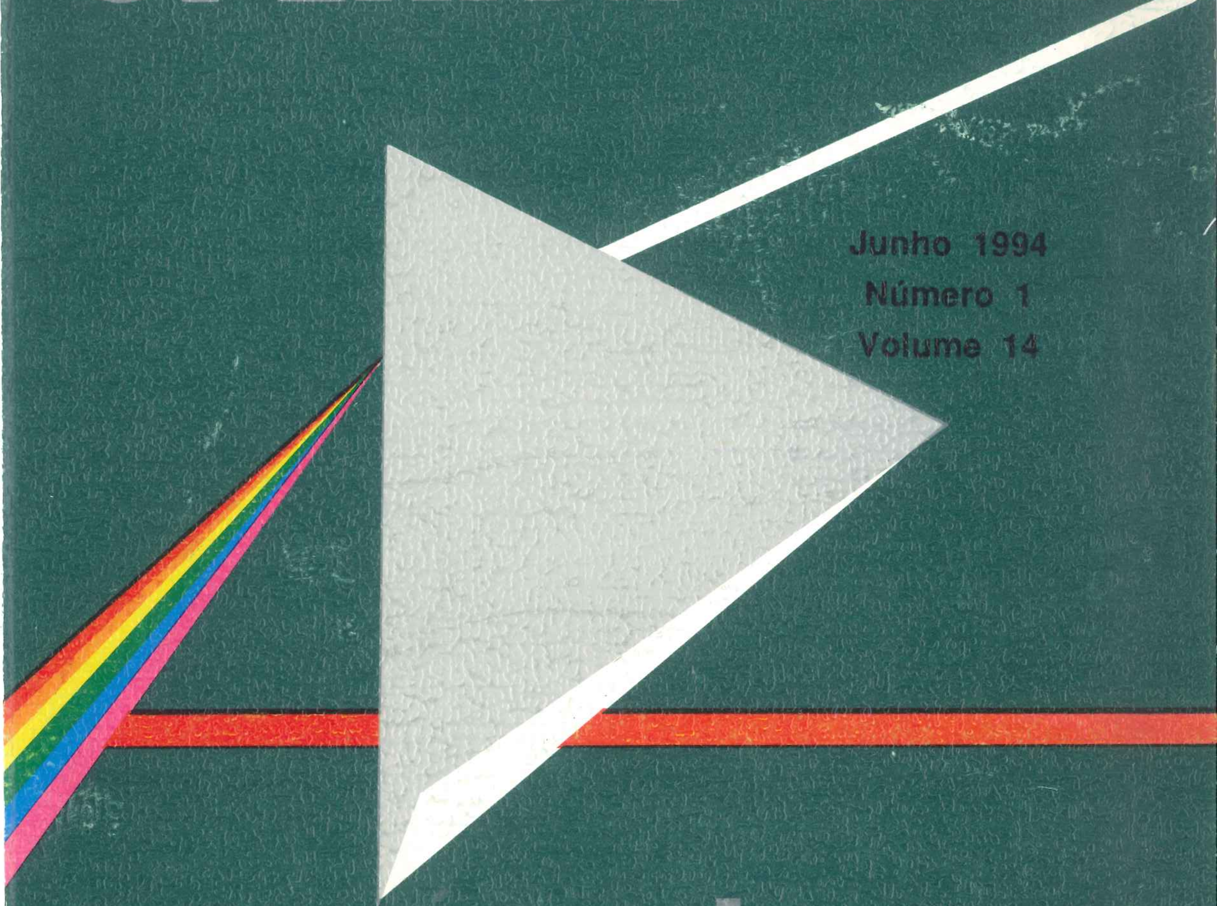
INVESTIGAÇÃO OPERACIONAL



Junho 1994
Número 1
Volume 14

Publicação Científica da

Associação Portuguesa para o Desenvolvimento
da Investigação Operacional

INVESTIGAÇÃO OPERACIONAL

Propriedade:

APDIO — Associação Portuguesa para o Desenvolvimento
da Investigação Operacional

ESTATUTO EDITORIAL

«Investigação Operacional», órgão oficial da APDIO cobre uma larga gama de assuntos reflectindo assim a grande diversidade de profissões e interesses dos sócios da Associação, bem como as muitas áreas de aplicação da I. O. O seu objectivo primordial é promover a aplicação do método e técnicas da I.O. aos problemas da Sociedade Portuguesa.

A publicação acolhe contribuições nos campos da metodologia, técnicas, e áreas de aplicação e software de I. O. sendo no entanto dada prioridade a bons casos de estudo de carácter eminentemente prático.

Distribuição gratuita aos sócios da APDIO

INVESTIGAÇÃO OPERACIONAL

Volume 14 - nº 1 - Junho 1994

Publicação semestral

Editor Principal: Joaquim J.Júdice
Universidade de Coimbra

Comissão Editorial

defence. The distinctive approach is to build a model of the system incorporating measurement of factors such as chance and risk, with which to predict and compare the outcomes of alternative decisions, strategies or controls. The purpose is to help management determine its policy and actions scientifically", in accordance with the definition of OR by the Operations Research Society.

Transportation R&D Programmes in the EEC, (DRIVE), as well as in US, (IVHS), and in Japan, (AMTICS, VICS), are looking for innovative solutions based on the applications of the new informatics and telecommunication technologies: the so called Advanced Transport Telematics, ATT Systems. In the EEC, DRIVE has identified seven Areas of major Operational Interest, (DRI92):

Area1: Demand Management

Covers the use of technology in helping urban authorities and the managers of urban transport to strike an efficient balance between traveller's demands and preferences and the capacity of the road and rail network, in accordance with the transport policy of the administrations.

Area2: Travel and Information Systems

This area addresses issues relating to the collection, processing and distributions of travel and traffic information of direct use to people at home, in the office and in the course of a journey, whether as driver of vehicles, public transport passengers or as fleet operators. Work covers organisational and social considerations related to the provision of information and the response of the users.

Area3: Integrated Urban Traffic Management Systems

The work in this area intends to improve and integrate transport systems in use in cities. This addresses traffic control, route guidance, travel and traffic information, parking management, emergency management and environmental control, tidal flow systems, traffic prediction tools and O/D flow estimations.

Area4: Integrated Inter-Urban Management Systems

Address systems for traffic control and driver information on motorways and parallel roads. Projects deal with automatic incident detection including techniques of image processing.

Area5: Driver Assistance and Cooperative Driving

Systems will be developed to assist the driver and to communicate information between vehicles. Attention is given to man-machine interaction techniques for improving the effectiveness of the systems.

Area6: Freight and Fleet Management

Freight and logistic management systems enabling inter-modal operations are being developed. Special attention is given to special transport like hazardous and perishable goods.

Area7: Public Transport Management

Tra attractiveness of public transport is crucial to sustain mobility in the future. Projects in this area develop, implement and test integral vehicle scheduling and control systems for inter-urban and rural transport applications and possibilities for integration with urban networks, user information in vehicles and stops.

DRIVE is envisaging the implementation of the ATT systems in an integrated way leading towards what is called the Integrated Road Transport Environment, (IRTE).

To cope with this integrated view of the transportation system one needs to go one step further, from the classical view of traffic control, that performs control over time, to the concept of Traffic Management System, which extends control over the space too (BAR91b). The conceptual approach to traffic management systems has been precisely formulated by Improta et al. (IMP86):

"The pattern of traffic in an urban area is the result of interaction between people's wish to travel or move goods in the area and the available road system, including the regulations governing its use and any control system that is in operation. The pattern of traffic itself gives rise to traffic and environmental conditions which may themselves both influence the location and timing of activities, and hence the demand for movements, and give rise to ideas and presures for changes in the road system and its managements and control."

A Traffic Management System which integrates control over time and over space requires:

a) Practical methods of measuring the degree of change in network traffic flows resulting from system modifications.

b) Real-time identification of imbalance situations in the use of available capacity.

c) Definition and assessment of real-time management decisions and control measures.

To build a traffic management system meeting these requirements is necessary a good understanding of the interactions that hold in a transportation system. A conceptual model of these interactions, with special emphasis on the interactions between public policy and user preferences, was formulated by the study of the Organisation for Economic Co-operation and Development in 1987, (OEC87). This model, displayed in figure 1, represents the interactions between the physical transportation system and the socioeconomic activity system, via the equilibrium process, to produce a set of flows on the links of the network.

The operation of this model of a Transportation System Management (TSM) can be explained as follows: In the System-Loop, the traffic manager assesses the system's performance according to his measures of effectiveness and intervenes in the physical transportation to achieve the best trade-off between the individual's and the community's interests. The tripmakers, on the other hand, perceive the flows according to their own values, which may be different from those of the traffic manager, and propagate an adjustment in the travel demand pattern via the User-Loop. While the traffic manager can intervene fairly quickly to further his objectives, we assume that the reactions of the trip makers have a longer-time constant. Therefore, in a short-range analysis, the demand pattern may be regarded as fixed.

To successfully develop and implement a traffic management system with all these features the following components are required:

1. Traffic models that efficiently represent the described interactions dynamically.
2. Advanced tools for the definition and the assessment of the management strategies.
3. Decision Support Systems embedding these models to enable an effective implementation of these management concepts.

And in designing and building such components is when we enter in the real of Operations Research.

Figure 2 displays graphically the conceptual architecture of such Traffic Management System.

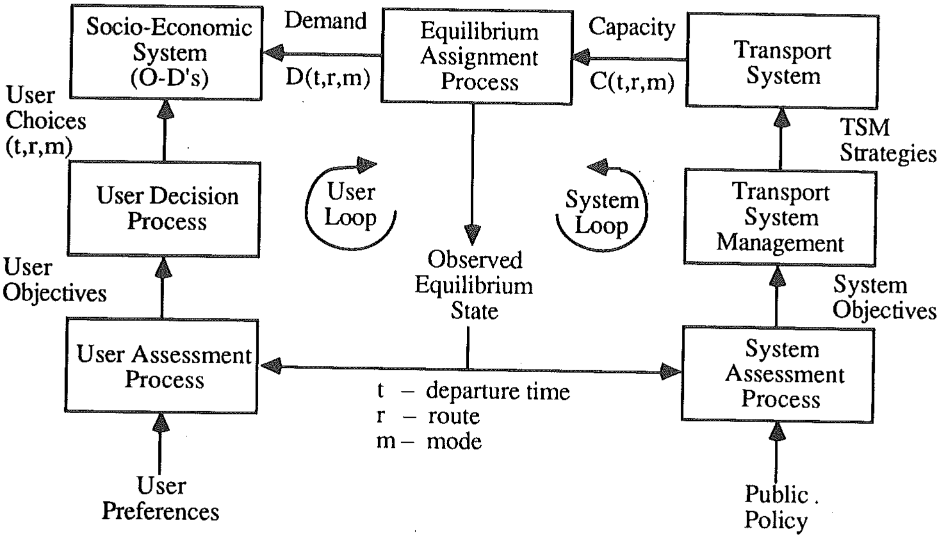


Fig. 1: Transport System Interactions

2. The Architecture of Decision Support Systems for Traffic Management

A widely accepted architecture for a modern Decision Support System, (DSS), is the one proposed by Sprague in 1986, (SPR86).

The main components of the proposed architecture are: a Database, a Model Base, and a Software System linking the user to each of them. The Database and the Model Base have some interrelated components, and the Software System comprises three sets of capabilities: Database Management Software (DBMS), Model Base Management Software (MBMS), and the software for managing the interface between the user and the system, called Dialogue Generation and Management Software (DGMS). These three major subsystems provide the scheme of the technical capabilities that a DSS must have.

When these concepts are applied to the architecture of a Decision Support system for Traffic Management, the Database must represent in a suitable way the underlying road network structure, Model Base must be understood as the set of interrelated traffic models used to deal with traffic representation and transport interactions as described, (examples of such models may be found in (BAR91b)), and the dialogue generation and management software may be conceived as a set of graphic interfaces comprising tools for traffic model building manipulating, and interpreting model results.

2.1 ASTERIX: A Decision Support Systems for Evaluating RTI Systems

An example of the proposed decision support systems architecture to assist traffic managers decisions, is provided by the system ASTERIX, developed by DRIVE project V1054, (BAR91a), and (AST92).

ASTERIX stands for A Simulation Tool for Evaluating Road Informati-X, and was a DRIVE I project which developed general purpose traffic simulation software for assessment of Road Transport Informatic Systems, (RTI), i.e. software guaranteeing a high degree of portability and user-friendliness, incorporating various modelling capabilities, using improved versions of already existing simulation systems, and designed easily to integrate future simulators with new modelling capabilities.

