenue use, but turns the model into a pure partial equilibrium tool. This is convenient for comparison with previous studies.

The model is calibrated for a given reference equilibrium (here, Brussels and London in 2005) using observed or forecasted money prices and quantities for all transport modes for a representative day of the year, together with information on the ease of substitution between transport and other goods as well as between the different means of transport.[36] Other important inputs are the structure of resource costs of private and public transport, the external costs and the network congestion function. The network congestion function summarises the available network information on the relation between volume of road transport and average speed (O'Mahony et al., 1997). The model is static: it represents the equilibrium for a representative day in a given year and assumes that the stock of all means of transport (private and public) is perfectly adapted to the demand for transport. The road infrastructure and public transport infrastructure (e.g. the rail network) are kept fixed.

The model can be operated in two modes. In the optimisation mode social welfare is maximised under constraints on the available set of policy instruments. In the simulation mode the effects of particular values of the policy instruments are computed.

<u>Transit supply and cost structure</u>

---

[35] The construction of these estimates is described in Van Dender and Proost (1998). Note that increasing transport taxes and reducing labour taxes mainly increases real wages in as far as transport is used by non-workers. The positive values for $1+\mu$ hence represent a tax shifting effect.

[36] The model uses nested CES utility functions, where elasticities of substitution for each nest determine ease of substitution and price elasticities in a given equilibrium.

It is assumed that all transit operations are and remain to be publicly owned, allowing full integration of urban transit services. This arguably is a necessary condition for the potential exploitation of economies of density (Nash, 1988). The public transit operator is not subject to a budget constraint. Any deficit following from optimal operations is covered from general tax revenues.[37] The cost of the public funds requirement is therefore measured at the economy-wide marginal cost of public funds. De Borger and Wouters (1998) show that the inclusion of an explicit budget constraint leads to optimal price rules which are identical to the ones obtained in our model when a positive marginal cost of public funds is assumed (though the resulting optimal price and supply levels may be different). Using the economy-wide cost of funds is therefore a consistent way to measure the welfare costs of revenue requirements.

The total user time cost of transit is the sum of in-vehicle travel time, walking time and waiting time. In-vehicle time depends on congestion conditions, which are determined by the number of passenger car units over all relevant modes on the network in the relevant period. Busses and trams contribute to, and suffer from, congestion, while metro does not. Walking times to bus stops are kept constant. Waiting times at bus stops depend on the demand level within each period, according to Mohring's square root principle for steady state routes (Mohring, 1972). This principle states that the optimal service frequency is proportional to the square root of demand for the service, if speed is independent of the level of service.[38] It is assumed that the level of service is optimally determined in the reference situation, given the prices in all relevant transport markets.

The theoretical analysis of section 3 does not rule out simultaneous changes of average public transport occupancy rates and of service frequency. Unfortunately, the TRENEN-model can deal with either waiting time reductions or increased occupancy rates, but not with the simultaneous combination of both. In peak hours, the average occupancy rates are equal to the operational maximum, which in turn is taken to be 80% of the technical seating capacity of the vehicle.[39] In off-peak hours, the occupancy rates are equal to 20% of the operational maximum. This implies that the vehicle capacity constraint is not binding in the off-peak. Consequently, when the occupancy rate is endogenous it can increase or decrease during off-peak hours, while it can only decrease in peak hours.

---

[37] It is implicitly assumed that the public transit company operates efficiently, or at least that the degree of X-inefficiency is not affected by the policy measures which we analyse.

[38] The (in)dependency between level of service and service speed does not refer to road congestion (this is absent from Mohring's model). Instead, speed is dependent on the service level in the sense that, when both the number of stops and the number of buses on the route segment can be optimised, an increase in the number of buses allows a decrease of the number of stops, such that the service speed increases (for a given level of demand). Our analysis keeps the number and the location of stops constant, such that walking times are constant.

[39] This technical limit is below the seating plus standing capacity of buses, because this capacity can not be fully used at all areas of the transit network throughout the peak hours.

4.2 Case study results

This section presents key results from case studies concerning the impact of economies of density on optimal urban transport prices and on the welfare effects of pricing reform. Section 4.2.1 discusses the effects on prices and traffic flows. The impact of economies of density on the public transport operators' budget is analysed in section 4.2.2. Table 1 is an overview of the policy scenarios which have been computed.

In all scenarios except scenarios 1C and 2C, the economies of density take the form of endogenous waiting times, while average occupancy rates are fixed. In scenarios 1C and 2C, occupancy rates are endogenous and waiting times are fixed. Excluding economies of density is done by switching off the relation between demand and waiting times and by fixing occupancy rates, so that both waiting times and occupancy rates remain equal to the reference values. Comparison of this case to the case with economies of density, allows to assess the effect of economies of density on the optimal price structure and on welfare.

First-best prices are computed for urban transport in Brussels and London, with a zero or positive marginal cost of public funds parameter. Second-best prices are computed for Brussels, assuming that car taxes remain unchanged with respect to the reference situation. In scenarios 7 and 8, optimal transit fares are computed under this restriction. Scenario 9 simulates the effects of free transit provision, using a positive tax revenue weight.

Table 1 Policy scenarios

	Pricing conditions	City, year	Tax revenue premium	Economies of density
Scenario 1A	First best	Brussels, 2005	0	No
Scenario 1B				Waiting times
Scenario 1C				Occupancy rates
Scenario 2A	First best	Brussels, 2005	6.6%	No
Scenario 2B				Waiting times
Scenario 2C				Occupancy rates
Scenario 3A	First best	London, 2005	0	No
Scenario 3B				Waiting times
Scenario 4A	First best	London, 2005	3.5%	No
Scenario 4B				Waiting times
Scenario 5A	Second best (fixed car taxes)	Brussels, 2005	0	No
Scenario 5B				Waiting times
Scenario 6A	Second best (fixed car taxes)	Brussels, 2005	6.6%	No
Scenario 6B				Waiting times
Scenario 7A	Free transit (fixed car taxes)	Brussels, 2005	6.6%	No
Scenario 7B				Waiting times

4.2.1 The effect of economies of density on optimal transport prices and traffic flows

First best pricing with endogenous waiting times (service frequency optimisation)

The welfare gain from first-best pricing with respect to the reference situation is given in table 2, for scenarios 1 to 4, cases A and B. As in (5.1), welfare is measured by the value of the indirect utility function in money terms (which is equivalent to real income for a CES function), plus the premium for additional tax revenue, plus the non-congestion external costs. Implementing optimal transport prices constitutes a substantial welfare gain. This increases when transit service levels are simultaneously optimised, and when transport tax revenues are used optimally. The extra welfare gain from optimising transit service levels becomes smaller in absolute and in relative terms when transport tax revenues receive a premium. The reason is that the efficiency gains in transport as such stand for a smaller part of the welfare gain, as reductions in distortionary non-transport taxes are taken into account. The absolute decrease indicates that economies of density -in the form of waiting time reductions- are exploited to a lesser degree when public funds are costly. Note that the size of the tax premium in Brussels nearly prohibits the realisation of extra welfare gains through waiting time reductions.

Table 2 Percent welfare gains with respect to reference situation for scenarios 1A to 4A and 1B to 4B

	Brussels		London	
	No tax premium	Tax premium 6.6%	No tax premium	Tax premium 3.5%
No econ. of density	1.80%	2.47%	2.02%	2.55%
Optimised service frequency	1.92%	2.50%	2.21%	2.69%

The driving force behind the realisation of the welfare gains is the change in traffic flow (passenger car units), which leads to increased travel speeds (table 3). With a zero premium to transport tax revenues, the transport demand (passenger-kilometre) decrease is concentrated in peak periods, and it is combined with a modal shift towards transit, such that traffic flow decreases more than transport demand. It is clear that waiting time reductions increase the relative attractiveness of transit, which further stimulates the travel time gains through a stronger modal shift towards transit. The transit share in peak hours in Brussels is 33% in the reference situation, 41% under efficient pricing and 45% under efficient pricing in combination with service level optimisation. Introducing a premium for transport tax revenues has no large impact on modal split nor on peak period demand levels; it causes, however, a major reduction in off-peak transport demand. This is not because of external costs, but because off-peak transport is a suitable tax base for tax revenue extraction.

The same patterns are found in London, but in a less extreme form. The marginal cost of public funds is lower, leading to a smaller impact on off-peak flows. Furthermore, the shape of the congestion function for London reflects less outspoken concentration of congestion in peak hours as compared to Brussels (see table A.1). Consequently the Pigouvian component of the tax is higher in off-peak hours and lower in peak hours.

Table 3 **Traffic flow and modal split under first best pricing Without and with optimised service frequency (scenarios 1A to 4A and 1B to 4B)**

	Brussels				
	Reference	First best pricing No economies of density		First best pricing Optimised service frequency	
		No tax premium	Tax premium	No tax premium	Tax premium
Peak					
Car share	67%	59%	61%	55%	57%
Transit share	33%	41%	39%	45%	43%
Off-peak					
Car share	80%	82%	82%	81%	81%
Transit share	20%	18%	18%	19%	19%
Traffic flow (pcu)					
% change peak	0	-20%	-22%	-21%	-23%
% change off-peak	0	-3%	-12%	-7%	15%
Speed (km/h)					
Peak	23.6	38.2	39.6	39.2	40.5
Offpeak	49.6	49.7	49.7	49.7	49.8
	London				
	Reference	First best pricing No economies of density		First best pricing Optimised service frequency	
		No tax premium	Tax premium	No tax premium	Tax premium
Peak					
Car share	74%	66%	67%	63%	64%
Transit share	26%	34%	33%	37%	36%
Off-peak					
Car share	79%	75%	75%	72%	72%
Transit share	21%	25%	25%	28%	28%
Traffic flow (pcu)					
% change peak	0	-17%	-19%	-19%	-20%
% change off-peak	0	-11%	-14%	-13%	-16%
Speed (km/h)					
Peak	22.6	26.2	26.5	26.5	26.8
Offpeak	30.6	31.0	31.1	31.0	31.1

Table 4a Indices of generalised price components, first best pricing (reference equilibrium = 1)
Without and with economies of density
No tax premium
(scenarios 1A and 3A, and 1B and 3B)

		Brussels - First best pricing					
		No economies of density			Optimised service frequency		
		Waiting time	Money price	Generalised price	Waiting time	Money price	Generalised price
Peak hours							
	Car	-	2.4	1.6	-	2.3	1.6
	Bus	1	3.6	1.2	0.9	2.7	1.0
	Metro	1	1.7	1.1	0.8	0.8	0.9
Off-peak hours							
	Car	-	1.2	1.1	-	1.2	1.1
	Bus	1	8.1	2.1	1.2	6.4	1.9
	Metro	1	2.9	1.2	0.9	1.7	1.0
		London - First best pricing					
		No economies of density			Optimised service frequency		
		Waiting time	Money price	Generalised price	Waiting time	Money price	Generalised price
Peak hours							
	Car	-	1.4	1.2	-	1.4	1.2
	Bus	1	7.6	1.9	1.1	6.5	1.7
	Metro	1	1.1	1.0	0.5	0.7	0.9
Off-peak hours							
	Car	-	1.2	1.1	-	1.2	1.1
	Bus	1	5.8	1.6	1.1	4.2	1.4
	Metro	1	1.2	1.0	0.8	0.01	0.8

Table 4a provides detail on the impact of first best pricing on generalised transport prices, when tax revenues receive no premium. When no economies of density are taken into account, first-best pricing implies substantial money price increases for all modes in all time periods. The peak period price increases are mainly driven by the internalisation of congestion externalities. The off-peak price increases for buses are large in Brussels, which is the consequence of very high subsidies to off-peak bus transport in the reference equilibrium. The London data show considerable subsidies to bus transport in both time periods. These subsidies are eliminated in the first best optimum. The money price increases are partly offset by time cost reductions, because of reduced congestion. Consequently the generalised price increases are fairly moderate, except for those types of bus transport which were strongly subsidised in the reference equilibrium.