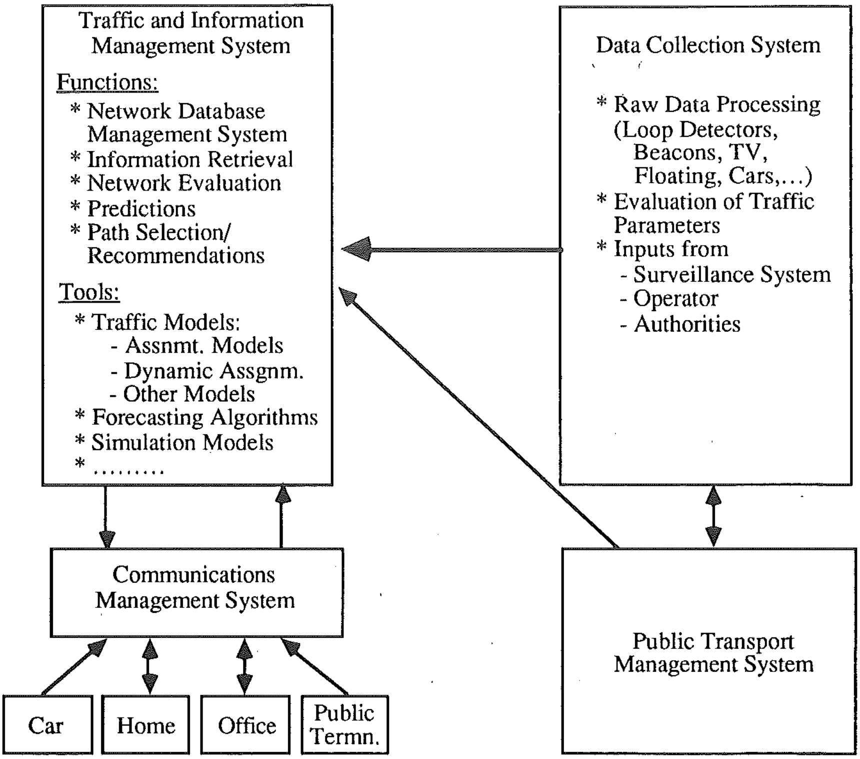


Fig. 2: Architecture of the Traffic Management System

ASTERIX has developed a software simulation environment embedding certain traffic simulation systems (SATURN, CONTRAM and SITRA-B+), designed according to the described specifications of the architecture of Decision Support Systems. ASTERIX has developed a basic conceptual approach consisting of a System Shell that supplies such a software environment, linking the user to a multimodel transportation network database and to the above mentioned traffic simulation models. These work at different levels, in complementary scenarios, simulating different RTI systems.

ASTERIX Software Environment for traffic simulation has the following features:

- Utilities for input data generation
- Assistance in building the simulation model
- Interactive design and running of simulation experiments
- Animated graphic outputs of simulation results

The approach taken in ASTERIX assumes that RTI systems will work at different levels, each one requiring a specific modelling approach. According to the level at which they will work, RTI systems can be classified as follows:

a) **Regional level**, for instance route guidance or travel information systems. Modelling at that level needs a macroscopic approach, for instance a traffic assignment approach, like SATURN, (HAL80), (VLI82).

b) **Intermediate size networks**, as for instance dynamic guidance or driver information systems, automatic debiting, tidal flow control systems, or incident management systems. Modelling at that level requires a microscopic approach, such as the heuristic dynamic assignment approach implemented in CONTRAM, (LEO82), (LEO89).

c) **Local level**, as will be the case for dynamic guidance, demand responsive traffic control systems, or road pricing systems in urban areas. Such a level can be properly modelled through a microscopic simulation approach, which SITRA-B+, (AST91), or AIMSUN, (BAR89), are examples.

ASTERIX provides integration between the three levels, guaranteeing consistent information exchange between them, in such a way that the user could communicate with the system by specifying a triple (level, RTI, simulator), with the following interpretation:

- **Level**, at which the RTI system will work
- **RTI** system to be assessed at that level
- **Simulation system** to be used in the assessment

The conceptual structure of the system, displayed in the figure below, (Fig. 3), can be interpreted as a system composed of a Software Environment for Simulation (or Software System Shell) interfacing the user, into which are plugged in different traffic simulation systems, like SATURN, CONTRAM, SITRA-B+, etc., that work at different levels, in complementary scenarios, simulating different RTI Systems.

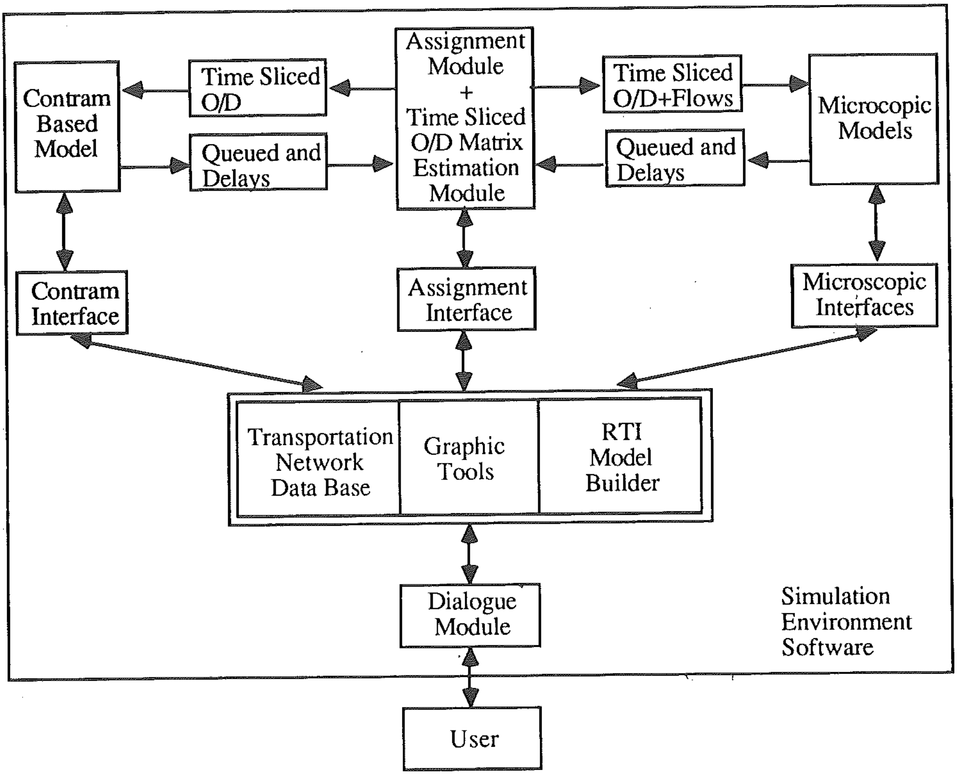


Fig. 3 – Conceptual Structure of the Integrated Simulation System

The kernel of ASTERIX, the software for the dialogue generation and management, is the so called ASTERIX Shell, composed of the User-System Graphic Interface, the Interface with the Database System and the Interface with the simulation systems integrated in the Shell. The figure 4 shows the ASTERIX Shell Subsystems and interfacing architecture.

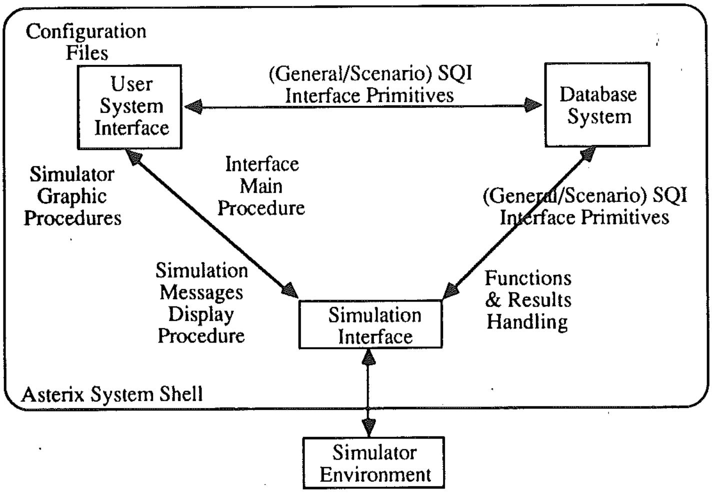

Configuration Files
User System Interface
(General/Scenario) SQI Interface Primitives
Database System
Simulator Graphic Procedures
Interface Main Procedure
(General/Scenario) SQI Interface Primitives
Simulation Messages Display Procedure
Functions & Results Handling
Simulation Interface
Asterix System Shell
Simulator Environment

Fig. 4 – The ASTERIX Shell

The User System Interface enables the manipulation of the graphic shell components, the simulation data and models, in a user friendly way, helping the user to manage and edit the transportation network information (demand, behavioural functions, traffic control, and so on), create simulation scenarios information, run the simulations without leaving the environment, and analyse and display graphically the results. These functionalities are performed by a set of graphic editors, components of the Graphical User Interface. The Database Management System (DBMS), contains shared and centralized information required by different simulators, and has been implemented as a hierarchical relational, SQL based DBMS.

The Databse Structure is partitioned into several hierarchical areas:

- General Common Area shared by all simulators;
- Simulator area shared by all simulators;
- Simulator Specific Areas;

The diagram in figure 5 represents this structure:

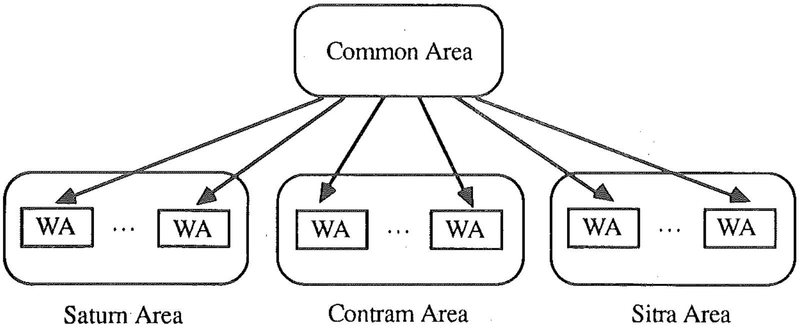
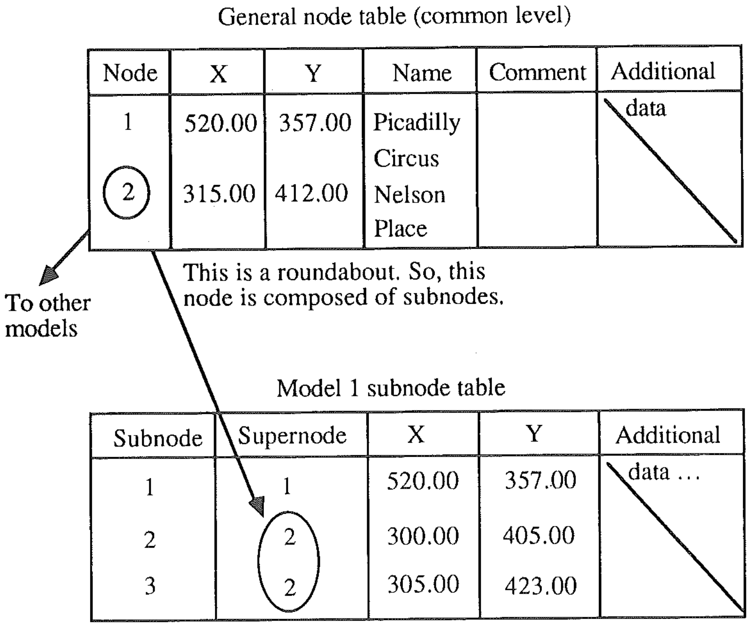


Fig. 5 – Database Structure
*WA Means Working Area

Given the various informations to provide to each traffic simulation model, data are stored into tables dedicated to common information and tables dedicated to specific information: Node table, Link table, Demand table, several Behaviour tables providing common parameters, and so on. The same set of tables is used for dedicated information belonging to a particular traffic simulation model.

To illustrate this subject, the figure 6 shows the node tables. The figure presents the internal aspects of node organization. The general node table contains node information at a common level. The common level is a large scale level (macroscopic). Any node defined at a macroscopic level may be a collection of nodes when modifying the current mode (i.e pass from a macroscopic mode to a microscopic one). All tables of the database follow the same principles, organized into a top-down hierarchy.



General node table (common level)

Node	X	Y	Name	Comment	Additional
1	520.00	357.00	Picadilly Circus		data
2	315.00	412.00	Nelson Place		

To other models

This is a roundabout. So, this node is composed of subnodes.

Model 1 subnode table

Subnode	Supernode	X	Y	Additional
1	1	520.00	357.00	data ...
2	2	300.00	405.00	
3	2	305.00	423.00	

Fig. 6

In terms of the Database subsystem a scenario is defined as the *subset of data that define a network and a signal setting system, in a given simulation data model*.

Scenarios are grouped in Working Areas and *are specific for a simulator*. There are elements within a Working Area, like the time slicing pattern, the O/D pairs structure, O/D matrices, Behavioural functions or parameters, that are common to all scenarios in the Working Area, while other elements, like the network, or the signal setting system are specific for a scenario.

There are two kinds of scenarios in a Working Area:

- Original scenario. This is the one directly created from the common area of the database and is the first one that a Working Area contains.
- Child scenarios. These scenarios can be considered as modified versions of the original scenario. They can be modified, deleted or duplicated.