Table 4b Indices of generalised price components, first best pricing (reference equilibrium = 1)
Without and with economies of density
Positive tax premium (6.6% for Brussels, 3.5% for London)
(scenarios 2A and 4A, and 2B and 4B)

		Brussels - First best pricing					
		No economies of density			Optimised service frequency		
		Waiting time	Money price	Generalised price	Waiting time	Money price	Generalised price
Peak hours							
	Car	-	3.5	2.2	-	3.3	2.0
	Bus	1	8.0	1.7	0.9	6.3	1.5
	Metro	1	5.8	1.7	0.9	4.2	1.4
Off-peak hours							
	Car	-	1.9	1.7	-	1.9	1.6
	Bus	1	14.7	3.1	1.3	12.2	2.8
	Metro	1	8.0	1.9	0.9	5.7	1.6
		London - First best pricing					
		No economies of density			Optimised service frequency		
		Waiting time	Money price	Generalised price	Waiting time	Money price	Generalised price
Peak hours							
	Car	-	1.8	1.5	-	1.8	1.5
	Bus	1	10.6	2.3	1.1	9.3	2.1
	Metro	1	2.4	1.3	0.5	1.8	1.2
Off-peak hours							
	Car	-	15	1.4	-	1.5	1.4
	Bus	1	8.8	2.0	1.1	6.8	1.8
	Metro	1	3.1	1.3	0.8	1.5	1.0

When service frequency is optimised, the generalised price of public transport increases less strongly, in both cities in both time periods. This reflects the impact of public transport demand increases on average waiting times. Money price increases are lower as well, for public transport, showing that subsidies –in the form of lower tax rises- are required to internalise the positive waiting time externality. However, service frequency optimisation does not necessarily imply that waiting time decreases: bus waiting times increase in the off-peak period in Brussels and in both periods in London. The reason is that the price effect of the abolition of subsidies is so large that demand levels can no longer sustain the reference service frequence. Metro, which is congestion free, becomes cheaper in terms of time and money during peak hours in both cities.[40] Finally, note that the introduction of service frequency optimisation has no (significant) impact on money prices and generalised prices of car transport. This suggests that the interaction between economies of density and road congestion is very limited.

When transport tax revenues are weighted at the marginal cost of public funds (table 4b), transport taxes are not only used to internalise externalities, but also as revenue raising instruments. Therefore, the money price increases in table 4b are higher than in table 4a, for

[40] This conclusion may no longer hold when account is taken of crowding externalities.

all modes and time periods in both cities. This is reflected in higher generalised price increases. The consequence is a larger decrease in transport demand, especially in off-peak hours (see table 3). Taking account of economies of density implies that the effect of the tax revenue premium on money prices is more moderate, but it remains substantial. The effect of the premium on optimal service frequencies is small, or even negligeable in some cases. The reason is that, while the tax revenue premium causes a general increase in transport prices, relative prices are less affected. The attractiveness of public transport is not fundamentally modified by the tax revenue premium. Together with the fact that public transport demand levels are not drastically lower when tax revenues receive a premium, implies that the optimal service frequency is only slightly affected.

Finally, observe that the model does not produce zero transit prices in any of the first best scenarios (except nearly for off-peak metro prices in London). Referring to the discussion of equation (5.7), this means that when public transit supply is at the minimal level which is technically feasible, there is a benefit in terms of waiting time reductions, but this benefit does not outweigh the costs of capacity expansion and of congestion.[41] Therefore, prices are positive in order to restrict demand. This does not mean that service should not be expanded when demand changes.

First best pricing with endogenous off-peak occupancy rates

We compare scenarios 1A to 1C and 2A to 2C, in order to assess the difference between realising economies of density through service frequency optimisation and through endogenous occupancy rates. This comparison is only given for the Brussels case study. The comparative welfare effects are in table 5. Tables A.7 and A.8 contain detail on the effects on prices and traffic flows.

Table 5 Percent welfare gains from first-best pricing with respect to reference situation Scenarios 1A to 2C

	Brussels	
	No tax premium	Tax premium 6.6%
No economies of density	1.80%	2.47%
Optimal service frequency	1.92%	2.50%
Optimal occupancy rates	2.30%	2.96%

Using endogenous occupancy rates and giving a zero weight to transport tax revenues permits a 2.3% welfare gain from first best pricing in Brussels, to be compared to 1.92% when the frequency of service is endogenous. With a positive tax revenue weight, we have 2.96% gain

[41] In fact, the structure of the TRENEN model implies an additional cause of taxing above marginal social cost. The model uses a fixed tax rate on the composite commodity, which represents the average VAT rate in the economy. Adaptation of the optimal pricing rules shows that this leads to an increase of the tax on transport goods. This increase depends on the substitutability between the transport good and the composite commodity. The nested CES structure, with the inherent separability patterns, implies that the deviation is the same for all transport goods. The statement concerning the causes of non-zero fares in transit holds after correction for the tax rate on the composite commodity, however. Of course, this tax rate does affect the optimal fare levels.

with endogenous off-peak occupancy rates, and 2.50% with endogenous service frequency. Increased occupancy rates therefore seem to be preferable from the social point of view, whether public funds are costly or not. However, this result is strongly dependent on the assumption that occupancy rates can be increased up to the technical limit in off-peak hours, without further implications for waiting and walking times.[42] It should therefore be interpreted as an upper bound to welfare gains from increasing occupancy rates.[43]

Two other effects are of more interest. First, endogenous occupancy rates generate an extra welfare gain which is (more or less) independent of the transport tax revenue weight in absolute terms (+0.5% gain under both tax weight assumptions, for Brussels), in contrast to the case of waiting time reductions. The reason is that increased occupancy rates do not generate an externality to the consumer (by assumption, in this model, as there are no crowding externalities). No costly subsidy is required to generate the savings from economies of density. Nevertheless, the optimal prices with endogenous occupancy rates are considerably lower than with waiting time reductions in off-peak hours (the reverse holding during peak hours, as the technical constraint is reached). The relative contribution of increased occupancy rates to the total welfare gain is decreasing in the tax revenue weight, again because transport efficiency is only one objective of pricing reform.

Second, endogenous off-peak occupancy rates lead to a downward revision of peak period car prices in comparison to the case of endogenous waiting times. The reason is that the off-peak transit price decreases so much in relative terms, that transit attracts more peak period car and bus users than is the case without economies of density or with waiting time reductions. Peak period traffic flows decrease by 23%, instead of 21% with waiting time reductions. While limited as such, this extra reduction has a large payoff in terms of travel speeds, and it therefore warrants lower prices during peak hours. The off-peak modal share for transit increases from 20% in the reference situation to 27%.

Second-best: optimal and zero transit fares with constant car taxes

It may be the case that an urban authority can only control transit prices and service levels, while car taxes are set by a higher level authority. In case the urban authority gives an equal welfare weight to all transport users in the urban area, this results in the welfare improvements given in table 6, where transport tax revenues receive a premium of 6.6%.

---

[42] Specifically, average off-peak occupancy rates of buses should increase to 40 (compared to 9 in the reference situation). This increase should not lead to increased walking and waiting times, so that service frequencies and the number of stops are not allowed to decrease.

[43] The larger welfare gain with endogenous occupancy rates is driven to a large extent by the large subsidy for off-peak transit in the Brussels reference situation. The combination of cost savings through increased occupancy rates and optimal prices solves this problem to a large extent. Service level optimisation does not permit the same marginal cost reductions.

Table 6 **Percent welfare gains with respect to reference situation for scenarios 6A to 7B***

	Brussels	
	Optimal transit prices	Zero transit prices
No econ. of density	0.13	0.0
Service level optimisation	0.20	0.02

* A zero weight to tax revenues (scenarios 5A and 5B) leads to lower welfare gains.

The exclusive but optimal use of transit instruments permits a limited welfare gain. Up to 7% of the welfare gain from first best pricing can be reached, through the combined use of fares and service levels increases. The car tax restriction has drastic effects on optimal transit fares: in off-peak hours fares are doubled with respect to the reference situation, while in peak-hours the fare is reduced to zero (or nearly zero in some settings). The latter implies that the transit fare is used as much as possible as an indirect way to reduce peak period congestion. The peak period public transport share increases from 33% to 36%, which is less than under first-best pricing (43%). The off-peak share remains at 20%, in contrast to first-best, where a small decrease is found for the off-peak share of public transport.

Setting zero fares in peak and off-peak hours reduces the welfare gain to zero or nearly zero, in the case where the urban authority can control prices and service levels, and where tax revenues are weighted at the marginal cost of public funds. This policy leads to a 2.1% increase of daily demand for passenger-kilometres, following from a 4% increase during peak hours and a 1% decrease in off-peak hours. We find, remarkably, a shift towards peak period transport despite the price decrease in the off-peak hours. This shift follows from substitution of off-peak car transport (-4.2%) to off-peak and especially peak period public transport (+12.8% and +17.5% respectively). During the peak period, 36% of all passenger-kilometres is by public transport (33% in the reference situation). The off-peak public transport share is 23% (instead of 20% in the reference equilibrium and 16% under first-best pricing).

Introducing a zero fare policy presumably is attractive from an electoral point of view, and it is not necessarily a welfare reducing policy. Furthermore it uses instruments which are usually controlled by urban authorities. It should be pointed out however that other feasible policies, such as the abolishment of free parking, allow much larger welfare gains (cfr Calthrop et al, 2000).[44] Also, the zero-fare policy is not optimal during off-peak hours, for the Brussels case study.

4.2.2 The effect of economies of density on the public transport operators' budget

The use of service characteristics as an instrument in urban transport policy has effects on the budget of the public transport operator. While we have argued that it is in general preferable to value revenue requirements at the economy-wide cost of public funds, the budgetary situation of the operator may nevertheless be a point of policy interest. Table 7 reports the operators' daily receipts, expenditures and revenue requirements for a number of the Brussels scenarios. It is assumed that the operator receives all net taxes, i.e. inclusive of external cost

[44] For readers familiar with current Belgian transport policy: the gains from free transit are entirely due to the indirect effects on congestion levels. The scale and the characteristics of the Brussels case study rule out large substitution from non-motorised transport to transit. Such substitution could drastically reduce the social benefits from free transit provision.

and revenue raising charges. The fixed costs of transit supply are not included in the table. They are equal to 0.373 mio Euro per day for the Brussels case.