The relationship between original scenarios, child scenarios, working areas and simulation systems is shown in figure 7.

Scenarios correspond to simulation experiments to be conducted in a particular network or subnetwork, for specific conditions, using a defined simulation system. The set of Simulation System Interfaces enables the user to run the simulation without leaving the System Shell, and analyse the simulation results.

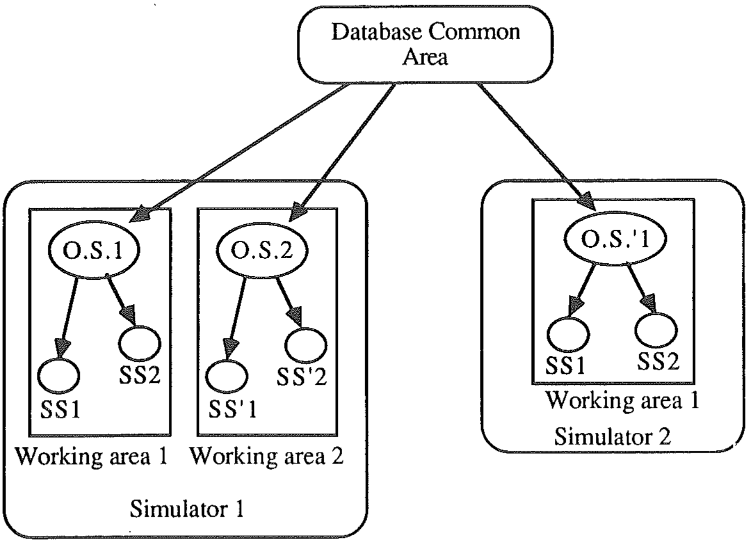


Fig. 7 - Scenarios
(O.S.i means Original Scenario number i and ssj means child scenario number j)

3. Advanced Tools for the Definition and Assessment of RTI Based Management Strategies

3.1 Assignment Based Procedures

The main traffic models, to estimate the distribution of traffic flows on a road network, are based on mathematical models of route choice, that is, the modelisation of how users select their routes under the prevailing traffic conditions, as required by the transport system interactions illustrated in the diagram in figure 1.

The concept of equilibrium plays a central role in this model building process. Wardrop, (WAR52), stated the two principles that formalised this concept of equilibrium, and introduced the behavioural postulate of the minimisation of total costs, that along with the principles are the fundamental modelling hypothesis. Traffic equilibria models are descriptive models that are aimed at predicting link flows and travel times that result from the way in which users choose routes from their origins to their destinations in a transportation network, (FLO86).

The road network is modeled in terms of a graph, whose nodes $n\varepsilon N$, represent origins, destinations, and intersection of links, and links, $a\varepsilon A$, represent the transportation infrastructure. The flow of trips on a link a is given by v_a, and the cost of travelling on a link is given by a user cost function $s_a(v)$, where v is the vector of link flows over the entire network. As these functions model the time delay for a travel on arc a, are called volume/delay functions.

The origin to destination demands, g_i, $i\varepsilon I$, where I is the set of origin/destination pairs, O/D, may use directed paths k, $k\varepsilon K_i$, where K_i is the set of paths for O/D pair i. The flows on paths k, h_k, satisfy flow conservation and nonnegativity conditions:

$$\sum_{k\in K_i} h_k = g_i, \forall i\in I$$
$$h_k \geq 0, k\in K_i, \forall i\in I \tag{1}$$

The corresponding link flows v_a are given by:

$$v_a = \sum_{i\in I} \sum_{k\in K_i} \delta_{ak}h_k, \forall a\in A \tag{2}$$

where:

$$\delta_{ak} = \begin{cases} 1 & \text{if link a belongs to path k} \\ 0 & \text{otherwise} \end{cases}$$

The network equilibrium model is formulated by supposing that for every O/D pair Wardrop's user optimal principle is satisfied, or in other words, that all the used directed paths are of equal cost, that is:

$$s_k^* u_i^* = \begin{cases} = 0 & \text{if } h_k^* \geq 0 \\ & k\in K_i,\ i\in I \\ \geq 0 & \text{if } h_k^* = 0 \end{cases} \tag{3}$$

over the feasible set (1)-(2).

This network equilibrium model may be restated in the form of a variational inequality:

$$(s_k^* - u_i^*)(h_k - h_k^*) \geq 0, k\in K_i, i\in I \tag{4}$$

where h_k is any feasible path flow. If $h_k^* > 0$, then $s_k^* = u_i^*$ since h_k may be smaller than h_k^*, if $h_k^* = 0$ then the inequality is satisfied when $s_k^* - u_i^* \geq 0$.

By summing over $k\varepsilon K_i$, and $i\varepsilon I$, and taking into account constraints (1) and (2), when the demand g_i is constant, model (4) can be reformulated as follows, (FIS83), (MAG84), (DAF80):

$$s(v^*)^T(v-v^*) \geq 0 \tag{5}$$

which is the variational inequality formulation derived by Smith, (SMI79).

When the user cost functions are separable, that is, depend only on the flow in the link: $s_a(v) = s_a(v_a)$, $a\varepsilon A$, the variational inequality formulation has the following equivalent convex optimization problem:

$$\begin{aligned} \text{Min} \quad & \sum_{a\in A} \int_0^{v_a} s_a(x)\, dx \\ \text{s.t.} \quad & \sum_{k\in K_i} h_k = g_i, \forall i\in I \\ & h_k \geq 0, k\in K_i, i\in I \end{aligned} \tag{6}$$

and the definitional constraint of v_a, (2).

Although the traffic assignment problem is a special case of nonlinear multicommodity network flow problem, and may be solved by any of the methods used for the solution of this problem, more efficient algorithms for solving this problem, based on an adaptation of the linear approximation method of Frank and Wolfe, (FRW56), have been developed in the past years, (LEB'75), (NGU76), (FLO76). Other eficient algorithms based on the restricted simplicial approach have been developed more recently by Hearn et al., (LAW82), (GUE82), or on an adaptation of the parallel tangents method, (PARTAN), (FLO83).

The above described procedures for static, and in some cases stochastic, user equilibrium assignment, are implemented in software systems like SATURN, (Simulation and Assignment of Traffic to Urban Road networks) developed at the Institute for Transport Studies, University of Leeds, (HAL80), (VLI82), or EMME/2, (Equilibrium Model-Modéle d'Equilibre), (INR92), developed at the Centre de Recherches sur les Transports, Université de Montréal, and others.

3.2 Modelling Vehicle Guidance Systems Using Traffic Assignment Models

Traffic Assignment Models can be modified and enhanced in order to enable it to assess the potential impact of route guidance strategies or related traffic information systems at a macroscopic level on network performance, on relatively large metropolitan networks (e.g. Barcelona). (AST90) reports the modification of the suite of SATURN programs for such purpose. Two general approaches are provided.

In the first a classical equilibrium (or stochastic equilibrium) assignment model is run to provide an average long-term flow pattern. Next, individual days are simulated by introducing various combinations of randomised trip matrix elements, random network parameters (e.g. capacities), etc. Travel times per link are then calculated and for each day a fraction of trips may then be re-assigned to "optimum" guided routes and the new resulting travel times assessed.

In the second a multiple user class assignment procedure has been implemented in which the trip matrix is divided into various classes with different levels of network information assumed. This guided trips may be modelled as having perfect - or near perfect - information; non-guided vehicles are subject to errors. Each class is then assigned under a framework which correctly allows for the interaction between the different flows in order to provide an average long-term pattern.

The basic modelling process is illustrated in the figure 8. Thus there are three essential steps:

1. The modelling of a long-term "steady state" assignment process using an average trip matrix plus average network conditions and based on either Wardrop Equilibrium (UE) or Stochastic User Equilibrium (SUE).
2. The simulation of a single day in which random fluctuations are postulated in the trip matrix (to represent day-to-day variability in demand) but the trips are assigned to the same routes and in the same proportion as in step 1. Since this will result in different link flows it will also result in different link travel times. Hence the O/D travel times along the long-term routes will also vary and need not correspond to

the conditions originally achieved; in particular routes which were minimum time (or cost) under step 1 no longer need to be minimum time/cost routes under step 2.

3. Given that the routes in step 2 no longer need to be optimal, it now becomes possible to re-assign a certain fraction of trips from their current routes to the actual "optimum" routes, thus reproducing the ability of a route guidance/driver information system to assist drivers to change to optimal routes.

A third approach, with the objective of estimating the number of vehicles that would need to be equipped to monitor link travel times adequately during peak periods, has been proposed by Boyce, (BOY91), and implemented in the feasibility study of the ADVANCE project in Chicago.

To obtain such numerical estimate, the following approach has been devised:

1. Solve a static, user-optimal route choice model for the road network under consideration for the concerned peak period. In the Chicago project a model in which traffic signal settings are adjusted in relation to link flows was applied.
2. Assume that the vehicles described by this model are selected randomly to be equipped with navigation systems.
3. Select a trip at random with regard to its origin, destination, route and departure time.
4. Record which network links each selected vehicle traverses during each 5 minute interval of the peak commuting period.
5. Tabulate the number of network links traversed by at least one vehicle during 5, 10, 15 and 20 minute intervals, versus the number of vehicles sampled.

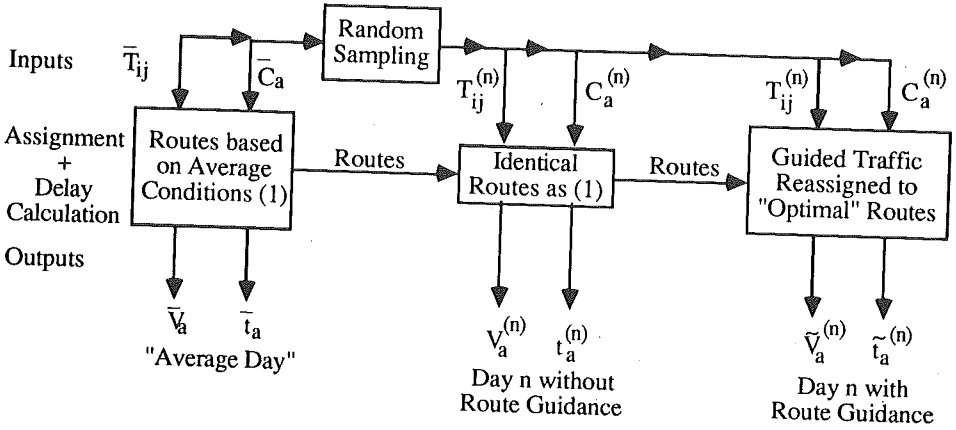


Fig. 8

To date the above steps have been tested to ascertain that they work as required but no tests on "real" networks have yet been carried out. There are clearly a very large number of parameters/options which might be varied in carrying out such tests, including:

1. Basing the steady state assignment on either WE or SUE assumptions.

2. The degree of variability in the trip matrix.
3. The degree of variability in day-to day network supply.
4. The percentage of guided trips.
5. The "reliability" of the optimum routes (a lack of perfect system information could be simulated by introducing a randomness into the link times as used by DBLOAD to provide "shortest routes").
6. Re-assignment according to minimum time or minimum marginal time (corresponding to user-optimised or system-optimised assignment).
7. Repeating the process for a number of "days" to achieve statistical accuracy.

The guidance model has been tested in two real-life networks: Weetwood (an area of Leeds) consisting of 70 zones, 104 intersections and 440 links; and Barcelona comprising 110 zones, 820 intersections and 2547 links. For each network the guidance model has been implemented under:

a) Three different demand levels.
b) Nine different levels of equipped vehicles.
c) Three different routing criteria.

The numerical results of the corresponding computational tests, have been reported in (AST90), and (VUR91).

3.3 Multi-User Traffic Assignment Models

The multi-user traffic assignment model is a generalization of the usual assignment model that allows the user to consider more than one class of vehicles. Each class of vehicles has its own O-D matrix and its own cost function vector. The flow vectors comprise, for each link, one flow variable for each class. The interactions between the different classes lead to asymmetric cross-link effects, and then no Beckman-type equivalent optimization problem is known to exist.

We must then use the variational inequality formulation of the problem:

$$s(v^*)^T(v-v^*) \geq 0 \qquad (5)$$

subject to (1) and (2). Alternative algorithms have been provided to solve this equilibrium problem. A family of algorithms is based on the diagonalization approach, according to Mahmassani, (MAH88), and Sheffi (SHE85). A separable-cost assignment problem is obtained, by fixing the cross-link effects at their current level, and is called the diagonalized subproblem. The diagonalization algorithm is an iterative one whose the main loop fixes the cross-link effects at their current level and then solves the diagonalization subproblem. We propose to solve the subproblems using the Frank-Wolfe iterative method. An iteration of the Frank-Wolfe method is called inner iteration to be distinguished from the outer iteration of the main loop. The subproblems are not to be solved exactly. At each outer iteration, a given number of inner iterations are performed. Numerical results show that the algorithm seems to converge the most quickly when two inner iterations are performed each time.