Table 7 Public transport budget for Brussels scenarios without economies of density and with endogenous waiting times (mio EURO/day)

Pricing conditions	Tax revenue premium	Economies of density	Receipts	Expenditures	Balance
Reference			0.311	0.321	-0.009
First best	0	No (1A)	1.056	0.312	0.745
		Waiting time (1B)	0.797	0.352	0.445
First best	6.6%	No (2A)	2.326	0.281	2.045
		Waiting time (2B)	2.023	0.315	1.708
Second best (fixed car taxes)	0	No (5A)	0.152	0.322	-0.170
		Waiting time (5B)	0.125	0.329	-0.204
Second best (fixed car taxes)	6.6%	No (6A)	0.176	0.319	-0.143
		Waiting time (6B)	0.135	0.326	-0.191
Free transit (fixed car taxes)	6.6%	No (7A)	0	0.357	-0.357
		Waiting time (7B)	0	0.363	-0.363

Clearly, the public transport sector is more than self-financing when first-best pricing conditions prevail, also after correction for the fixed costs. When economies of density in the form of waiting time reductions are present, the receipts decrease and the expenditures increase, but the balance remains positive because of the charges for congestion and other externalities.[45] For all scenarios it has been assumed that tax revenues are used to decrease existing taxes. This implies that any surpluses should not remain within the public transport sector. The same logic applies to car taxes: first-best pricing of transport is likely to generate more revenues than the present tax system, in urban areas, but there is no general reason to retain these revenues within the transport sector.

When car prices can not be changed and public transport prices are optimised, the public transport sector shows a substantial deficit. The deficit becomes larger when service frequencies are optimised. Setting zero transport prices in all public transport markets makes for a substantially larger deficit than in the previous case. Finally, the fact that the budget is nearly balanced in the reference situation may reflect regulatory conditions.

5. Conclusion

Transit service characteristics were considered as an instrument in the reform of an urban transport system, together with transit fares and car taxes. The theoretical analysis indicated

---

[45] This also is the case when occupancy rates are endogenous and waiting times are fixed.

that supply levels above the technical minimum can be justified up to the point where the social value of waiting time savings equals the marginal resource costs of supply expansion and the marginal congestion costs. In case public funds are not costly, the optimal fare then equals zero. When the waiting time gains are too small, supply levels have to be at the technical minimum, and fares should be used to restrain demand. When the marginal cost of public funds is positive, fares may be positive even when the benefits from waiting time savings are large.

Empirical illustrations for Brussels and for London show that allowing for service level optimisation brings limited additional welfare gains, on top of the gains from internalisation of transport externalities. The results for Brussels indicate that increasing off-peak transit occupancy rates performs better than reducing average waiting times through increased service levels, especially when transport tax revenues are weighted at the marginal cost of public funds. When the urban transport authorities can not change car taxes, it may be optimal to provide peak period public transport for free. The welfare gains from such a policy are small, however.

Several caveats should be mentioned. First, no account was taken of boarding and alighting externalities. They might increase optimal fares above zero, even when the benefits from waiting times are substantial. Second, crowding externalities were disregarded in the analysis of endogenous occupancy rates. These tend to decrease the benefits from increasing occupancy rates. More in general, the supply of transit could be modelled in more detail. Ideally a spatial model should be used, which would allow to take account of bus route characteristics and of endogenous determination of bus stops. It may, e.g., be preferable to use smaller busses on more routes, as this implies less discomfort externalities and shorter travel times than large busses on less routes (which is cheaper in terms of resource costs). Finally, we used efficiency as a social welfare criterion. As public transport is more intensively used by low income groups, it may be justified to price this commodity below marginal social costs for reasons of equity (Mayeres and Proost, 2001).

Appendix Background to the case studies

A.1 The reference situation

Table A.1 shows the cost and price structure for the two reference cases. It shows two dominant types of inefficiencies in reference car prices: (1) peak period prices do not reflect marginal external congestion costs, and (2) free parking for a majority of drivers constitutes a huge subsidy to car use. Note that peak period congestion costs are higher in Brussels than in London, while the reverse holds for off-peak hours. The Brussels network congestion function is very steep at high traffic levels, and relatively flat at low levels. The London function shows less change in slope as traffic levels rise. This difference follows from the difference in geographical scope between both cases. The Brussels application considers the central area within the outer ringroad. This is a small and dense urban region, served by a dense transit network. The London case study is more extended geographically, as it takes account of areas adjacent to the business centre (Greater London). Also, observe that parking costs per kilometre are much higher in London than in Brussels, while other resource cost components of car use are of similar magnitude. With respect to public transport, the case of Brussels shows that prices more or less cover variable resource costs during peak hours, while large subsidies are given for off-peak transit. In London, buses are subsidised in peak and off-peak periods, while metro is not. Table A.2 provides detail on the construction of the reference operational cost structure of transit for the Brussels case. For more detail, see Van Dender and Proost (1998). A similar procedure produced the London reference costs.

Table A.1 Reference prices and social costs, Brussels, London, 2005 (Euro/pkm)

Brussels			Money price	Generalised price	Marginal external congestion cost	Total marginal external cost	Tax
Car*	Peak	Paid parking	0.474	0.855	2.232	2.323	0.093
		Free parking	0.304	0.685	2.232	2.323	0.093
	Off-peak	Paid parking	0.466	0.613	0.003	0.097	0.087
		Free parking	0.296	0.443	0.003	0.097	0.087
Transit	Bus, tram	Peak	0.120	0.924	0.112	0.320	0.016
		Off-peak	0.120	0.773	0.001	0.579	-0.150
	Metro	Peak	0.120	0.877	0	0.002	0.024
		Off-peak	0.120	0.934	0	0.0009	-0.09
London			Money price	Generalised price	Marginal external congestion cost	Total marginal external cost	Tax
Car*	Peak	Paid parking	0.925	1.267	0.445	0.536	0.118
		Free parking	0.325	0.667	0.445	0.536	0.118
	Off-peak	Paid parking	0.913	1.152	0.035	0.130	0.108
		Free parking	0.313	0.552	0.035	0.130	0.108
Transit	Bus, tram	Peak	0.127	0.893	0.068	0.710	-0.02
		Off-peak	0.113	0.854	0.005	0.406	-0.01
	Metro	Peak	0.153	0.822	0	0.0005	0.095
		Off-peak	0.124	0.938	0	0.001	0.093

*We take the example of a small gasoline car with one occupant, who is a city inhabitant. The model distinguishes between several car types, and between inhabitants of and commuters to the city.

Table A.2 Operating costs for transit in the reference situation, Euro, Brussels, 2005

		Metro	Tram	Bus	Tram+Bus
	Yearly costs				
1	Personel	6,626,270	10,610,072	19,179956	29,790,028
2	Energy	3,195,000	4,112,500	4,100,957	8,213,457
3	Maintenance	11,312,500	22,612,500	16,105,000	38,717,500
4	Capital (rolling stock)	7,062,039	11,843,780	2,780,735	14,624,515
	Yearly kilometers				
5	Total	11,106,890	11,137,190	20,132,834	31,270,024
6	Peak (% of total)	33.6%			
7	Off-peak (% of total)	66.4%			
8	Peak cost per vehicle kilometre [8 = (1+2+3)/5 + 4/(6*5)]	3.82599	6.81877	2.66524	4.14457
9	Off-peak cost per vehicle kilometre [9 = (1+2+3)/5 + 4/(7*5)]	1.93117	3.33837	1.94238	2.43958
	Average occupancy rates (pass. per vehicle or per carriage)				
10	Peak	40			
11	Off-peak	9			
12	Peak cost per passenger kilometre [12 = 8/10]	0.09565	0.17047	0.06663	0.10361
13	Off-peak cost per passenge kilometre [13 = 9/11]	0.21457	0.37093	0.21582	0.27105
14	Fixed yearly cost	136,181,837			

A.2 Overview tables counterfactual scenarios

Table A.3 Waiting times and prices under optimal pricing without and with economies of density, Brussels, London, 2005 (reference = 1; waiting time (wait), money prices (MP), and generalised prices (GP))

Brussels		No economies of density			Positive economies of density		
		wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	3.62	1.17	0.86	2.68	1.00
	Metro	1	1.66	1.09	0.82	0.78	0.93
Off-peak	Bus, tram	1	8.13	2.11	1.21	6.43	1.92
	Metro	1	2.88	1.24	0.86	1.62	1.03
Car							
Peak		-	2.38	1.58	-	2.27	1.59
Off-peak		-	1.16	1.13	-	1.16	1.13
London		No economies of density			Positive economies of density		
		wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	7.56	1.88	1.11	6.46	1.75
	Metro	1	1.05	1.01	0.47	0.65	0.89
Off-peak	Bus, tram	1	5.84	1.64	1.06	4.25	1.44
	Metro	1	1.21	1.02	0.77	0	0.79
Car							
Peak		-	1.38	1.24	-	1.36	1.23
Off-peak		-	1.17	1.14	-	1.17	1.14

Table A.4 Welfare and traffic flow characteristics, reference, optimal pricing without and with economies of density, Brussels, London, 2005

Brussels		Reference equilibrium	No economies of density	Positive economies of density
% welfare change		0	1.80	1.92
% change traffic flow (pcu)	Peak	0	-19.9	-21.39
	Off-peak	0	-3.19	-7.13
% change in travel time per km	Peak	0	-44.01	-45.49
	Off-peak	0	-0.14	-0.19
Modal split (%)				
Peak	Car	67.3	58.6	55.1
	Bus and tram	18.3	22.6	24.2
	Metro	14.3	18.7	20.6
	Total	100	100	100
Off-peak	Car	79.6	82.14	80.7
	Bus and tram	13.5	9.2	9.6
	Metro	6.9	8.4	9.7
	Total	100	100	100
London		Reference equilibrium	No economies of density	Positive economies of density
% welfare change		0	2.02	2.21
% change traffic flow (pcu)	Peak	0	-17.2	-18.7
	Off-peak	0	-11.5	-13.0
% change in travel time per km	Peak	0	-13.9	-14.7
	Off-peak	0	-1.25	-1.4
Modal split (%)				
Peak	Car	73.9	66.0	63.3
	Bus and tram	11.1	9.2	9.5
	Metro	15.0	24.7	27.2
	Total	100	100	100
Off-peak	Car	78.8	74.8	71.6
	Bus and tram	12.1	11.2	11.6
	Metro	9.1	14.0	16.8
	Total	100	100	100

Table A.5 Waiting times and prices under optimal pricing and optimal revenue use without and with economies of density, Brussels, London, 2005 (reference = 1; waiting time (wait), money prices (MP), and generalised prices (GP))

Brussels (mcpf 6.6%)		No economies of density			Positive economies of density		
		Wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	7.98	1.72	0.90	6.29	1.48
	Metro	1	5.81	1.66	0.88	4.20	1.41
Off-peak	Bus, tram	1	14.72	3.13	1.26	12.22	2.85
	Metro	1	7.96	1.89	0.91	5.70	1.57
Car							
Peak		-	3.48	2.17	-	3.31	2.08
Off-peak		-	1.91	1.70	-	1.89	1.63
London (mcpf 3.5%)		No economies of density			Positive economies of density		
		Wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	10.63	2.32	1.11	9.29	2.15
	Metro	1	2.42	1.26	0.47	1.83	1.20
Off-peak	Bus, tram	1	8.85	2.04	1.06	6.81	1.79
	Metro	1	3.14	1.29	0.77	1.53	1.00
Car							
Peak		-	1.78	1.53	-	1.76	1.51
Off-peak		-	1.53	1.42	-	1.53	1.42

Table A.6 Welfare and traffic flow characteristics, reference, optimal pricing and optimal revenue use without and with economies of density, Brussels, London, 2005

Brussels		Reference equilibrium	No economies of density	Positive economies of density
% welfare change		0	2.47	2.50
% change traffic flow (pcu)	Peak	0	-22.0	-23.43
	Off-peak	0	-12.19	-15.44
% change in travel time per km	Peak	0	-46.04	-47.29
	Off-peak	0	-0.26	-0.30
Modal split (%)				
Peak	Car	67.3	60.7	57.3
	Bus and tram	18.3	21.8	23.5
	Metro	14.3	17.5	19.2
	Total	100	100	100
Off-peak	Car	79.6	82.5	80.9
	Bus and tram	13.5	9.3	9.7
	Metro	6.9	8.2	9.5
	Total	100	100	100
London		Reference equilibrium	No economies of density	Positive economies of density
% welfare change		0	2.55	2.69
% change traffic flow (pcu)	Peak	0	-19.0	-20.5
	Off-peak	0	-14.3	-15.7
% change in travel time per km	Peak	0	-14.9	-15.7
	Off-peak	0	-1.5	-1.64
Modal split (%)				
Peak	Car	73.9	66.6	63.9
	Bus and tram	11.1	9.2	9.5
	Metro	15.0	24.2	26.6
	Total	100	100	100
Off-peak	Car	78.8	75.0	71.8
	Bus and tram	12.1	11.1	11.6
	Metro	9.1	13.8	16.6
	Total	100	100	100

Table A.7 Waiting times and prices under optimal pricing, without and with optimal revenue use, without and with endogenous occupancy rates, Brussels, 2005 (reference = 1; waiting time (wait), money prices (MP), and generalised prices (GP))

Brussels (mcpf 0%)		No endogenous occupancy rates			Endogenous occupancy rates		
		wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	3.62	1.17	1	3.33	1.12
	Metro	1	1.66	1.09	1	1.42	1.05
Off-peak	Bus, tram	1	8.13	2.11	1	3.08	1.33
	Metro	1	2.88	1.24	1	0.83	0.98
Car							
Peak		-	2.38	1.58	-	1.68	1.39
Off-peak		-	1.16	1.13	-	1.14	1.14
Brussels (mcpf 6.6%)		No endogenous occupancy rates			Endogenous occupancy rates		
		Wait	MP	GP	wait	MP	GP
Transit							
Peak	Bus, tram	1	7.98	1.72	1	7.58	1.66
	Metro	1	5.81	1.66	1	5.42	1.61
Off-peak	Bus, tram	1	14.72	3.13	1	7.42	1.99
	Metro	1	7.96	1.89	1	4.92	1.50
Car							
Peak		-	3.48	2.17	-	2.50	1.96
Off-peak		-	1.91	1.70	-	1.82	1.73

Table A.8 Welfare and traffic flow characteristics, reference, optimal pricing, without and with optimal revenue use, endogenous occupancy rates, Brussels, 2005

Brussels		Reference equilibrium	Endogenous occupancy rates / mcpf 0.0%	Endogenous occupancy rates / mcpf 6.6%
% welfare change		0	2.30	2.96
% change traffic flow (pcu)	Peak	0	-22.66	-25.2
	Off-peak	0	-15.4	-22.7
% change in travel time per km	Peak	0	-46.6	-48.6
	Off-peak	0	-0.35	-0.41
Modal split (%)				
Peak	Car	67.3	54.8	56.6
	Bus and tram	18.3	24.8	24.1
	Metro	14.3	20.4	19.2
	Total	100	100	100
Off-peak	Car	79.6	73.3	73.4
	Bus and tram	13.5	15.6	15.6
	Metro	6.9	11.1	10.9
	Total	100	100	100

Bijlage 5

Congestion pricing with multiple trip purposes

1. Introduction

Recently there is an increased interest in the theoretical and applied analysis of welfare effects of congestion pricing in the presence of pre-existing tax and pricing distortions (Mayeres and Proost, 1997 and 2001; Calthrop et al., 2000b; Parry and Bento, 1999 and 2000; Parry, 2000; Bento and Silva, 2001). These and other studies apply the insights of the optimal taxation literature and of the more recent double dividend debate to the transport sector, taking account of the fact that congestion is a 'feedback externality', meaning that it has effects on the consumption of taxed commodities. Typically, the results are that optimal second-best congestion charges will deviate from marginal external congestion costs. The welfare potential of second-best congestion charges is seen to critically depend on revenue use. In particular, it is preferable to reduce labour taxes rather than to redistribute the revenues in a lump sum way. Given that a large share of trips during congested hours are commuting trips, and that commuting is complementary to labour supply, the interaction between congestion taxes and labour taxes can be expected to be important. In this paper, we investigate this interaction, taking account of the fact that non-commuting trips use the road simultaneously with commuters.

Although the bulk of the congestion pricing literature makes no explicit assumptions on the practical implementation of congestion charges, it is often assumed that the charging system as such creates no congestion. This can be achieved through high-tech systems, e.g. on-road tolling points which deduct the charge from a smart card as a vehicle passes the tolling point. These systems are operational, and probably not prohibitively expensive (e.g. Small and Gomez-Ibanez, 1998). It is clear that such a system can not adapt the charge to the trip purpose. The purpose of this paper is to estimate the welfare cost of not differentiating charges between commuting trips and leisure related trips. Using a stylised dataset, it is suggested that these costs may be substantial. The policy implication is that high-tech charging systems should be complemented by other fiscal measures, e.g. income tax deductions of (part of) the congestion charges for commuting trips.

The problem is studied in a simple general equilibrium model, which extends the analysis by Parry and Bento (1999). We model a transport system in which cars and buses simultaneously use a congestible network in order to produce leisure trips and commuting trips. There is a strict complementarity between the number of commuting trips and the supply of fixed length labour days, in that each labour day requires a morning and an evening peak trip. Government can raise taxes on labour and on transport in order to finance a given transfer. In this context, we analyse the welfare effects of various types of tax reform.[46]

[46] The extensions with respect to Parry and Bento, 1999, are: (a) explicit representation of leisure trips in the theoretical analysis, (b) assuming a shared congestible network for both transport modes. The methodology is different as well.

Section 2 presents the theoretical analysis. The optimal tax structure is derived for the cases where government can or cannot differentiate transport taxes between trip purposes. Section 3 discusses a numerical illustration, suggesting that transport tax differentiation between leisure trips and commuting trips is strongly desirable when the labour tax rate is taken to be fixed. Section 4 concludes.

2. Theoretical analysis

On order to gain some insight in the characteristics of the optimal tax structure, we proceed as follows. Section 2.1 presents the model structure and the first order conditions for a consumer optimum. The optimal tax structure is derived in section 2.2, for the second-best situation in which government has no access to a lump sum transfer. A distinction is made between the situation where tax differentiation between trip purposes is and is not feasible (sections 2.2.1 and 2.2.2 respectively). For the case with differentiation, the components of the optimal taxes are explained, and the optimal leisure and commuting transport taxed are given for the case where there are no cross-effects between leisure and commuting transport. For the case without differentiation, we limit ourselves to a presentation of the system of first order conditions where we neglect cross-price elasticities. This is sufficient to indicate what the basic effects of the uniformity constraint are.

2.1 Model components and consumer optimum

We use a strongly stylised representation of the urban transport sector, describing preferences by a single representative consumer's utility function and assuming that the transport network can be represented as a single congestible link between a single origin and destination. Only morning and evening peak transport are analysed. The representative consumer neglects her impact on the congestion externality. The utility function (4.1) is an extension of the form used by Parry and Bento (1999).

$$\begin{aligned} &u = U\left(q_0, q_1, q_2, N\right) + T\left(q_3, q_4\right) \\ &\text{where } q_0 = \text{composite commodity (untaxed numéraire)} \\ &\qquad q_1 = \text{leisure related car trips} \\ &\qquad q_2 = \text{leisure related bus trips} \\ &\qquad q_3 = \text{commuting by car} \\ &\qquad q_4 = \text{commuting by bus} \\ &\qquad N = \text{leisure} \end{aligned} \tag{4.1}$$

The subutility function T allows that the consumer is not indifferent between transport modes for commuting. The choice of commuting mode has no direct impact on other consumption, however. In particular, modal choices in leisure transport and in commuting are made independently, up to an income effect. A strict complementarity between labour supply L and commuting trips is imposed, such that $L = q_3 + q_4$. We assume that the same transport mode is used for the morning and evening commuting trip.

Resource costs (*c*), taxes (*t*), money prices (*p*) and time costs (*a*, *z*) are as follows.

$$
\begin{aligned}
&p_i = c_c + t_i && \text{money price of a car trip } (i = 1,3) \\
&p_i = t_i && \text{money price of a bus trip } (i = 2,4) \\
&a = a(X) && \text{in vehicle time required for a trip, } X = \text{traffic flow} \\
& && a' \equiv \frac{\partial a}{\partial F} > 0 \\
&z = \overline{z} && \text{average waiting time for a bus}
\end{aligned}
\tag{4.2}
$$

This structure implies a number of assumptions and restrictions:

- Since all modes simultaneously use the same link and travel at the same speed, in vehicle time (*a*) for a trip is equal for all modes and trip purposes.
- The resource cost of a car trip is equal across trip motives. Taxes are allowed to differ across motives. In section 2.2.2, we introduce the restriction that transport taxes can not be diversified according to trip motives. The resource costs of car and bus trips are taken to be constant per trip.
- The money price of bus trips is equal to the tax. This tax may or may not cover the resource costs c_p, which are covered out of tax revenues (centralised supply of bus transport).
- All transport modes contribute to congestion. The contribution may differ between modes, but not between trip motives. In the theoretical analysis, the differential impact of modes on congestion is neglected, for reasons of clarity. The applied model in section 3 takes account of the different congestion effects of cars and buses.
- In general, average waiting times at bus stops are a decreasing function of the supply of bus trips (economies of density). The supply of bus trips may increase when demand for bus trips increases. We abstract from economies of density here. Including them is a straightforward extension, as long as bus supply decisions are not endogenised. A more detailed (partial equilibrium) analysis of the role of economies of density in urban transport pricing is in Van Dender and Proost (2001).