Another family of algorithms is bases on the use of projection methods, (DAF80), and in particular the one proposed by Bertsekas and Gafni, (BER82). Quite recently, a combination of Bertsekas projection method and Simplicial Decomposition, (BAR91d), has proven its efficiency in using the methodology proposed by Van Vliet, described in paragraph 3.2, for dealing with vehicle guidance systems when behaviour of guided and non guided vehicles are modeled as different user classes with interactions.

The concept of stochastic equilibrium assignment and multiple user classes, (when it is possible to divide the demand, a priori, into "user classes", where the cost definitions are the same within a user class, but may differ between classes) were brought together by Daganzo, (DAG83). He considered a family of link cost function of the form

$$c_{ai} = d_{ai} + \beta_i t_a(\mathbf{F})$$

where for each link a and user class i,

c_{ai} is the cost to user class i of using link a

$F_a = \sum_j a_j f_{aj}$

$F = (F_1, F_2, \ldots)$

f_{ai} is the flow of user class i on link a

t_a is a continuously differentiable function, and

d_{ai}, a_i (≥ 0), b_i (≥ 0) are finite constants

The perceived cost of travel C_{ai} for link a and user class i is assumed to be given by

$$C_{ai} = c_{ai} + \Phi_{ai}$$

where F_{ai} is the random variable which represents the perception error. Daganzo imposed two sets of conditions, firstly on the cost functions and secondly on the perception errors, under which a stable equilibrium flow pattern exists and is unique:

1. The inverse of t(F) is a monotonically increasing, continuously differentiable function in the domain where it takes finite values, is defined for all t, and is uniformly bounded.
2. For each user class i, the components of $F_i = (F_{1i}, F_{2i}, \ldots)$ are mutually independent, independet of the costs c_{ai}, and have densities which are finite, have at most a finite number of descontinuities and have finite second moments.

Daganzo proposed two solution algorithms for the multiple user class SUE problem, one in terms of link costs and the other based on flows. And ad hoc version of this later has been implemented within SATURN, and has been used for the assessment of route guidance strategies, (VUR89), (VUR91).

3.4 Looking for Real-Time Applications: Parallelization of Traffic Models

Real-time applications on the traffic and transportation domain are clear candidates for implementation on high performance computers due to their computing requirements. Hardware developments in recent years have enabled the development of real-time applications of limited size, the extensions of these applications to large traffic networks will be made possible by the development of parallelized versions of most of the traffic models to fully exploit the advantages and power of modern high performance computers. Consequently an intense research has been undertaken, and is currently in progress, in the field of the parallelization of network flow models, and their applications to traffic problems.

The first step has been the parallelization of shortest path algorithms, key component of most network flow algorithms.

3.4.1 Parallelization of Shortest Path Algorithms

This is the primary step to parallelize any transportation algorithm provided that almost all the main algorithms have to solve some kind of shortest path problem as an intermediate step or subproblem.

For any kind of transportation, network, the determination of shortest paths - in terms of distance, time, or any other criterion of generalized cost - between different nodes of the network, is one of the key problems in most transportation applications, or appears as one of the main subproblems in most optimization algorithms for transportation problems.

Alternative approaches have been proposed, (MOH88), (BER91), althoug the parallelization of algorithms based on the dual "auction" approach developed by Bertsekas, (BER92) and (PAL91), seems to be among the more promising.

3.4.2 Parallelization of Static Traffic Algorithms

Depending on the kind of application two different parallelization approaches can be used, a parallelization in the space of link flows or a parallelization in the space of the path flows.

The sequential versions of the algorithms for these problems, as presented in 3.1, are habitually formulated on the space of link flows, (FLO84), (FLO86). The model can be stated as the convex optimization problem:

$$\begin{aligned} &\text{Min} \quad \sum_{a\in A} \int_0^{v_a} s_a(x)\,dx \\ &\text{s.t.} \quad \sum_{k\in K_i} h_k = g_i,\ \forall i\in I \\ &\qquad\quad h_k \geq 0,\ k\in K_i,\ i\in I \end{aligned} \tag{6}$$

and the definitional constraint (2) in 3.1.

The most used algorithmic approaches to solve these problems are based on ad hoc adaptations of the Linear Approximation Method, (LAM), of Frank and Wolfe, (FLO86), which lie in the following algorithmic framework:

STEP 1: Initialization: Find a feasible initial solution v; $s_a = s_a(v_a)$, $a\varepsilon A$

STEP 2: Solution of the linearized subproblem at iteration 1

$$\text{Min} \sum_{i\in I} s_k^1 f_k$$

$$\sum_{k\in K_i} f_k = \bar{g}_i, \forall i\in I$$

$$f_k \geq 0, \forall k\in K_i, \forall i\in I$$

where s_k^1 is the cost on path k at iteration 1. For each O/D pair i finds the shortest path k_i^*, and performs an all-or-nothing assignment:

$$f_a = f_a + g_i, \ a\varepsilon k_i^*$$

STEP 3: Stopping Test

STEP 4: Find the optimal step size:

$$\lambda^* = \text{Arg Min} \sum_{a\in A} \int_0^{v_a+\lambda(f_a-v_a)} s_a(x)\ dx$$

$$0 \leq \lambda \leq 1$$

STEP 5: Update variables v_a and costs $s_a(v_a)$.

$$v_a = v_a + \lambda^* (f_a - v_a)$$

$$s_a = s_a(v_a)$$

Return to step 2

The LAM approach is a natural candidate to a parallelization approach consisting of the distribution of the link computations between the processors network. This is recommendable first step in the migration of an already existing assignment algorithm based on that LAM approach, as indicate the explorative research done by Chen and Meyer, (CHE88), and the numerical experiments of Babini, (CHB90) on a network of transputers.

A coarse-grain parallelization of steps 2, 3 and 5, partitioning the computations into tasks in order to minimize the message exchange between processors, has been proposed, and successfully computationally tested by Chabini et al., (CHB93). Parallelization of step 2 involves the parallelization of Shortest Path Trees computation from each origin to all destinations. Step 3 can be restated in terms of the annulation of the gradient of the function to be minimized:

$$\lambda^* = \left\{\lambda\in(0,1): \sum_{a\in A} s_a\left[v_a+\lambda(f_a-v_a)\right](f_a-v_a) = 0\right\}$$

and step 5 concerns the simultaneous evaluation of a set of independent functions.

3.4.3 Restricted Simplicial Decomposition

For the extensions to the dynamic approaches it seems more recommendable to look for algorithms with a better convergence rate than the Linear Approximation Method. A promising candidate is the approach based on the exploitation of second order Quasi-Newton information for a problem formulated on the space of path flows.

The proposed approach is an extension of the Restricted Simplicial Decomposition for convex nonlinear problems, (HOH77), and a reformulation of the link-flow restricted simplicial algorithm for traffic assignment problems due to Hearn et al., (HEA87). At each iteration the decomposition has to solve a master nonlinear problem for which the Newton projection method of Bertsekas, (BER82), is specially well suited.

A sequential version of that approach was successfuly implemented and tested in the scope of ASTERIX DRIVE I project, (AST91). In this formulation the constraint matrix has a block-diagonal shape very well suited for parallelization.

The proposed parallelization approach for this formulation of the problem is based on a network partitioning to exploit the features of the constraint matrix, and a parallelization of the Newton projection methods to fully exploit the structure of the master problem.

3.5 Origin-Destination Estimates

The statement of the first mathematical programming models for estimating or updating Origin/Destination matrices from traffic counts appears quite early in the transportation research literature, most of them being contemporary of the first formulations of the traffic assignment problems as a mathematical programming problem. The papers of Robillard, [ROB75], and Nguyen, [NGU77], among others, are a good example. An interesting formulation dated back on these early times is the statement of the problem as a combined assignment-distribution problem, [ERN79], for which approaches based in mathematical programming procedures - mainly the Benders dual decomposition - were proposed, [JÖR79]. The text book of Sheffi, [SHE85], gives a good overview of these formulations.

The reason for that interest is clear, as Van Zuylen and Willunsen pointed out in a paper, [VZW80], that has now become almost a classic: the usual method for obtaining an Origin/Destination trip matrix employs a combination of home interviews and roadside surveys, whose high cost precludes its use for most other applications. Alternative methods - number plate surveys, for instance - suggested for small scale studies, tend also to be expensive in terms of manpower requirements and/or processing. That makes appealing the resort to methods based on one of the most common pieces of traffic information, traffic counts, since they are relatively unexpensive to and are usually collected for several purposes and their automation is relatively advanced, mainly in those urban areas with advanced traffic control systems requiring real-time traffic data collection.

The advent of the Advanced Transport Telematic Applications, as envisaged in R&D Programmes like the DRIVE EEC Programme has increased this interest in the last years, given that the performance of most of these applications will rely directly on the quality of the information collected, and among the required information that regarding the mobility in a given area plays a crucial role. A good example of that are any kind of trip information systems, and dynamic route guidance systems, see for instance [BEL91], or the final report of DRIVE I project V1047, ODIN, [ODI92], that required a continuous updating of O/D matrices to provide the dynamic input data requested by many of the proposed procedures for estimating and predicting traffic conditions and travel times on the road network, see for instance the TANGO report [TAN92].

A dynamic update of O/D matrices is also a requisite for advanced planning systems based on dynamic assignment approaches which take also into account day-to-day and within day variations of traffic flows, and simulation tools for the assessment of RTI applications. See for instance DRIVE I project V1054 ASTERIX, final Report, [AST92].

In 1988, E. Cascetta and S. Nguyen, proposed an unified framework for the O/D estimation problems that today seems to be widely accepted, [CAS88], that includes the former formulations as particular cases. According to their poposal the **Trip Demand Estimation Problem** is defined as: *"Determine an estimate of the O/D trip demand matrix G by efficiently combining traffic count based data and all other available information"*.

More recently, as part of the basic research in DRIVE I project V1054, ASTERIX, Lundgren, [LUN91], [AST91], has proposed a more comprehensive framework in which the O/D estimation problem can be formulated as:

$$\begin{aligned} &\text{MIN } Z(G,V) = \alpha_1 F_1(G,\hat{G}) + \alpha_2 F_2(V,\hat{V}) \\ &\text{s.t.} \\ &\qquad V = \text{assign } (G) \\ &\qquad\quad G \in M \end{aligned} \qquad [P1]$$

where $F_1(G,\hat{G})$ is any suitable distance measure between the estimated O/D matrix G and the observed target matrix $\hat{G}$, and $F_2(V,\hat{V})$ is any suitable distance measure between the estimated link flows V and the given observed flows $\hat{V}$, available for a sybset $\hat{A} \subset A$ of links. All models assume F_1 and F_2 to be convex functions, and they may be for example the quadratic or entropy functions of the models so far studied.

An interesting feature of this framework is that problem [P1] can be interpreted in the scope of the multiobjective programming as a two-objective mathematical programming problem in which α_1 and α_2 are the corresponding weighting factors expressing the relative importance of the two measures, thet can be related, for instance, with the reliability of the observed data, for example, an old, possibility outdated, demand matrix G, and quite good real-time flow $\hat{V}$ measurements.

Assign (G) denotes the flow version of the assignment map of the matrix G onto the network. The constraint **V = assign (G)**, defines the estimated link flows V as a function of the O/D matrix G through the assignment. M is the set of feasible O/D matrices, e.g. defined by marginal totals, by a constraint on the total number of travellers, or simply by non-negativity constraints.

While the framework of Cascetta and Nguyen mainly addresses the featuring of the distance measures leading to the different estimation models, Lundgren's framework classifies the models into two groups based on the assumptions made for the assignment procedure used. Models in the first group assume that the cost (travel time) of traversing a link in the network is independent of the flow on the link. These models are only concerned with uncongested networks, where the path proportions are proportional to the path costs which are not flow dependent. This means that **assign (G)** is given a priori, based on a simple all-or-nothing assignment or on some proportionate assignment procedure, and that the constraints describing the relationships between link flows v_a and demand matrix G ca be rewritten as:

$$v_a = \sum_{i \in I} p_{ai}\ g_i,\ \forall a \in A$$

where p_{ai} is now the proportion of usage of link a for the flow between the i-th O/D pair. Models using this type of assignment cannot deal accurately with congestion effects, but are computationally tractable since the problem to solve becomes a convex problem.

A version of the general model as multi-objective mathematical programming model using entropy measures in the definitions of F_1 and F_2 has been proposed by Brenninger-Göthe et al., [BRE89].

Models proposed by Maher [MHE83], Cascetta [CAS84], and the general models of Cascetta and Nguyen framework [CAS88], fit into this framework when functions F_1 and F_2 are defined either by entropy functions or by functions using dispersion matrices.

The limitations and drawbacks of these models, because of their lack of adequacy to the congested networks, lead to define general models for the congested case when the proportions are flow dependent and, consequently, demand dependent. A way of considering implicitly that dependency is assuming that **assign** (G) is given by an equilibrium assignment. That leads to the so called bi-level approaches, and correspond to the second group of models in Lundgren's framework and include an assumption of equilibrium assignment in the model. Problem [P1] is reformulated then as:

$$\begin{aligned} &\text{MIN } Z(G,V) = \alpha_1\ F_1(G,\hat{G}) + \alpha_2\ F_2(V,\hat{V}) \\ &\text{s.t.} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad [P2] \\ &\qquad G \in M \end{aligned}$$

where

$$V(G) = \text{assign } (G)$$

This means that the link flows are assumed to satisfy the user-equilibrium conditions. This formulation has a bilevel structure in which an optimization problem comprises a constraint which is itself an optimization problem, the equilibrium assignment problem in this case, formulated as , (6) in 3.1., [FLO86].

The first models which took explicitly into account the dependencies between proportions and flows, tried to avoid, as far as possible, to deal directly with the bilevel structure of the problem. The family of models based on Nguyen, [NGU77], and later extended by Jörnsten and Nguyen, [JÖR79], and Nguyen, [NGU83], can be considered as a special case of [P1], when α_2 = C and V is assumed to reproduce the observed path costs given by $\hat{V}$.

Although theoretically interesting, due to practical and computational problems, most of the models proposed for the O/D matrix estimation problem have only been applied to problems of small size.

Until quite recently no attempts have been made to deal directly with the bilevel formulation [P2] of the O/D estimation problems, due to the difficulties inherent to that formulation. In general, bilevel programming problems are difficult to solve because of their inherent nonconvexity and nondifferentiability.

In general, a bilevel programming problem, Bard, [BRD88], can be expressed as follows:

$$\begin{array}{ll} \underset{x}{\text{MIN}} & F(x,y) \\ \text{subject to} & G(x,y) \leq 0 \end{array}$$

where y is obtained by solving

$$\begin{array}{ll} \underset{x}{\text{MIN}} & f(x,y) \\ \text{subject to} & g(x,y) \leq 0 \end{array}$$

In game theory, this problem is called a Stackelberg game with an upper-level vector of decision variables x for the leader, and a lower-level vector of decision variables y for the follower. It is assumed that the leader is given the first choice and selects an x in accordance with his constraints to minimize his objective function F, while tacking into account the reaction of the follower. In light of this decision, then selects a y according to his constraints to minimize his objective function f, [BRD88].

We will now consider the application of the Stackelberg game framework to the O/D matrix estimation problem where origin-destination flows are the upper-level decision variables.

In the scope of Lundgren's framework, Spies, [SPI90], has formulated problem [P2], with $\alpha_1 = C$, M defined to include only non-negativity constraints and $F_2(V,\hat{V})$ defined as

$$F_2(V,\hat{V}) = \frac{1}{2} \sum_{a\in A} (v_a - \hat{v}_a)^2$$

under the assumption that the observed link flows $\hat{v}_a$ are an equilibrium flow pattern in the sense of Wardrop's first principle, therefore the O/D matrix obtained is such that when assigned to the network according to equilibrium principles, generates link volumes which constitutes a local minimum with respect to the given distance measure F_2. The method is applied to situations where the target matrix is assumed to be reasonable accurate, and where the traffic count information is used for updating or adjusting this target matrix. The target matrix is only used as a starting point in the solution method. Computational experience for large-scale problems is reported.

Given the already mentioned inherent difficulties to the bilevel approaches, Spiess method is heuristic in nature, of steepest descent type, and no guarantee that a global optimum to the formulated problem will be found.

The iterative heuristic works as follows:

At iteration k:

- Given a solution g_i^k, an equilibrium assignment is solved giving link flows v_a^k, and proportions $\{p_{ia}^k\}$ satisfying the relationship

$$v_a^k = \sum_{i\in I} p_{ia}^k g_i^k, \forall a\in A$$

Note: the target matrix is used in the first iteration (i.e. , $g_i^1 = \hat{g}_i, \forall i\in I$)

- The gradient of the objective function $F_2(V,\hat{V})$ is computed. For a more realistic approach the gradient is based on the relative change of the demand, written as:

$$g_i^{k+1} = \begin{cases} \hat{g}_i \text{ for } k = 0 \\ g_i^k \left(1 - \lambda^k \left[\dfrac{\delta F_2(g)}{\delta g_i}\right]_{g_i^k}\right) & \text{for } k = 1,2,3,\ldots \end{cases}$$

(Then a change in the demand is proportional to the demand in the initial matrix and zeros will be preserved in the process).

The gradient is approximated by

$$\frac{\delta F_2(g)}{\delta g_i} = \sum_{k \in K_i} h_k \sum_{a \in \hat{A}} \delta_{ak} (v_a - \hat{v}_a),\ i \in I$$

(where $\hat{A} \subset A$ is the subset of links with flow counts).

- The step length is approximated as:

$$\lambda^* = \frac{\sum_{a \in \hat{A}} v'_a(\hat{v}_a - v_a)}{\sum_{a \in \hat{A}} v'^2_a}$$

where

$$v'_a = -\sum_{i \in I} g_i \left(\sum_{k \in K_i} \sum_{a \in \hat{A}} \delta_{ak} (v_a - \hat{v}_a)\right) \left(\sum_{k \in K_i} \delta_{ak} h_k\right)$$

One of the advantages of the proposed heuristic is that it can be easily implemented using the EMME/2 transportation planning software, [INR92].

The main computational effort in this approach comes from the solution of the assignment problems. The use of an algorithm which have nice re-optimization features, providing direct information on path flows, as the Restricted Simplicial Decomposition with Disaggregated Representation, Larsson and Patrikson, [LAR92], [AST91], makes this approach even more attractive.

An extension of Spiess procedure adding a function $F_1(G,\hat{G})$, of entropy type has been successfully tested by Lundgren, [LUN91], [AST91], on small networks.

Last but not least, Florian and Chen, [FLC93], develop a Gauss-Seidel type algorithm for the general problem [P2] of adjusting an O/D matrix by using observed flows in congested networks. They derive necessary conditions for some properties of the solution of the O/D matrix adjusting problem and develop a coordinated descent method. The method provides an interpretation and characterization of the Spiess and Yang methods. Florian method is also heuristic, and seems to perform well when the adjustments to the O/D matrix are relatively small and hence, the assumption of fixed path proportions nearly holds. The method can also be easily implemented using the EMME/2 transportation planning software. Computational experience with large networks is reported.

3.6 Dynamic Assignment Models

In the static assignment models we have assumed that the demand is constant over time. This assumption is realistic for the analysis of intercity freight transport networks over long periods of time where the traffic flows supposedly have reached some steady state. As for a city, the traffic over its network of streets is not constant over a day. During peak periods, in general, heavy congestion occurs on strategic links of the network, and a static model is no longer sufficient to explain the traffic flows; this has motivated the study of dynamic models in which interest has been increased by the requirements of modern management systems.

The implementation of Traffic Management Systems or Traffic Information Systems with the features described in the introduction, able to work in real-time requires algorithmic developments in two major areas of interest:

Dynamic Traffic Assignment as basis for network state estimation and forecasting, and
Traffic Simulation Models as tools for the assessment of the recommended policies.

The importance of Dynamic Traffic Assignment to deal with the dynamic effect of traffic flows on transportation networks was soon identified in the framework of DRIVE and IVHS Programmes, since then a number of researchers has been working in this area and some sequential codes, (some of them evolved from the seminal paper by Merchant and Nemhauser, (MER78), in 1978) implementing the first algorithmic approaches, are being currently tested computationally, (COD92), (BOY91b), (OMA92a), (MAH90).

The CPU requirements of most algorithmic approaches for the dynamic problem are usually so big for the performance of the current workstations, that in those cases when the researchers, (MAH90a), (BOY91b), had available the access to more powerful computers (Dual Cyber 170/750, for instance) they have used them to conduct the computational experiments.

The CPU requirements of these problems, and the possibility of parallelizing substantial parts of the code, (PIN90), (ZEN88), (SCH91), shows clearly that these classes of problems are natural candidates to be solved on parallel computers.

That has also been recognized in the main reports presenting DRIVE or IVHS Programmes, (DRI92), where the use of parallel computers to cope with these problems is suggested.

An algorithm based on a dynamic formulation ofthe equilibrium principles used in the static approaches, using an optimal control approach has been developed by Codina (COD92).

The sequential version of this algorithm discretizes the problem over the time horizon and solves a static problem at each time interval. The problem can then be considered as a flow assignment to a hyper-network built explicitly on the space-time. The approach followed in (COD92), although inspired in (BOY91b), can be considered as an extension that takes into account explicitly the nonnegativity constraints over flow variables, and by developing an ad hoc version of optimal control problems with nonnegativity constraints over the control variables, formulates the problem as a primal problem on the hyper-network. One of the key aspects of this algorithm is the possibility of reoptimizing from one time interval to the next.

The restricted simplicial algorithm mentioned in 3.4.3 has proven to be very successful for these reoptimizations, and as a consequence a direct parallelization of (COD92) is envisaged.

3.7 Traffic Simulation Models as Tools for Assessment of the Recommended Policies

The key role of Traffic Simulation Models as assessment and evaluation tools of transportation policies has been widely recognized in all Traffic and Transportation R&D Programmes, (DRI92), mainly in the framework of the coming IRTE as consequence of the Advanced Transport Telematics Programmes. In DRIVE I specific projects of the Programme, as the ASTERIX Project, (AST92), (BAR91a), have had as main objective the development of traffic simulation systems for such objectives, as well as the software environments that will make them accessible for the final users, (BAR91c).

The possibilities of dealing with large networks is directly related to the available computer power. Because of that most of the new simulation modeling developments, (CHA85), have used directly big computers (Dual Cyber 170/750), or parallel computers in projects related to IVHS, (MAH90b).

The requirements of resorting to high performance computer capabilities to achieve full real-time traffic management and information systems was explicitly recognized in DRIVE II Workplan, and is being used in the american IVHS Programme, and the japanese AMTICS, (a System with models and features quite close to the ones described in the introduction to this paper has already been implemented at HITACHI Research Laboratories, and is starting to be tested on a hypercube transputer network).

Traffic Simulation uses mainly two complementary approaches depending on the features of the transportation network to be simulated. For urban networks, or networks in general where traffic interruptions due to traffic lights may occur, a microscopic approach based on a car by car simulation has proved to be the most suitable. For freeway or motorway networks where vehicles flow continously without interruption, other than the ones derived from the actual traffic conditions, a macroscopic approach based on a hydrodynamic simile gives better results. A Traffic Management Application need the two kinds of traffic simulators provided that both kinds of transportation infrastructures coexist in large metropolitan areas. In the past parterns of this Consortium have developed traffic simulators of both types, therefore the task is splitted into two subtasks addressing the parallelization of each type of simulator.

3.7.1 Microscopic Models

Car-following models are used to describe the behaviour of the driver-vehicle system in a stream of interacting vehicles, and provide the basic component of microscopic traffic simulation models.

Car-following models consist of differential difference equations giving the acceleration of a vehicle with respect to the behaviour of the preceding ones. A basic exposition of car-following theory can be found in (GAZ74), and a recent overview in (GAB91). The general form of car-following models can be represented by the expression:

response(t+T) = sensitivity * stimulus(t)

where T is the reaction time of the driver-vehicle system. In most models the response will always be the acceleration (decceleration) of the following car, and the stimulus of the difference in velocity between the lead car and the follower.

Simple linear car-following models formulate the above expression in terms of a second order differential difference equation function of parameter T and a sensitivity coefficient a, that are usually estimated emprically, although theoretical analysis, (HER59), can establish upper bounds on the relationships between a and T to ensure the stability of the model:

$$\ddot{x}_{n+1}(t+T) = a[\dot{x}_n(t) - \dot{x}_{n+1}(t)]$$

Simple models are not very satisfactory because they do not account properly for stability. Stability of a traffic model means that changes in velocity by the lead vehicle of a stream of cars will not be amplified by successive vehicles in the stream until a collision occurs. There are two types of stability. Local stability considers the response of a vehicle to the change in motion of the vehicle immediately in front of it, while assymptotic stability deals with the propagation of a fluctuation through a platoon of vehicles.

Non-linear Car-following models postulate more complex relationships for a more realistic description of car's behaviour and that asks for more parameters whose values have also to be empirically estimated. More recently Gipps, (GIP81), (GIP86), proposed a new car-following model based on the assumption that each driver sets limits to his desired bracking and acceleration rates.

The model has two components, which cover acceleration and braking separately. For acceleration:

$$\dot{x}^{a}_{n+1}(t+T) = \dot{x}_{n+1}(t) + 2.5\ \alpha_{n+1}T\left[1\ \frac{\dot{x}_{n+1}(t)}{V_{n+1}}\right]\left[0.025 + \frac{\dot{x}_{n+1}(t)}{V_{n+1}}\right]^{1/2}$$

where $\dot{x}^{a}_{n+1}(t+T)$ is the maximum speed to which vehicle n+1 can accelerate during the time interval (t,t+T), Vn+1 is the desired speed for vehicle n+1 and an+1 is the maximum acceleration for vehicle n+1.