Normalising the gross wage to one, taking the numéraire (q_0) as the untaxed good and normalising its producer price to one, the consumer's money and time budget constraints can be written as in (4.3). In these equations, t_L is the tax on labour supply, S is a lump sum transfer and $\overline{L}$ is the time endowment.

$$
\begin{aligned}
&(1 - t_L) L + S = q_0 + \sum_{i=1}^{4} p_i q_i \qquad [l] \\
&\overline{L} = N + L + \sum_{i=1}^{4} (a_i + z_i) q_i \qquad [g] \\
&\text{where } a_i = a,\ i = 1,\dots,4 \text{ and } z_i = \overline{z},\ i = 2,4 \text{ and } z_i = 0,\ i = 1,3
\end{aligned}
\tag{4.3}
$$

Using the multipliers in square brackets, the first order conditions for a consumer optimum are:

$$
\begin{aligned}
U_0 &\equiv \frac{\partial U}{\partial q_0} = l \\
U_1 &\equiv \frac{\partial U}{\partial q_1} = l\, p_1 + g\, a \\
U_2 &\equiv \frac{\partial U}{\partial q_2} = l\, p_2 + g\left(a + z\right) \\
T_3 &\equiv \frac{\partial T}{\partial q_3} = -l\left(1 - t_L\right) + l\, p_3 + g\left(1 + a\right) \\
T_4 &\equiv \frac{\partial T}{\partial q_4} = -l\left(1 - t_L\right) + l\, p_4 + g\left(1 + a + z\right) \\
U_N &\equiv \frac{\partial U}{\partial N} = g
\end{aligned}
\tag{4.4}
$$

The marginal value of time hence is endogenous, and given by $\frac{U_N}{U_0} = \frac{g}{l}$. It is not equal to the gross or net wage, as can be seen, e.g, for car commuting. : $\frac{g}{l} = \frac{\left((1 - t_L) - p_3\right) + \frac{T_3}{l}}{(1 + a)}$. As is common in the transport literature (e.g. Jara-Diaz, 2000), the value of time equals the net wage corrected for the disutility of commuting travel.

2.2 Optimal transport taxes

The government's problem is to maximise consumer welfare (given by the indirect utility function), subject to a budget constraint. The budget constraint stipulates that the available tax instruments must be used to finance a given transfer S and the costs of bus supply. Hence, all instruments have two functions: raising revenue and internalising externalities. Section 2.2.1 deals with the case in which all transport commodities can be taxed separately. In section 2.2.2, the taxes are constrained in the sense that they are uniform across trip purposes.

In the formulation of the social welfare programme, production efficiency is assumed. From Diamond and Mirrlees (1971), it is known that production efficiency is not necessarily optimal when not all final commodities can be taxed. Issues of production efficiency will be neglected in the present analysis.

2.2.1 Differentiation of transport taxes across trip purposes

Government uses transport taxes and labour taxes in order to maximise the Lagrangian in (4.5). Use of the lump sum transfer is ruled out, so that we are in second-best.

$$\Im = V + m\left(\sum_{i=1}^{4} t_i q_i + t_L L - \sum_{i=2,4} c_p q_i - S\right) \tag{4.5}$$

In this set-up, there are five instruments (four transport taxes and the labour tax), for five taxeable commodities (four transport commodities and labour). However, the strict complementarity between labour and commuting trips ($L = q_3 + q_4$) implies that one instrument is redundant. In other words, there is an indeterminacy in the choice of the five instruments. We first discuss the first order conditions, and then look at optimal leisure transport taxes and commuting taxes when there are no cross-price effects between both.

First-order conditions

After some re-arranging, the system of first-order conditions can be written as in equations (4.6). We write them out in full in order to clarify the similarities and the differences. The expressions are in terms of the uncompensated price elastictities. Note, however, that due to the additive structure of the utility function, there is no substitution effect between labour and leisure trips by either transport mode. Price changes for one trip purpose only affect demand for the other purpose through the income effect. Regarding notation, F denotes the aggregate traffic flow, so that $a'F$ gives the marginal external congestion costs in time units.

The main characteristics of the optimal taxes are discussed in the next four points.

First, note that public transport taxes (4.6.2 and 4.6.4) reflect the marginal resource cost of the bus trip. This of course follows from our assumption that government supplies bus trips, and uses taxes to finance this supply. As car users directly incur the resource costs of their trips, these resource costs do not appear in the expressions for the car taxes.

Second, all transport taxes (4.6.1 to 4.6.4) contain three similar components: the Ramsey component, the trip interaction component and the Pigouvian component. We briefly discuss these components.

The Ramsey component refers to the revenue raising function of the tax. It appears because of the absence of the lump sum instrument. It is decreasing in the own price elasticity of the taxed good, and increasing in the marginal cost of public funds (μ/l). Since we are in second-best, the marginal cost of public funds is larger than one. Therefore, the Ramsey component is positive as long as the uncompensated own price elasticity of the taxed good is negative.

The Pigouvian component (last term in the transport tax equations) stipulates that the transport taxes are used to internalise the congestion externality caused by the taxed good. However, this component is decreasing the marginal cost of public funds. Together with the Ramsey component, this indicates that the revenue raising function of the tax becomes relatively more important as the marginal cost of public funds increases, and less attention is given to the correction of the inefficiencies from the externality.

The trip interaction component consists of three subcomponents, each of which relates to the deviation between taxes and marginal external congestion costs (plus the marginal resource cost of a bus trip in the case of bus transport markets) on the other transport markets. When these deviations are zero, the trip interaction components drop from the tax expression. However, as can be seen by considering the system of equations formed by the first order conditions, marginal social cost pricing in transport is not optimal. It is not a solution to the system of first order conditions. This directly implies that trip interaction components will matter in the optimal tax structure.

Third, note that for commuting transport (4.6.3 and 4.6.4) only the sum of the transport tax and the labour tax is determined by the first order condition. This is the consequence of the strict complementarity between commuting and labour supply, which causes one tax instrument to be redundant. The indeterminacy of the commuting and labour taxes is further discussed below.

Fourth, the expression for the labour tax contains no specific Pigouvian component. The Ramsey and the trip interaction components are similar to those of the transport taxes. The absence of the Pigouvian term suggests that labour supply as such generates no congestion. However, as commuting trips clearly do cause congestion and as the complementarity between commuting and labour supply is strict, it may as well be said that labour supply causes congestion. This again points to the indeterminacy of the tax configuration concerning labour and commuting.

In the next paragraphs, the system of first order conditions is solved under the simplifying assumption that the gross price elasticities between commuting and leisure transport markets are zero. This allows to solve for the optimal leisure transport taxes from equations (6.4.1) and (6.4.2), and for the labour- and commuting-related taxes from equations (6.4.3) to (6.4.5). We first discuss the optimal taxes for leisure trips, and then turn to commuting and labour taxes.

$$\frac{t_1}{p_1} = \underbrace{\frac{1-\mu/l}{\mu/l}\frac{1}{e_{11}}}_{\text{Ramsey}} - \underbrace{\frac{1}{p_1}\left[\frac{e_{21}}{e_{11}}\frac{q_2}{q_1}\left(t_2 - c_p - \frac{g}{m}a'F\right) + \frac{e_{31}}{e_{11}}\frac{q_3}{q_1}\left(t_3 + t_L - \frac{g}{m}a'F\right) + \frac{e_{41}}{e_{11}}\frac{q_4}{q_1}\left(t_4 + t_L - c_p - \frac{g}{m}a'F\right)\right]}_{\text{Trip interaction}} + \underbrace{\frac{g}{m}\frac{a'F}{p_1}}_{\textit{Pigou}} \tag{4.6.1}$$

$$\frac{t_2}{p_2} = c_p + \frac{1-\mu/l}{\mu/l}\frac{1}{e_{22}} - \frac{1}{p_2}\left[\frac{e_{12}}{e_{22}}\frac{q_1}{q_2}\left(t_1 - \frac{g}{m}a'F\right) + \frac{e_{32}}{e_{22}}\frac{q_3}{q_2}\left(t_3 + t_L - \frac{g}{m}a'F\right) + \frac{e_{42}}{e_{22}}\frac{q_4}{q_2}\left(t_4 + t_L - c_p - \frac{g}{m}a'F\right)\right] + \frac{g}{m}\frac{a'F}{p_2} \tag{4.6.2}$$

$$\frac{t_3 + t_L}{p_3} = \frac{1-\mu/l}{\mu/l}\frac{1}{e_{33}} - \frac{1}{p_3}\left[\frac{e_{13}}{e_{33}}\frac{q_1}{q_3}\left(t_1 - \frac{g}{m}a'F\right) + \frac{e_{23}}{e_{33}}\frac{q_2}{q_3}\left(t_2 - c_p - \frac{g}{m}a'F\right) + \frac{e_{43}}{e_{33}}\frac{q_4}{q_3}\left(t_4 + t_L - c_p - \frac{g}{m}a'F\right)\right] + \frac{g}{m}\frac{a'F}{p_3} \tag{4.6.3}$$

$$\frac{t_4 + t_L}{p_4} = c_p + \frac{1-\mu/l}{\mu/l}\frac{1}{e_{44}} - \frac{1}{p_4}\left[\frac{e_{14}}{e_{44}}\frac{q_1}{q_4}\left(t_1 - \frac{g}{m}a'F\right) + \frac{e_{24}}{e_{44}}\frac{q_2}{q_4}\left(t_2 - c_p - \frac{g}{m}a'F\right) + \frac{e_{34}}{e_{44}}\frac{q_3}{q_4}\left(t_3 + t_L - c_p - \frac{g}{m}a'F\right)\right] + \frac{g}{m}\frac{a'F}{p_4} \tag{4.6.4}$$

$$\frac{t_L}{p_L} = \frac{1-\mu/l}{\mu/l}\frac{1}{e_{LL}} - \frac{1}{p_L}\left[\frac{e_{1L}}{e_{LL}}\frac{q_1}{L}\left(t_1 - \frac{g}{m}a'F\right) + \frac{e_{2L}}{e_{LL}}\frac{q_2}{L}\left(t_3 + t_L - \frac{g}{m}a'F\right) + \frac{q_3}{L}\left(t_3 - \frac{g}{m}a'F\right) + \frac{q_4}{L}\left(t_4 - c_p - \frac{g}{m}a'F\right)\right] \tag{4.6.5}$$

Optimal taxes for leisure trips

In case the uncompensated price elasticities between leisure and commuting markets are zero, the optimal leisure car tax, conditional on an optimal leisure bus tax, is implicitly defined by equation (4.7). The expression for the optimal leisure bus tax is analogous, except that it contains the resource cost of the bus trip (as in 4.6.2).

$$\frac{t_1}{p_1} = \left[\frac{e_{22}}{e_{11}e_{22} - e_{21}e_{12}} - \frac{p_2 q_2}{p_1 q_1} \frac{e_{21}}{e_{11}e_{22} - e_{21}e_{12}} \right] \left[\frac{1 - \mu/l}{\mu/l} \right] + \frac{g}{\mu} \frac{a'F}{p_1} \tag{4.7}$$

Simultaneously setting optimal taxes for leisure trips implies that the modal taxes do no longer reflect the deviation between prices and marginal social costs in the substitute mode. The optimal tax expression hence only contains the Ramsey component and the Pigouvian component, while the modal interaction term drops out. The Pigouvian component is unchanged with respect to equation (4.6.1). However, the dependence of the car tax on the cross-price elasticities with respect to the substitute mode (leisure bus transport), remains. This is reflected in the modification, with respect to equation (4.6.1), of weight of the Ramsey term. We now discuss the role of the cross-price effects.

When the cross-price elasticities are zero, the weight of the Ramsey term simplifies to $(1/\varepsilon_{11})$, which is the same as in equation (4.6.1). With non-zero cross price effects, the weight consists of two components. Assuming that the uncompensated own price elasticties are negative, that the uncompensated cross-price elasticities are positive and that the own price effects are larger (in absolute terms) than the cross-price effects[47], it follows that the first component is negative and the second one positive. As the second component is subtracted from the first one, the weight is negative under our assumptions on the elasticities. The overall Ramsey term is therefore positive.

According to the first component of the weight, the Ramsey term is set such as to avoid excessive distortions in the taxed market and in substitute markets. The component is decreasing in the own price elasticity of leisure car trips and in the cross-price elasticities with respect to the leisure bus market. It is increasing in the own price elasticity of leisure bus trips. However, the second component of the weight amends the first one, in the sense that it increases the Ramsey element of the optimal tax on leisure car trips in as far as leisure bus trips are a substitute. This increase is larger, the better a substitute leisure bus trips are for leisure car trips. This effect is conditional on the assumption that leisure bus trips are taxed optimally. If this is the case, the tax base is not reduced strongly in case leisure car taxes primarily cause a substitution towards (optimally taxed) leisure bus trips.

---

47 These assumptions are satisfied in the applied model of section 3.

Optimal taxes for commuting trips and for labour

Now consider the commuting transport taxes and the labour tax, while retaining the assumption of no cross-price effects between leisure transport and commuting transport markets. The first order conditions only define the sum of the labour tax and the commuting tax for each mode. The combination of the conditions for car and bus commuting, determines the car and bus commuting taxes up to a difference with the labour tax. This is indicated by expression (4.8), which is the counterpart of (4.7) for the car commuting tax. The interpretation is the same as for (4.7), except that now only the sum of the car commuting tax and the labour tax is determined. The expression for the optimal 'bus plus labour tax' is fully analogous, except for the inclusion of the bus resource cost in the bus tax. It is omitted for reasons of brevity.

$$\frac{t_3+t_L}{p_3}=\underbrace{\left[\frac{e_{44}}{e_{33}e_{44}-e_{43}e_{34}}-\frac{p_4q_4}{p_3q_3}\frac{e_{43}}{e_{33}e_{44}-e_{43}e_{34}}\right]}_{R_3}\left[\frac{1-\mu/l}{\mu/l}\right]+\frac{g}{\mu}\frac{a'F}{p_3} \tag{4.8}$$

Substitution of the optimal commuting tax expression in condition (4.6.5) for the labour tax (takin account of the simplification with respect to cross-price elasticities), does not yield an implicit equation for the labour tax, because of the indeterminacy in the taxes (which in turn is caused by the fact that there are three taxes for two commodities, due to the complementarity constraint). However, the following relationship between the Ramsey term weights of the commuting taxes can be established. Let R_3 denote the Ramsey term weight of the car commuting tax (as indicated in (4.8)), and R_4 the Ramsey term weight of the bus commuting tax. We then have:

$$e_{LL}=\frac{p_L L}{q_3R_3+q_4R_4}=\frac{p_L\left(q_3+q_4\right)}{q_3R_3+q_4R_4} \tag{4.9}$$

Equation (4.9) indicates that the Ramsey terms weights of the commuting transport taxes are inversely related to the elasticity of labour supply. A higher elasticity of labour supply implies less revenue raising through commuting transport taxes. Hence, commuting is taxed less when the tax discourages labour supply more. Further, note that the weighted sum of the Ramsey weights has the same sign as the labour supply elasticity. As the latter here refers to the tax, it can expected to be negative. This implies that the overall Ramsey component of commuting transport taxes is positive. Note that this does not necessarily lead to commuting taxes above marginal external congestion costs, as the Pigouvian component is corrected downwards for the marginal cost of public funds.

Summary

Concerning the optimal tax expressions for the case where government can differentiate taxes across trip purposes, we find that:

- Transport taxes contain a Ramsey component, a Pigouvian component and a trip interaction component. The interactions between transport markets influence each optimal transport tax, as marginal social cost pricing in transport is not optimal. The

Ramsey components of the taxes take account of these interactions with substitute markets.

- The labour tax condition contains no Pigouvian term, as labour supply as such creates no congestion. However, commuting does create congestion, and commuting is strictly complementary to labour supply.
- The commuting transport taxes can only be determined up to the difference with the labour tax. Multiple solutions are possible for the ‘commuting plus labour tax’ configuration. The Ramsey components of the commuting taxes are intimately related to the elasticity of labour supply, in the sense that revenue raising from commuting decreases as labour supply becomes more elastic.

It will be interesting to compare these results to the case of uniformity of transport taxes across trip purposes, which is analysed in the next section.

2.2.2 Uniform transport taxes across trip purposes

In contrast to the previous section, transport taxes are now required to be uniform across trip purpose, so that $t_1 = t_3 = t_{car}$ and $t_2 = t_4 = t_{bus}$. The government problem becomes:

$$\Im = V + m\left(\sum_{i=1,3} t_{car} q_i + \sum_{i=2,4} t_{bus} q_i + t_L \left(q_3 + q_4 \right) - \sum_{i=2,4} c_p q_i - S \right) \qquad (4.10)$$

Since there are now three instruments for five commodities, the model describes an additional source of second-best, in terms of transport tax instruments. All taxes are uniquely defined. The optimal transport taxes now depend on the price sensitivity and the traffic flow shares of both leisure trips and commuting trips. The general tax expressions for this model become rather intractable, so that we limit the discussion to the system of equations in (4.11), where cross-price effects between transport modes are neglected. This is sufficient to point out the main differences with the case of section 2.2.1.

$$\begin{aligned}
t_L &= \left(\frac{1 - m/l}{m/l} \right) \frac{p_L}{e_{LL}} + \frac{q_3}{L} \left(t_{car} - \frac{g}{m} a' F \right) + \frac{q_4}{L} \left(t_{bus} - c_p - \frac{g}{m} a' F \right) \\
t_{car} &= \left(\frac{1 - m/l}{m/l} \right) \frac{q_1 + q_3}{\frac{\partial q_1}{\partial t_c} + \frac{\partial q_3}{\partial t_c}} - t_L \frac{\frac{\partial q_3}{\partial t_L}}{\frac{\partial q_1}{\partial t_c} + \frac{\partial q_3}{\partial t_c}} + \frac{g}{l} \sum_{i=1}^{4} q_i a' \qquad (4.11) \\
t_{bus} &= \left(\frac{1 - m/l}{m/l} \right) \frac{q_2 + q_4}{\frac{\partial q_2}{\partial t_c} + \frac{\partial q_4}{\partial t_c}} - t_L \frac{\frac{\partial q_4}{\partial t_L}}{\frac{\partial q_2}{\partial t_c} + \frac{\partial q_4}{\partial t_c}} + \frac{g}{l} \sum_{i=1}^{4} q_i a' + c_p
\end{aligned}$$

The labour tax is seen to depend on the commuting transport taxes. The trip interaction terms are the same as in the case with differentiated transport taxes. However, as will be seen next, the transport taxes will be set differently because of the uniformity constraint.

The transport tax expressions contain a Ramsey term, which depends on the price sensitivities of the affected trip purposes. This is a first effect of the uniformity constraint. Whereas in the unconstrained case, the Ramsey term only depends on the price elasticity per mode and per purpose, the present Ramsey term refers to both affected trip purposes.

The second effect of the uniformity constraint shows up in the co-dependence of the transport taxes on the labour tax (cfr. the second term in the transport tax expressions). In particular, transport taxes are revised downward in as far as a marginal labour tax change has effects on commuting (hence on labour supply), relative to the (negative) effect of a modal price on total modal demand. Ceteris paribus, the higher the sensitivity of commuting to the labour tax, the larger the downward revision of the commuting tax. To the reverse, a higher sensitivity of transport (for all purposes) to the transport tax, implies a lower downward revision of the transport tax.

Intuitively, this second effect indicates how, under the restriction of uniform taxes across trip purposes, transport taxes are set to strike a balance between two policy objectives:

- Allowing a reduction of the effective labour tax, through lower commuting taxes, and
- Promoting efficient reductions of the total modal demand. The latter policy objective is not directly related to the marginal external congestion cost, as this is tackled by the third component of the transport tax expressions. The actual policy goal is to reduce leisure transport, so as to allow faster commuting travel (by either commuting mode). Due to the uniformity constraint on the transport tax, this policy goal can only be served imperfectly.

To repeat, the basic effects of the uniformity constraint in transport taxes are that (a) the Ramsey term refers to the price sensitivities of all affected trip purposes, and (b) the correction of the transport taxes for the labour tax is counteracted by the desire to have a high transport tax in order to decrease leisure trips. As will be illustrated in the empirical illustration in section 3, the effect of the uniformity constraint plays out differently under different assumptions on the flexibility of the labour tax.

Before turning to the numerical implementation of the model, we briefly compare our results to those of Parry and Bento (1999). Their model contains no leisure transport, and public transport is assumed to take place on a separate non-congested network. Non-time resource costs of transport are neglected. There are only two policy instruments: the labour tax and the car commuting tax. Using an equal yield tax reform method, they find that the optimal car commuting tax is equal to the marginal external congestion cost (a pure Pigouvian tax).

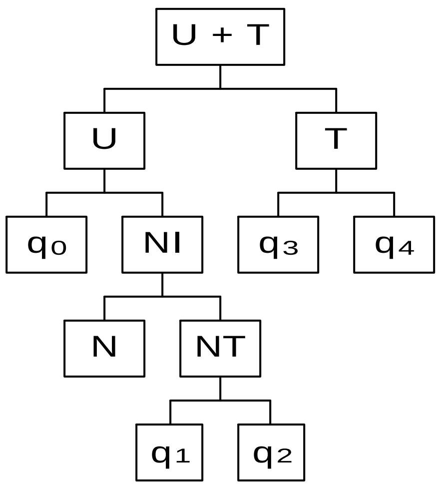
The intuition for this result is that the Pigouvian tax guarantees an optimal modal split, since the relative modal prices will then reflect the marginal social costs. However, if the public transport tax were allowed to differ from zero, the taxes would be indeterminate as in our model of section 2.2.1. Note that our results indicate that setting the public transport tax equal to zero is not necessarily optimal, even when public transport generates no congestion, as the Ramsey component may cause the optimal tax to be non-zero. The absence of the public transport tax in Parry and Bento (1999) should therefore be interpreted as a constraint on the available set of instruments. Consequently, the optimality of a pure Pigouvian tax on car commuting is a second-best result, induced by the requirement of a zero public transport tax. This policy constraint is not necessarily realistic. Finally, the absence of leisure transport modes in Parry and Bento (1999) naturally implies that trip purpose interactions do not matter.

3. Numerical illustration

3.1 Structure and calibration

The numerical model is fully analogous to the theoretical model, except that the preference structure imposes separability restrictions. The model is programmed in GAMS. Data used in this paper are meant to reflect realistic orders of magnitude for Belgian urbanised environments, but do not refer to any particular case. The representative consumer's preferences are modelled with nested CES functions, according to the structure depicted in figure 1.