For braking:

$$\dot{x}^{b}_{n+1}(t+T) = \beta_{n+1}T+\left([\beta_{n+1}T]^2 - \beta_{n+1}\left\{2[x_n(t)-1_n-x_{n+1}(t)] - \dot{x}_{n+1}(t)\ T\ \frac{\dot{x}^2_{n+1}(t)}{\hat{\beta}}\right\}\right)^{1/2}$$

where $\dot{x}^{b}_{n+1}(t+T)$ is the maximum safe speed for vehicle n+1 with respect to vehicle n, β_{n+1} is the most severe braking the driver of vehicle n+1 can undertake, 1_n is the effective length of vehicle n, and $\hat{\beta}$ is the estimate of β_n used by the driver of vehicle n+1. In any case the speed of vehicle n+1 is:

$$\dot{x}_{n+1}(t+T) = \mathrm{MIN}\left\{ \dot{x}^{a}_{n+1}(t+T),\ \dot{x}^{b}_{n+1}(t+T) \right\}$$

The model has been used to simulate vehicular traffic in multilane arterial roads with special attention devoted to the structure of lane-changing decisions. An adaptation of this model has been used in the design and implementation of the microscopic traffic simulation model AIMSUN, (BAR89), (BAR91).

Microscopic simulation models are specially suited to reproduce accurately actual traffic conditions on road networks on a computer, this feature makes microscopic simulators the most suitable tool for assessing the effects of RTI systems. An example of that is the use of AIMSUN that is being made in the context of DRIVE Project SOCRATES for the research on vehicle guidance systems based on the use of cellular radio. AIMSUN has been adapted to deal with guided vehicles on the network. One of the key aspect to study is the data collection from the equipped vehicles. On board equipment is able to produce a wide set of traffic related data that will be sent to the Traffic Information Centre.

Data collection must supply the raw data after a suitable processing will produce road network data of a sufficiently high quality to enable the production of good navigation data bases. This information must be presented in a form that can be processed by in-car computer to identify the best routes, taking into account not only the prevailing traffic conditions but also forecasts, covering the whole duration of the trip, updated periodically.

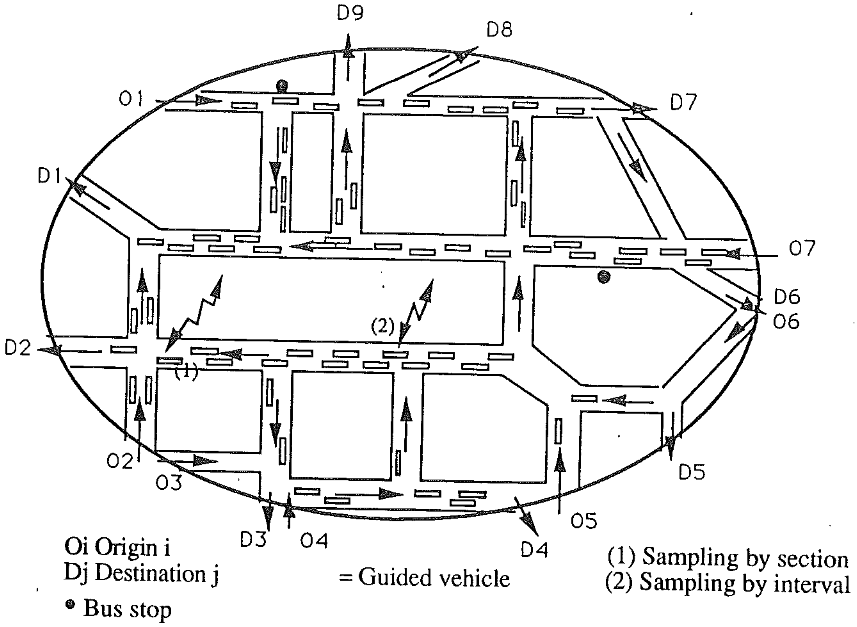
In an initial phase, as well as during the transient period of operation of a guidance system, until a threshold of penetration of equipped cars has been reached, the number of equipped cars will be limited, and consequently the number of uplink messages, i.e. those transmitted by the cars to the base stations, will be insufficient to represent traffic conditions only by themselves.

Research Methodology is based on building a microscopic simulation model of an urban area large enough to be sensitive to dynamic guidance able of simulating the behaviour of each equipped vehicle, reproducing their on-board data collection/processing their two way communications, to investigate two main aspects. Decide which is the most suitable message contents, both from the communications point of view, to not incur in unacceptable communication overheads, and from the point of view of data requirements for sound estimates of traffic conditions and traffic forecasting, looking for a balanced solution. For example, the concatenation of traffic data into longer messages, which may be more efficient for certain communication options may be of less value for traffic predictions.

Second, seek to optimize the sampling procedure. A potential advantage of a mobile two way communications technology is that it makes posible a variety of sampling procedures to collect data from equipped vehicles, going, for instance, from an emulation of bacon based systems to a variety of poolling-type strategies.

Of special interest is the dual mode sampling of data from floating cars: section and time. The equipped car would send a message each time a specific position on a road section is reached (i.e. the start or the end of the section). However, taking into account that in urban areas there are a lot of reasons for stopping a vehicle on the road without congestion, for example passenger drop-off, waiting behind a loading vehicle to merge to the other lane, stopping for pedestrians who want to cross the road and so on, a stop can only be verified by evaluation of the period after the stop. That means a complete section analysis has to be done for the evaluation of travel times. On the other hand on motorways, because of the potential danger of a stop (emergency or congestion), a message containing the exact positions should be sent to the Traffic Information Centre nearly immediately. Section messages can bear too long delay because the vehicle is stuck in the congestion and cannot reach the end of the section, or the beginning of the next, within the next minutes. Therefore is suggested a dual mode sampling of the in-car data: *a vehicle should send a message at the end or the beginning of every section and every $\Delta\tau$ minutes as well.*

The simulation methodology is illustrated in the following figure showing how a microscopic simulation model looks like. The model, as approximate representation of the real world, reproduces approximately the actual geometry of the simulated road network and the conditions constraining traffic flows, i.e. traffic lights. Cars behaviour is simulated on a car by car basis, according to leader-follower and lane change models, reproducing in that way the behaviour of guided cars and the proposed sampling procedure (a guided car follows a predefined route between its origin Oi and its destination Dj in the network sending messages to the central computer in accordance with a sampling rule)



The simulation experiments seek to optimize both the sampling of the data and its transmission to the information centre.

3.7.2 Macroscopic Approaches: Hydrodynamic Models

Mathematical modelling of traffic flow dynamics is a prerequisite for a number of important tasks including traffic surveillance and monitoring, incident detection, systematic control strategy design, simulation and forecasting. Macroscopic traffic flow models are based on a hydrodynamic analogy by regarding traffic flow as a particular fluid process whose state is characterized by aggregate macroscopic variables such as traffic density (in veh/Km), traffic volume (in veh/h), and mean speed (in Km/h). The story of dynamic macroscopic modelling probably starts with the celebrated paper by Lightill and Whitham, (LIG55), that describes the dynamic evolution of macroscopic traffic variables on highway by means of:

(i) A partial differential equation (in space and time) reflecting the law of conservation of vehicles

(ii) A static volume-density relationship which is broadly known as the fundamental diagram of Traffic Engineering.

Later on, Payne, (PAY71), (PAY79), suggested an improved model by replacing the static relationship (ii) with a second partial differential equation which corresponds to the momentum equation in fluid dynamics and which includes the fundamental diagram as a special case. Payne's model provided good results under certain traffic conditions, but was found to lack accuracy under dense traffic conditions near on-ramps and/or lane drops. A number of introduced extensions, (PAP89), (PAP90), (ROS88), (MIC91), (KÜH89), contributed to an accuracy improvement of Payne's model, most of these extensions include a relaxation term that represents the tendency of traffic flow to adjust speeds due to changes in free-flow speeds along the roadway, a traffic friction term that models traffic friction at freeway ramp junctions due to ramp flows, as function of a friction parameter which depends on the ramp volumes entering or leaving and is estimated empirically, and an anticipation term that represents the effect of drivers reacting to downstream traffic conditions. Considerable research (see (PAD91) for an overview) in the last two decades has contributed substantially to the understanding and the quantitative description of traffic flow dynamics.

The simple continuum traffic model, as formulated by Payne, (PAY71), consists of the conservation equation:

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = g(x,t) \qquad (7)$$

where k(x,t), and q(x,t) are the traffic density and flow respectively at the space-time point (x,t). The generation term g(x,t) represents the number of cars entering or leaving the traffic flow in a`freeway with entries/exits. The traffic flow, density and speed are related by the equation:

$$q = ku \qquad (8)$$

where the equilibrium speed u(x,t) = u(k) must be provided by a theoretical or emprical u-k model, equation of state, that can take the general form:

$$u_e = u_f \left[1-(k/k_{jam})^{\alpha}\right]^{\beta} \qquad (9)$$

where u_f is the free flow speed, and k_{jam} the jam density.

Since the simple continuum model does not consider acceleration and inertia effects, it does not faithfully describe non-equilibrium traffic flow dynamics. This is taken into account in the high-order continuum formulation which replaces equation (3) with a momentum equation:

$$\frac{\partial k}{\partial t} + u\frac{\partial q}{\partial x} = \frac{1}{T}\left[u_e(k) - u\right] - \frac{\nu}{k}\frac{\partial k}{\partial x} \qquad (10)$$

where T is the relaxation time, n is the anticipation parameter, the first term in the right hand side is the relaxation to equilibrium, that is the effects of drivers adjusting their speeds to the equilibrium speed-density relationship, and the second term represents the anticipation, that is, the effect of drivers reacting to downstream traffic conditions.

To numerically integrate these equations, (MES90), (PAP89), PAP90a), (CHR92), MIC91), each traffic model of the road section (space dimension) is discretized in time and space. Numerical methods which are used in computational fluid dynamics can be applied to solve these equations, (HIR88).

Some of these methods, like the LAX, the TRAPEZOIDAL or the EULER method, already implemented in early sequential versions, are good candidates for parallelizåtions in transputer networks. For instance in LAX method a parallelization consisting on partitioning the freeway in as many main sections as processors, in such a way that adjacent road sections be assigned to adjacent processors, appears as the most natural way of parallelizing these models.

4. References

[1] (AST90) ASTERIX DRIVE Project 1054, Deliverable 4, (1990).

[2] (AST91) T.Larson and M.Patrikson, Simplicial Decomposition with Disaggregate Representation: An Application to the Traffic Assignment Problem, DRIVE I, ASTERIX (V1054), Project, Deliverable 7, DRCO, Brusseles, 1991 (To appear in Transportation Research).

[3] (AST92) ASTERIX, DRIVE Project V1054, Final Report, DRIVE Central Office, Brussels 1992, (prepared by J.Barceló).

[4] (BAR89) J. Barceló, J. Ferrer and L. Montero, AIMSUN: Advanced Interactive Microscopic Simulator for Urban Networks. Vol. I: System Description, Vol. II: User's Manuel, Departamento d'Estadística i Investigació Operativa, Universitat Politécnica de Catalunya, 1989.

[5] (BAR91a) J. Barceló, Software Environments for Integrated RTI Simulation Systems, Proceedings of the DRIVE Conference on Advanced Telematics in Road Transport, Brussels, Elsevier, 1991.

[6] (BAR91b) J. Barceló, Traffic Management Systems, in M. Papageorgiou (ed.), Concise Encyclopedia of Traffic and Transportation Systems, Pergamon Press, 1991.

[7] (BAR91c) J. Barceló, Urban Traffic Simulation: Software Environments, in M. Papageorgiou (ed.), Concise Encyclopedia of Traffic and Transportation Systems, Pergamon Press, 1991.

[8] (BAR91d) J. Barceló and L. Montero, A Simplicial Decomposition Approach for Solving the Variational Inequality Formulation of the General Traffic Assignment Problem for Large Scale Problems, Paper presented at XI EURO Conference, Aachen, 1991.

[9] (BEL91) M. Bell, D. Inaudi, J. Lange, M. Maher, Techniques for the Dynamic estimation of O/D Matrices in Traffic Networks, Proceedings of the DRIVE Conference, Brussels, (1991).

[10] (BER82a) D.P. Bertsekas and E.M. Gafni, Projection Methods for Variational Inequalities with Application to the Traffic Assignment Problem, Mathematical Programming Study 17 (1982) 139-159.

[11] (BER82b) D.P. Bertsekas, Projected Newton Methods for Optimization Problems with Simple COnstraints, SIAM Journal of Control and Optimization 20 (1982) 221-246.

[12] (BER91) D.P.Bertsekas, An Auction Algorithm for Shortest Paths, SIAM Journal on Optimization 1 (1991) 425-447.

[13] (BER92) D.P.Bertsekas, Auction Algorithms for Network Problems: A Tutorial Introduction, Computational Optimization and Applications 1 (1991) 7-66.