Figure 1 CES implementation of the utility function



where:

U	utility from non-work related activities
T	utility from commuting and working time
q_0	composite consumption commodity
q_1	leisure transport by car
q_2	leisure transport by bus
q_3	commuting transport by car
q_4	commuting transport by bus
N	leisure time
NT	CES index of leisure transport
NI	CES index of leisure activities

The model is calibrated to a dataset representing current Belgian transport prices in urban contexts (Van Dender and Proost, 1998) using a congestion function and quantities which are derived from a network model for the city of Namur, Belgium (Cornélis and Van Dender, 2001). Calibrating this model requires the simultaneous solution of the system of first order conditions. This is in contrast to models with fixed marginal values of time, where a (separable and parameterised) bottom-up calibration procedure is applicable. The simpler parameterised calibration procedure is not suitable here, as the multipliers of the money and time budget constraints for the consumer are not observed. These multipliers determine, a.o., the marginal value of time.

The congestion function is linear. The free flow speed is 60 km/h. At the reference traffic flow of 2450 passenger car units (PCU) speed is 30 km/h. Since a one-way trip distance of 20km is assumed, daily travel time for a round trip by car is ca. 1.3h. To this we add a fixed average waiting time of 7.5 minutes (1/2 of a 15 minute headway) for a one-way bus trip. The value of time, as implied by the calibration, is 7.65 Euro/hour. This is 47% of the gross wage and 78% of the net wage. These values are closely in line with available estimates (Small, 1992).

The total number of potential network users is 5000 individuals, of whom 1800 supply 8 hours of labour per day.[48] The length of the working day is fixed. The representative individual hence supplies 0.36 days, or 2.88 hours of labour per day. The gross wage is 16.25 Euro/hour. With a labour tax of 40%, the net daily labour income per representative individual is 28.1 Euro.

The modal split for commuting transport is 2/3 car trips and 1/3 bus trips. For leisure trips the car is used for ¾ of all trips. Commuting trips stand for 53% of the total number of trips. These proportions are roughly consistent with the Belgian Mobility Survey (Pollet, 2000) and with more detailed data for Brussels (IRIS, 1993), if it is assumed that the bus mode is easily accessible in our example. The reference traffic flow of 2450 PCU is obtained by assuming that the average occupancy rate and PCU equivalent of a car is 1, while the average occupancy rate of a bus is 40 and the PCU equivalent of a bus is 2. The reference money prices and taxes are in column 2 of table 2. Note that the reference taxes for bus trips are sufficient to cover marginal external congestion costs, while this is not the case for car trips.

In a CES function, the combination of elasticities of substitution with prices and quantities determines price elasticities. The values of the elasticities of substitution used for the central case, and the resulting compensated price elasticties for the reference equilibrium, are in table 1. The sensitivity of the results to these parameters is discussed in section 3.3.

The compensated elasticity of labour supply is 0.19. This is in line with the central estimates (0.15) in Hansson and Stuart (1985) and Ballard et al. (1985). It is below the values of 0.35 and 1.09, used by Parry and Bento (1999) and Mayeres (1999) respectively. The elasticity of labour supply is a key parameter in the model. It is mainly determined by the elasticity of substitution between q_0 and NI. We refer to section 3.3 for some insight into the dependence of model results on this parameter.

[48] The population size as such is of little importance. It is scaled in order to accord with available road capacity. The labour force participation rate is taken to be 36%.

The own price elasticity of demand for commuting trips is determined by the labour supply elasticity. It is lower for commuting trips than for leisure trips. In addition, we choose parameters in the central scenario such that the cross-price elasticities between transport modes is larger for commuting trips than for leisure trips. The underlying assumption is that the availability of public transport modes is larger for commuting trips than for leisure trips, because of differences in the spatial distribution of origins and destinations. The model results are not very sensitive to these parameters however (section 3.3).

Table 1 Elasticities of substitution and main compensated price elasticities, central scenario

Elasticities of substitution		
Composite commodity – leisure index	q_0 – NI	0.7
Leisure transport – other leisure	NT – N	1.1
Leisure car trips – leisure bus trips	$q_1 - q_2$	0.9
Commuting car trips – commuting bus trips	$q_3 - q_4$	0.6
Compensated price elasticities		
	E_{11}	-1.02
	E_{12}	0.03
	E_{21}	0.12
	E_{22}	-0.93
	E_{NN}	-0.19
	E_{34}	0.08
	E_{43}	0.52

3.2 Policy experiments and results

3.2.1 Optimal tax structure for equal tax revenues (central scenario)

Table 2 provides an overview of the main results of two policy experiments. First, we look at unconstrained optimisation of all tax instruments for an equal government revenue requirement. This is in line with the analysis of section 2.2.1. Second, tax instruments are optimised for an equal revenue requirement, imposing equal transport taxes across trip purposes, in accordance with section 2.2.2.

When differentiation of transport taxes between trip purposes is possible (column 3 of table 2), taxes rise above marginal external congestion costs for leisure trips. They are below marginal external congestion costs for commuting trips. As was noted in section 2.2.1, the tax configuration is not unique. Alternative proportions between taxes on commuting transport and taxes labour can achieve the same maximal welfare level for the given revenue constraint. The tax levels as such are therefore of less importance than the quantity effects that they induce. Nevertheless, the tax changes clearly achieve a shift of the tax burden from labour and its complements, to leisure activities.

The maximal welfare level, obtained by differentiated transport taxes, is 0.4% higher than the reference level. The gain follows from a decline in the number of trips and a modal shift towards buses. Because of the modal shift, the decline in the traffic flow is larger than the decline in the number of trips. This leads to faster travel. Note that the decrease in the number of trips is the net result of a strong decline in leisure transport demand and an increase in the number of commuting trips. The increase in the number of commuting trips is exactly matched by the increase of labour supply. Finally, the modal shift from cars to buses is much larger for commuting trips than for leisure trips.

When comparing the constrained tax reform exercise (uniformity of taxes across trip purposes, column 4 of table 2) to the unconstrained one, the most striking result is that only a very small part of the welfare gain is lost. However, a strongly different tax structure is required to achieve the gain. This structure is the unique optimal tax solution (see section 2.2.2). Note that, as the unconstrained optimisation does not produce a unique solution, comparing levels of tax instruments across both cases is of limited relevance.

With uniform transport taxes, all transport is taxed above marginal external costs, and the optimal labour tax is much lower. The uniformity constraint hence mainly generates a shift of the tax burden to transport, and this is compensated by a strong decrease of the labour tax rate. This compensation is an indirect way of arriving at the desired differentiation of taxes across trip purposes, or –essentially- a more efficient distribution of the tax burden between labour and leisure activities.

Table 2 Optimal tax structure for equal revenue requirement, with and without tax differentiation between trip purposes

1	2	3	4
	Reference	Optimal equal yield tax structure	
		Differentiation	Uniformity
% efficiency change	0	0.421%	0.420%
% labour supply change	0	6.75%	6.75%
Tax commuting car trips (Euro/round trip)	4.24	5.54	16.06
Tax leisure car trips (Euro/round trip)	4.24	16.57	16.06
Tax commuting bus trips (Euro/round trip)	0.53	0.04	10.49
Tax leisure bus trips (Euro/round trip)	0.53	9.06	10.49
Labour tax rate	0.40	0.32	0.24
Ratio tax / marginal external congestion cost:			
Commuting car trips	0.62	0.784	2.27
Leisure car trips	0.62	2.344	2.27
Commuting bus trips*	1.53	0.12	29.70
Leisure bus trips*	1.53	25.63	29.70
Modal share cars:			
Commuting trips	67%	63%	63%
Leisure trips	75%	74%	75%
Share commuting trips in total trips	53%	63%	63%
Trip volume	0.68	0.61	0.61
PCU volume	2.45	2.12	2.11

* The marginal external cost of a bus trip is obtained by proportionally distributing the marginal external cost of the vehicle over all occupants.

The fact that the welfare gain is nearly equal to the unconstrained case draws from the result that the main policy objective is to decrease the number of leisure trips in the total number of trips. This can equally well be achieved without differentiation, through a general increase of transport taxes, combined with an appropriate downward revision of the labour tax. The small relative welfare loss from introducing the pricing constraint derives from the problem that the modal split in leisure and commuting trips is not simultaneously corrected. However, the modal split in leisure transport is not very sensitive to price changes. As the unconstrained case suggests, there is little gain to be expected from correcting it. Consequently, the relative modal prices mainly are used to correct for the modal split in commuting, and applying the same relative modal prices to leisure transport entails a small welfare loss only.

Finally, note that the welfare gain of optimised policies for equal revenue requirements is relatively small compared to models where transport prices are optimised for endogenous tax revenues, or where revenues are not taken into account, or where the complementarity between labour supply and commuting transport is not taken into account. Such models typically produce welfare gains between 0.5 and 1.5% for urban environments (Proost et al, 2001; Small, 1992). While the predicted gains from these alternative models should be considered as upper limits to the potential gains from innovations in transport pricing, the present model probably produces a lower limit, as in reality the complementarity between labour supply and commuting is not as strict as assumed here. It should also be stressed that the size of the gain strongly depends on the labour supply elasticity (see section 3.2.3) Also, the gains from optimal transport and labour tax reforms are large in comparison to the welfare effect of marginal decreases in the government revenue requirement: a 1% decrease in the requirement generates 0.05% of welfare gain.

3.2.2 Transport tax reform for equal transfers (central scenario)

This section discusses six policy scenarios. The first four work under the constraint of uniform transport taxes across trip purposes. The last two release this constraint.

Transport tax reform with uniform taxes across purposes

In the previous section, the tax structures were optimised, for an equal government revenue requirement, assuming flexibility of the labour tax and of transport taxes for both modes. In the following four exercises, we analyse the effect of changing one tax, allowing optimal adaptation of a second instrument, while the third one is fixed. Indeed, there are only three instruments, as uniformity of taxes is imposed in the next four scenarios. As before, the government revenue requirement is fixed. In particular, the following second-best scenarios are computed:

- Exogenous increases of car taxes with constant public transport fares and endogenous labour tax rate (**d tc, d tl**),
- Exogenous increases of car taxes with endogenous public transport fares and constant labour tax rate (**d tc, d tp**),
- Exogenous de- and increases of public transport fares with constant car taxes and endogenous labour tax rate (**d tp, d tl**)
- Exogenous de- and increases of public transport fares with endogenous car taxes and constant labour tax rate (**d tp, d tc**).

Figure 1 shows the welfare effects of these policy experiments. The horizontal axis shows by which factor the exogenous taxes (car or bus taxes) are multiplied. At factor 1, the model replicates the reference equilibrium. It is clear from the figure that the combination of increased car taxes and reduced labour taxes (d tc, d tl) outperforms the other policies, and that it is the only one of the analysed second-best policies that produces a significant efficiency gain (0.1%, or some 25% of the gain from the full optimisation exercises described in section 3.2.1). This is so because this policy offers the best combination in terms of reducing congestion (from both trip purposes) and producing an efficient modal split in terms of commuting. Indefinite increases in car taxes are not desirable however: the optimal increase in car taxes is ca. 50% with respect to the reference level, at which point the ratio between car taxes and marginal external congestion costs is 0.92.

Combining increased car taxes with reduced public transport taxes (d tc, d tp) yields welfare gains, but this policy becomes infeasible for large increases in car taxes. The reason is that public transport taxes can not be decreased enough while still satisfying the revenue constraint. The combination of increased public transport taxes with reduced labour taxes (d tp, d tl) produces limited welfare gains. The size of tax the increase for this policy to attain its maximal gain is much larger than for cars, because of the low reference value of bus taxes. Increasing bus taxes and lowering car taxes (d tp, d tc) is welfare decreasing. Lowering bus taxes and increasing car taxes, on the other hand, produces significant welfare gains. The model structure indeed implies that zero public transport prices are optimal when car taxes can not decrease (at least when net bus subsidies are not allowed).

Transport tax reform with uniform taxes across purposes

The uniformity constraint is released for the two second-best scenarios with constant labour taxes and an optimal adjustment of the bus tax. The results are summarised in figure 2, which compares the results of the additional scenarios (**dtp, dtc diff** and **dtc, dtp diff**) to those of the previous scenarios (same as figure 1). As the effect of releasing the uniformity constraint is to a large extent independent of the distance from the reference point, we only consider relatively small deviations from the reference tax levels.

The most important result here is that differentiating taxes between commuting car trips and leisure car trips for constant labour taxes (d tp, d tc diff), generates a slightly larger welfare gain than increasing car taxes and simultaneously decreasing labour taxes (d tc, d tl). The gain is now 27.5% of the maximal gain, instead of 25%. The optimal second-best policy under fixed labour taxes is to combine the policy of car tax differentiation between trip purposes, with zero bus taxes. This generates an efficiency improvement of 0.13%, which is roughly 1/3 of the maximal gain. The analysis thus suggests that, if high-tech tolling systems are used to collect congestion taxes, these systems should be complemented by tax policies which produce a tax differentiation across trip purposes. Partial or full income tax deductability of taxes on commuting trips is an example of such a complementary policy.

The driving force behind the result is that tax differentiation across trip motives allows a separate treatment of commuting transport and leisure transport. The benefits of such separate treatment are larger than the benefits of uniformly increasing car taxes for all trip purposes while decreasing labour taxes. The reason is that the separate treatment constitutes a labour tax reduction in itself. This reduction is not plagued by the inefficiency which characterises the uniform car tax, which only partially corrects the car tax for the interaction with the labour tax (see equation (4.11) in section 2.2.2).

Comparing the impact of car tax differentiation in the present case (with constant labour taxes) to the case described in section 3.2.1 (with endogenous labour taxes), it is remarkable that tax differentiation between trip purposes becomes the main source of the welfare gains when labour taxes are constant, whereas in the case of endogenous labour taxes the uniformity constraint leads to only a very small reduction of the efficiency. In other words, the results suggest that a transport tax reform is to a large extent a reform of labour taxes, combined with a reduction of peak period leisure trips. If labour taxes are fixed, the most effective policy is to differentiate transport taxes across trip purposes. However, this policy only generates a limited efficiency improvement in comparison to the case of endogenous labour taxes. If all taxes can be optimised, the labour tax is reduced, and the uniformity constraint in transport taxation is of secondary importance. The efficiency improvement from such a policy is significantly larger.

Figure 1 **Welfare effects of equal yield policy reforms**

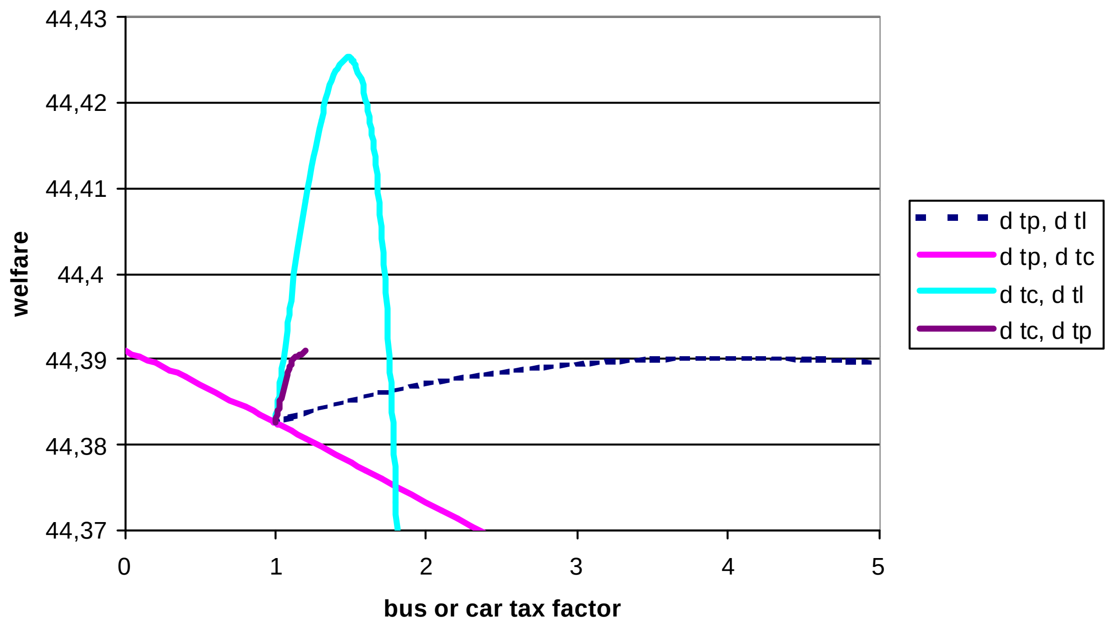
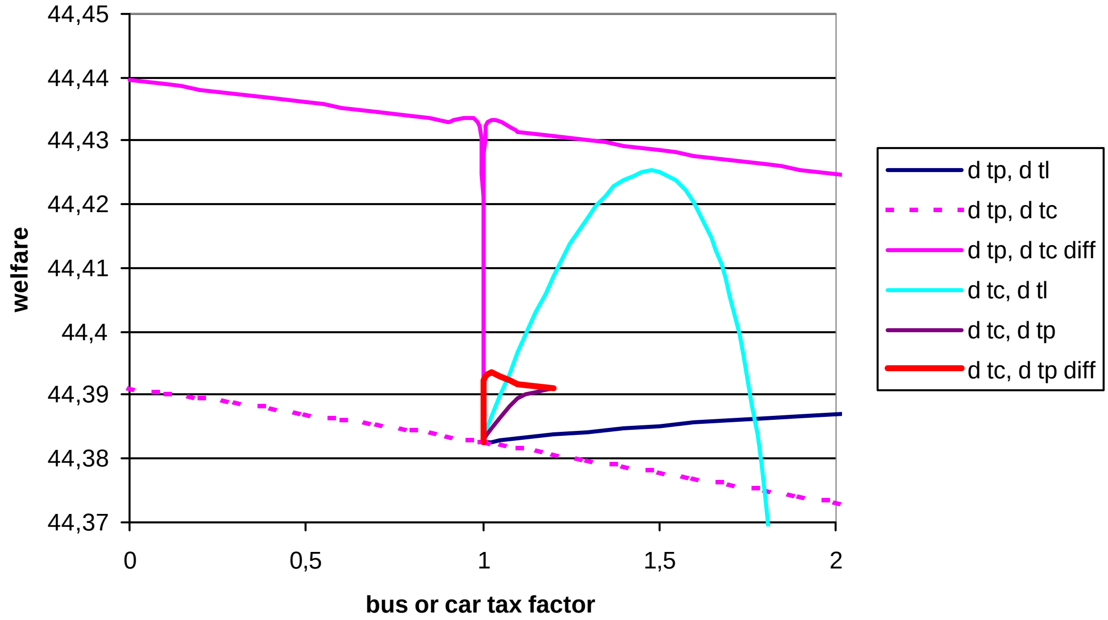


Figure 2 **Welfare effect of tax differentiation across trip purposes**



3.2.3 Sensitivity analysis

Elasticity of substitution between leisure and other commodities

As the preceding analysis suggests that the welfare effects of transport tax reform depend to a substantial degree on the interaction with labour supply, it is no surprise that the model results are mainly dependent on the elasticity of substitution between leisure and other goods. We summarise the basic impact of increasing and decreasing the elasticity of substitution (the compensated labour supply elasticity) from 0.7 (0.19) to 1.2 (0.28) and 0.3 (0.11) respectively.

The welfare potential of optimal tax reforms for constant tax revenue requirements increases with the labour supply elasticity, from 0.06% for the lowest and 1.13% for the highest elasticity. Traffic levels reduce less strongly as the labour supply elasticity rises, mainly because of a relatively higher number of commuting trips by car. Hence, the optimal congestion levels are an increasing function of the labour supply elasticity. Also, it is noteworthy that the level of leisure trips falls less drastically as the labour supply elasticity rises. This means that there is less need to 'tax leisure trips off the road' for higher elasticities of labour supply.

The same basic patterns are found when different labour supply elasticity values are used to compute optimal differentiation of car taxes, for constant bus and labour taxes. For most values of the elasticity, the optimal car commuting tax is actually zero. However, when the labour supply elasticity is sufficiently small, the model suggests a qualitative change in the nature of the second-best policy. Instead of focussing on tax differentials between trip purposes in order to increase labour supply, it becomes preferable to set taxes such that the modal split for commuting trips is strongly changed, in favour of bus trips. This requires a positive car commuting tax. In other words, as the labour supply elasticity diminishes, the policy objective of reducing congestion for current labour supply levels becomes more important than the objective of increasing labour supply (with possibly increased congestion levels). The reduction of congestion is achieved by a transport tax structure that promotes a modal shift towards buses.

Cross price elasticities between transport modes

Decreases in the elasticities of substitution between transport modes, in commuting and/or in leisure transport, have no major impacts on the welfare effects of the various policy experiments, nor on the values of the taxes. The most important changes are that (a) as the degree of substitutability between commuting by car and by bus decreases, leisure transport is decreased relatively more, so as to allow increases in both types of commuting, but at a higher aggregate congestion level; (b) as the degree of substitutability between leisure transport modes decreases, labour supply increases somewhat less compared to the central scenario while the optimal (equal yield) congestion level is slightly higher.

4. Conclusion

On the basis of the numerical implementation of a simple general equilibrium model for multi-modal and multi-purpose passenger transport, we find that the welfare effects of transport tax reform depend to a large extent on the impact on labour supply. In order to have a substantial impact on labour supply, transport tax reform should be accompanied by labour tax changes. When this is not possible through a direct change of the labour tax, the reform should treat commuting transport and leisure transport differently. In general, the results suggest that the relative welfare cost of uniformly taxing different trip purposes is substantial.

Optimal transport tax reforms lead to a combined increase of labour supply and a decrease in the total trip volume and the associated costs of congestion. The volume of commuting trips increases, but this is more than compensated by a decrease in peak period leisure trips. It can be expected that in a multiperiod model, leisure trips will be seen to shift to offpeak hours, so that the total demand decrease becomes smaller.

Note that exogenous parameter values have large effects on the results, in quantitative and in qualitative terms. The results described here should be considered as exploratory. The sensitivity analysis suggests that the results of carefully implemented case studies will not be transferable across cases, both because of the impact of the reference composition of traffic flows and because of the elasticity values. For instance, in our illustration, the reference share of leisure trips is substantial, and it is assumed that a substantial shift of commuting trips towards buses is possible at constant marginal cost. Certainly the latter assumption may not be valid for all urban areas. The gains from transport tax reforms will decrease when peak period traffic consists mainly of commuting trips, and when expansion of public transport supply is not possible at a constant marginal cost.