[14] (BOY91a) D.E. Boyce, J. Hicks and A. Sen, In-vehicle Navigation Requirements for Monitoring Link Travel Times in a Dynamic Route Guidance System, paper presented at 70th Annual Meeting, Transportation Research Board, Washington D.C., 1991.

[15] (BOY91b) D.E. Boyce, B. Ran and L.J. Leblanc, Dynamic User-Optimal Traffic Assignment: A New Model and Solution Technique, paper presented at the First Triennial Symposium on Transportation Analysis, Montréal, 1991.

[16] (BRD88) J.F. Bard, Convex two-level optimization, Mathematical Programming 40 (1988) 15-27.

[17] (BRE89) M. Bremminger-Göthe, K.O. Jörnsten and J.T. Lundgren, Estimation of Origin-Destination matrices from traffic counts using multi-objective programming formulations, Trnaspn. Re. 23B (1989) 257-269.

[18] (CAS84) E. Cascetta, Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator, Transpn. Re. 18B (1984) 289-299.

[19] (CAS88) E. Cascetta and S. Nguyen, A unified framework for estimating or updating Origin/Destination matrices from traffic counts, Transpn. Res. B, 22B (1988) 437-455.

[20] (CHA85) G.L. Chang, H.S. Mahmassani and R. Herman, A Macroscopic Traffic Simulation Model to Investigate Peak-Period Commuter Decision Dynamics, Transportation Research Record 1005 (1985) 107-120.

[21] (CHB90) I. Chabini, Des Implantations Parallèles de l'Algorithme d'Approximation Linéaire pour la Résolution di Problème d'Affectation du Trafic, M'emoire de Maitrise, Département d'Informátique et Recherche Opérationelle, Université de Montréal, 1990.

[22] (CHB93) I. Chabini, O. Drissi-Kaïtouni and M. Florian, Solving the Network Equilibrium Problem on a Network of Transputers, Centre de Recherche sur les Transports, Université de Montréal, Publication 876, 1993.

[23] (CHE88) R.J. Chen and R.R. Meyer, Parallel Optimization for Traffic Assignment, Mathematical Programming 42 (1988) 327-345.

[24] (CHR92) A.T. Chronopoulos, P. Michalopoulos and J. Donohoe, Efficient Traffic Flow Simulation Computations, In: Mathematical and Computer Modelling 16 (1992) 107-120.

[25] (COD92) E. Codina and J. Barceló, An Algorithm for Extremals Calculation in Optimal Control Problems with Applications to the Dynamic Traffic Assignment Problem, Paper presented at the 39th North American Regional Science Association Meeting, Chicago, 1992.

[26] (DAF80) S. Dafermos, Traffic Equilibrium and Variational Inequalities, Transportation Science 14 (1980) 42-54.

[27] (DAG83) C. Daganzo, Stochastic Network Equilibrium with Multiple Vehicle Types and Asymmetric, Indefinite Link Cost Jacobians, Transportation Science 17 (1983) 282-300.

[28] (DRI92) DRIVE'92, Research and Technology Development in Advanced Road Tranport Telematics in 1992, Commission of the European Communities, DGXIII, DRI293.

[29] (ERN79) S. Erlander, S. Nguyen and N. Stewart, On the calibration of the combined distribution assignment model, Transpn. Res. 13B (1979) 259-267.

[30] (FIS83) C. Fisk and D. Boyce, Alternative Variational Inequality Formulations of the Network Equilibrium, Transportation Science 17 (1983) 454-463.

[31] (FLC93) M. Florian and Yang Chen, A coordinate descent method for the bilevel O-D matrix adjustment problem, Centre de Reserche sur les Transports, Université de Montréal (1993)

[32] (FLO76) M. Florian and S. Nguyen, An Application and Validation of Equilibrium Trip Assignment Methods, Transportation Science 10 (1976) 374-389.

[33] (FLO83) M. Florian, J. Guelat and H. Spiess, An Efficient Implementation of the PARTAN Variant of the Linear Approximation Method for the Network Equilibrium Problem, Publication 395, Centre de Recherche sur les Transports, Université de Montréal, 1983.

[34] (FLO84)M. Florian, An Introduction to Network Models Used in Transportation Planning, in Transportation Planning Models, M. Florian (ed.), North-Holland, pp.137-152, 1984.

[35] (FLO86) M. Florian, Nonlinear Cost Network Models in Transportation Analysis, Mathematical Programming Study 26 (1986) 167-196.

[36] (FRW56) M. Frank and P. Wolfe, An Algorithm for Quadratic Programming, Naval Research Logistic Quarterly 3 (1956) 95-110.

[37] (GAB91) J.F. Gabard, Car-Following Models, In: Concise Encyclopedia of Traffic and Transportation Systems, M. Papageorgiou (Ed.), Pergamon Press, Oxford, 1991.

[38] (GAZ74) D.C. Gazis, Traffic Science, John Wiley, 1974.

[39] (GIP81) P.G. Gipps, A Behavioural Car-following Model for Computer Simulation, Transp. Res. 15B (1981) 105-111.

[40] (GIP86) P.G. Gipps, A Model for the Structure of Lane-Changing Decisions, Transp. Res. 20B (1986) 403-414.

[41] (GUE82) J. Guélat, Algoritmes pour le Probleme d'Affectation du Traffic avec Demandes Fixes - Comparisons, Publication #299, Centre de Recherche sur les Transports, Université de Montréal, 1982.

[42] (HAL80) M.D. Hall, D. Van Vliet and L.G. Willumsen, SATURN: A Simulation-Assignment Model for the Evaluation of Traffic Management Schemes, Traffic Eng. Control 21 (1980) 168-176.

[43] (HEA87) D.W. Hearn, S. Lawphonpanich and J.A. Ventura, Resticted Simplicial Decomposition: Computation and Extensions, Mathematical Programming Study 31 (1987) 99-118.

[44] (HER59)R.C. Herman, E.W. Montroll, R.B. Potts and R.W. Rothery, Traffic Dynamics: Analysis of Stability in Car-following, Oper. Res. 7 (1959) 86-106.

[45] (HER92) R. Herman, Technology, Human Interaction and Complexity: Reflections on Vehicular Traffic Science, OR Forum, Operations Research 40 (1992) 199-212.

[46] (HIR92) C. Hirsch Numerical Computation of Internal and External Flows Vol.2, John Wiley and Sons, 1988.

[47] (HOH77) B. von Hohenbalken, Simplicial Decomposition in Nonlinear Programming Algorithms, Mathematical Programming 13 (1977) 49-68.

[48] (IMP86) G. Improta, R.E. Allsop and B.G. Heydecker, Network Models for Traffic Management, Centre de Recherches sur les Transports, Seminar on Transportation Systems, Université de Montréal, 1986.

[49] (INR92) INRO Consultants, EMME/2: Release 6.0, User's Manual 1992.

[50] (JÖR79) K. Jörnsten and S. Nguyen, On the estimation of a trip matrix from network data, Publication 153, Centre de Recherche sur les Transports, Université de Montréal, (1979).

[51] (KÜH89) R. Kühne, Microscopic Distance Strategies and Macroscopic Traffic Flow Models, in: CCCT'89, Cotrol, Computers, Communication in Transportation, AFCET, Paris (1989) 267-273.

[52] (LAR92) T. Larsson and M. Patriksson, Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem, Transp. Sci. 26 (1992) 4-17.

[53] (LAW82) S. Lawphongpanich and D.W. Hearn, Simplicial Decomposition of the Asymmetric Traffic Assignment Problem, Research Report 82-12, Departament of Industrial and Systems Engineering, University of Florida, Gainsville, 1982.

[54] (LEB75) L.J. LeBlanc, E.K. Morlok and W.P. Pierskalla, An Efficient Approach for Solving the Road Network Equilibrium Traffic Assignment Problem, Transportation Research 5 (1975) 309-318.

[55] (LEO82) D.R. Leonard and P. Gower, User Guide to CONTRAM version 4, Transport and Road Research Laboratory, Supplementary Report 735, TRRL, Crowthorne, UK.

[56] (LEO89) D.R. Leonard et al., CONTRAM - Structure of the Model, Department of Transport, Transport and Road Research Laboratory, Research Report 178, Crowthorne, UK.

[57] (LIG55) M.K. Lightill and G.B. Whitham, On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads, Proc. Royal Society of London, Series A 229 (1955) 317-345.

[58] (LUN91) J.T. Lundgren, Models for the OD-Matrix Estimation Problem, Dept. of Mathematics, Institute of Technology, Linkpöking, Working Paper, LiTH-MAT/OPT-WP-1991-14, (1991).

[59] (MAG84) T.L. Magnanti, Models and Algorithms for Predicting Urban Traffic Equilibrium, in: M. Florian ed., Transportation Planning Models, North-Holland, pp.153-186, 1984.

[60] (MAH88) H.S. Mahmassani and K.C. Mouskos, Some Numerical Results on the Diagonalization Algorithm for Networks Assignment with Asymmetric Interactions Between Cars and Trucks, Transp. Res. 22B (1988) 275-290.

[61] (MAH90a) H. Mahmassani and R. Jayakrishnan, Dynamic Analysis of Traffic Performance Under Real-Time Information, paper presented at the 69th Annual TRB Meeting, Washington, 1990.

[62] (MAH90b) M. Mahmassani, R. Jayakrishnan and R. Herman, Network Flow Traffic Theory: Microscopic Simulation Experiments on Supercomputers, Trans. Res. 24A (1990) 149-162.

[63] (MER78) D.K. Merchant and G.L. Nemhauser, A Model and an Algorithm for the Dynamic Traffic Assignment Problem, Trans. Sci. 12 (1978) 183-199.

[64] (MES90) A. Messmer and M. Papageorgiou, METANET: A Macroscopic Simulation Program for Motorway Networks, Traffic Engineering and Control 31 (1990) 466-470.

[65] (MHE83) M.J. Maher, Inferences on trop matrices from observations on link volumes: a Bayesian statistical approach, Transpn. Res. 17B (1983) 435-447.

[66] (MIC91) P.G. Michalopoulos, Ping Yi, D.E. Beskos and A.S. Lyrintzis, Continuum Modelling of Traffic dynamics, In.: Proc. of the 2nd Int. Conf. on Appl. of Advanced Tech. in Transportation Eng., 1991, Minneapolis, Minnesota, pp.36-40.

[67] (MOH88) T. Mohz and C. Pasche, A Parallel Shortest Path Algorithm, Computing 40 (1988) 281-292.

[68] (NGU76) S. Nguyen, A Unified Approach to Equilibrium Methods of Traffic Assignment, in: M. Florian ed., Traffic Equilibrium Methods, Lecture Notes in Economics and Mathematical Systems 118, Springer Verlag, pp.148-182, 1976.

[69] (NGU77) S. Nguyen, Estimating an OD matrix from network data: a network equilibrium approach, Publication 60, Centre de Recherche sur les Transports, Université de Montréal, (1977).

[70] (NGU83) S. Nguyen, Modéle de distribution spatiale tenant compte des itinéraires INFOR 21 (1983) 270-292.

[71] (ODI92) DRIVE I Project V1047 ODIN, Final Report, DRCO, 1992.

[72] (OEC87) Organisation for Economic Co-operation and Development, Dynamic Traffic Management in Urban and Suburban Road Systems, OECD, Paris, 1987.

[73] (OMA92a) Omar Drissi-Kaïtouni and Abdlhamid Hameda-Benchekroun, A Dynamic Traffic Assignment Model and a Solution Algorithm, Transportation Science 26 (1992) 119-128.

[74] (OMA92b) Omar Drissi-Kaïtouni and M. Gendreau, A New Dynamic Traffic Assignment Model, Centre de Recherche sur les Transports, Université de Montréal, Publication #854, 1992.

[75] (PAL91) S. Pallotino and M.G. Scutella, Strongly Polynomial Auction Algorithms for Shortest Paths, TR-19/91, Dipartimento di Informática, Universitá di Pisa, 1991.

[76] (PAP89) M. Papageorgiou, J.M. Blosseville and H. Haj-Salem, Macroscopic Modelling of Traffic Flow on the Boulevard Periphérique in Paris, Transportation Research 23B (1989) 29-47.

[77] (PAP90a) M. Papageorgiou, J.M. Blosseville and H. Haj-Salem, Modelling and Real-Time Control of Traffic Flow on the Southern Part of Boulevard Périphérique in Paris, Part I: Modelling, Transportation Research 24A (1990) 345-359.

[78] (PAP90b) M.Papageorgiou, J.M. Blosseville and H. Haj-Salem, Modelling and Real-Time Control of Traffic Flow on the Southern Part of Boulevard Périphérique in Paris, Part II: Coordinated On-Ramp Metering, Transportation Research 24A (1990) 361-370.

[79] (PAP91) M. Papageorgiou, J.M. Blosseville and H. Haj-Salem, ALINEA: A Local Feedback Control Law for On-Ramp Metering, Transportation Research Record 1320 (1991) 58-64.

[80] (PAY71) H.J. Payne, Models of Freeway Traffic and Control, Simulation Council Proc. 1 (1971) 51-61.

[81] (PAY79) H.J. Payne, FREEFLO: A Macroscopic Simulation Models of Freeway Traffic, Transpn. Res. Rec. 772 (1979) 68-75.

[82] (PIN90) M. Pinar and S. Zenios, Parallel Decomposition of Multicommodity Network Flows Using a Linear-Quadratic Penalty Algorithm, Tech. Report 90-12-06, Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, PA, 1990.

[83] (ROB75) P. Robillard, Estimating the OD matrix from observed link volumes, Transpn. Res. 9 (1975) 123-128.

[84] (ROS88) P. Ross, Traffic Dynamics, Transp. Res. 22B (1988) 421-435.

[85] (SCH91) G.L. Schulz and R. Meyer, An Interior Point Method for Block Angular Optimization, SIAM J. Optimization 1 (1991) 583-602.

[86] (SHE85) Y. Sheffi, Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods, Prentice-Hall, 1985.

[87] (SMI79) M.J. Smith, Existence, Uniqueness and Stability of Traffic Equilibria, Transportation Research 1B (1979) 295-304.

[88] (SPR86) R.H. Sprague, A Framework for the Development of Decision Support Systems, in Decision Support Systems: Putting Theory into Practice, R.H. Sprague and H.J. Watson, eds., Prentice-Hall, 1986.

[89] (SPI90) H. Spiess, A gradient approach for the O-D matrix adjustment problem, Publication #693, Centre de Recherche sur les Transports, Université de Montréal (1990).

[90] (TAN92) TANGO, DRIVE II Project V2054, Deliverable 6, Workpackage 7.3.3, (1992).

[91] (VLI82) D. Van Vliet, SATURN: a Modern Assignment Model, Traffic Eng. Control 23 (1982) 578-581.

[92] (VUR89) T. Van Vuren, D. Van Vliet and M.J. Smith, Combined Equilibrium in a Network with Partial Route Guidance, in: S. Yagar and S.E. Rowe (eds.) "Traffic Control Methods", Engineering Foundation, New York, pp.375-387, (1989).

[93] (VUR91) T. Van Vuren and D. Watling, Multiple User Class Assignment Model for Route Guidance, Institute of Transport Studies, University of Leeds, 1991.

[94] (VZW89) J.H. Van Zuylen and L.G. Willumsen, The most likely trip matrix estimated from traffic counts, Transpn. Res. 14B (1980) 281-293.

[95] (WAR52) J.G. Wardrop, Some Theoretical Aspects of Road Traffic Research, Proc. Inst. Civil Engineers, Part II, pp.325-378, 1952.

[96] (ZEN88) S.A. Zenios and R.A. Lasken, Nonlinear Network Optimization on a Massive Parallel Connection Machine, Annals of Operations Research 14 (1988).

INVESTIGAÇÃO OPERACIONAL, vol. 14, n.° 1 (Junho 1994) 35-59

THE WEISZFELD METHOD IN SINGLE FACILITY LOCATION

J.B.G. Frenk, M. T. Melo[1] and S. Zhang
Econometric Institute
Erasmus University Rotterdam
P.O.Box 1738
3000 DR Rotterdam - The Netherlands

Abstract

In this paper two algorithms based on the Weiszfeld method are proposed to solve the single facility continuous space location problem with distances modelled by some L_p-norm. We derive a generalization of the Weiszfeld method and prove its convergence for the Euclidean case (p=2) given some restrictions on the objective function related to quasiconvexity. We also show that the convergence property does not hold in general for other L_p-norms. Moreover, since the objective function is not everywhere differentiable, we use the well-known hyperbolic approximation to obtain an optimization problem which approximates uniformly the original one. An adapted version of the Weiszfeld method is then derived and its convergence is proved under some conditions for $1 < p \leq 2$. Furthermore, it is also shown that both algorithms have a linear rate of convergence provided certain stronger conditions are satisfied. Finally, some computational results are presented.

Resumo

Descrevem-se neste trabalho dois algoritmos baseados no método de Weiszfeld para a resolução do problema de localização simples num espaço contínuo envolvendo distâncias medidas por uma norma L_p. Começa-se por desenvolver uma generalização do método de Weiszfeld para o problema em estudo, provando-se que existe convergência no caso da distância Euclideana (p = 2) e desde que certas condições relacionadas com a quasi-convexidade da função objectivo sejam satisfeitas. Esta propriedade não é em geral válida para outras normas como se mostra neste artigo.

Dado que a função objectivo não é diferenciável em todo o espaço $\mathbb{R}^n$, utiliza-se uma aproximação hiperbólica obtendo-se um problema que aproxima uniformemente o original. Efectua-se em seguida uma adaptação do método de Weiszfeld ao novo problema e prova-se a convergência deste segundo método para os casos em que $1<p\leq 2$ mediante certas condições. Mostra-se ainda que ambos os métodos desenvolvidos convergem linearmente para a solução óptima se condições mais fortes forem verificadas. Finalmente, apresentam-se alguns resultados computacionais.

Keywords

Weiszfeld method, linear convergence, quasiconvexity.

1. Introduction

The single facility continuous space location problem concerns the location in n-dimensional space, $n \geq 2$, of a new facility so as to minimize a given function of the distances between the new facility and a set of existing facilities or demand points. Let $\mathbf{a}_j$, $j = 1,\ldots, m$ be

[1] This research was carried out at DEIO-Departamento de Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Bloco C2, Campo Grande, Cidade Universitária, 1700 Lisboa, Portugal.

m different demand points in $\mathbb{R}^n$ and $x \in \mathbb{R}^n$ the unknown location of the new facility. Moreover, let $\|\bullet\|_p$ be some L_p-norm with $p > 1$ and denote by $d_j(x) := \|x - a_j\|_p$, $j = 1, \ldots, m$ the distance between a_j and the new facility. Empirically for $n = 2$, it has been observed by Love and Morris [16], [17] that for some values of p belonging to [0.9, 2.29] the L_p-norm provides a better measure of actual travel distances than the Euclidean distance (p=2). Berens and Körling [5] applied the methodology of Love and Morris to Germany and, contrary to the proposal of these authors, observed that the optimal value for the parameter p was in most cases close to 2. Hence one may conclude that the choice of a distance function greatly depends on the configuration of the transportation network and that it is not always accurate to use Euclidean distances. Related papers discussing estimation of road distances are [4], [7], [8], [18], [24] and [25].

To introduce the objective of our optimization problem consider a nondecreasing function $g : S \rightarrow \mathbb{R}$ with $\mathbb{R}^m_+ \subseteq S$ and S open and suppose g is continuously differentiable on S with nonnegative gradient vector $\nabla g(z)$. Moreover, let the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be given by

$$f(x) = g(\mathbf{d}(x)) \tag{1}$$

with $\mathbf{d}(x)^T := (\|x - \mathbf{a}_1\|_p, \ldots, \|x - \mathbf{a}_m\|_p)$ and $p > 1$.

Since an increase in the distance between a new facility and a demand point usually causes an increase in transportation costs, the assumption that the function g is a nondecreasing function in each argument is quite natural. Furthermore, in order to assure that the algorithms presented in Sections 2 and 3 are properly defined we need to impose on the gradient vector of g a slightly stronger condition than nonnegativity. The single facility location problem is now defined by

$$\inf\{f(x) : x \in \mathbb{R}^n\} \tag{P}$$

with f given by (1).

A special instance of (P) is the well-known Weber problem which was initially formulated as the minimization of a weighted sum of Euclidean distances ([12], [15], [19]) and later generalized to other L_p-distances ([22], [23]). Also a special instance of (P) is the stochastic queue location problem ([13], [33]) where the demand points generate clients according to independent Poisson processes. In this case the random arrival process of the clients and the corresponding congestion effects caused by the nonavailability of the server are incorporated into the objective function.

Observe, since any L_p-norm, $1 < p < \infty$ is everywhere, except in $\mathbf{0}$, differentiable, that the function f is differentiable in $\mathbb{R}^n \backslash D$ with D denoting the finite set of demand points. Hence our optimization problem is nonsmooth and so we might apply general solution techniques from nonsmooth optimization as discussed by Lemaréchal in [26]. However, the objective function f is only nonsmooth in a finite set of points and due to this weak form of nonsmoothness and the computational power needed for algorithms in nonsmooth optimization we would like to apply classical optimization techniques by following the next two approaches. The first approach is to approximate f by a smooth and uniform perturbation of f and to solve this perturbed problem by a generalization of the well-known Weiszfeld algorithm. Under the condition that none of the demand points are optimal and hence an optimal solution must be a differentiable point of f, this algorithm was first proposed by Weiszfeld [32] to solve the nonsmooth Weber problem. Basically, after writing down the nonlinear equation system resulting from setting the first-order partial derivatives of the Weber function equal to zero, the Weiszfeld algorithm tries to solve this system iteratively. By this observation it should be clear how to generalize this approach to instances of (P).

Another approach is to consider the original problem and to derive a fast procedure for checking whether a demand point is a local optimum. This procedure can then be applied to all the demand points and if one suspects that there exists a differentiable local optimum point one starts a generalization of the Weiszfeld procedure in a suitable chosen starting point. In Sections 2 and 3 these different approaches will be considered. Using general results for quasiconvex functions new results are presented for resp. $p = 2$ in the nonperturbed case and for p belonging to $(1, 2]$ in the perturbed case. Moreover, in Section 4 we also discuss rate of convergence results. Finally, in Section 5 some computational experiments are presented for $n = 2$ while Section 6 concludes with a summary of the results.

To conclude this section we observe that there are also more sophisticated methods available to solve the above problems. Some of these methods can even be applied to single facility location models with arbitrary norms or even gauges (see [28]) and can be seen as specializations of procedures to solve general convex\quasiconvex optimization models. However, if one is only interested in obtaining quickly rough estimates of the solution in case L_p-norms are used to measure the distance ($1 < p \leq 2$) it is preferable to apply the generalized Weiszfeld method. As such this method is easy to implement and suitable for "quick" engineering purposes.

2. A Generalization of the Weiszfeld Algorithm

Due to the differentiability of the function $g: S \to \mathbb{R}$ with S open and $\mathbb{R}_+^m \subseteq S$ it is easy to verify that the directional derivative

$$f'(x; y) := \lim_{t \downarrow 0} \frac{f(x+ty) - f(x)}{t}$$

exists for every $x, y \in \mathbb{R}^n$ and so one can introduce the set Γ_f given by

$$\Gamma_f = \{x \in \mathbb{R}^n : f'(x; y) \geq 0 \text{ for every } y \in \mathbb{R}^n\}$$

Clearly any solution x^* of the optimization problem (P) must belong to Γ_f.

Even for smooth instances of any unconstrained nonlinear optimization problem it is known that every algorithm constructing search directions using only local information cannot guarantee in advance to find an optimal point. The best one can hope for is the identification of a stationary point. This implies for the optimization problem (P) that any algorithm based on local information can only help to determine an element of the set Γ_f. Fortunately we do not need a complicated algorithm to decide whether a demand point belongs to Γ_f. To show this introduce for notational convenience and $1 < p < \infty$ fixed the functions $c_0 : D^c \to \mathbb{R}^n$ and $c_\ell : (D\backslash\{\mathbf{a}_\ell\})^c \to \mathbb{R}^n$, $1 \leq \ell \leq m$, given by

$$c_0(x) := \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(x))\ \nabla\ (\|x - \mathbf{a}_j\|_p) \tag{2}$$

and

$$c_\ell(x) := \sum_{\substack{j=1 \\ j\neq\ell}}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(x))\nabla(\|x - \mathbf{a}_j\|_p) \tag{3}$$

Observe S^c denotes the complement of an arbitrary set $S \subseteq \mathbb{R}^n$ and $\nabla(\|x - \mathbf{a}_j\|_p)$ the gradient vector of the function $x \to \|x - \mathbf{a}_j\|_p$ evaluated in $x \in \{\mathbf{a}_j\}^c$. Using the above notation it is not difficult to verify that

$$f'(x;\, y) = \begin{cases} c_0(x)^T y & \text{if } x \in D^c \\ c_\ell(x)^T y + \frac{\partial g}{\partial z_\ell}(\mathbf{d}(x))\ \|y\|_p & \text{if } x = \mathbf{a}_\ell,\ \ell = 1,\dots,m \end{cases} \tag{4}$$

Moreover, if $\delta : \mathbb{R} \to \mathbb{R}$ denotes the so-called Kronecker symbol, i.e.

$$\delta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

it is easy to prove the next result.

Lemma 2.1

If $1 < p < \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$ then $\mathbf{a}_\ell$ belongs to Γ_f if and only if $\|c_\ell(\mathbf{a}_\ell)\|_q \leq \frac{\partial g}{\partial z_\ell}\ (\mathbf{d}(\mathbf{a}_\ell))$.

Furthermore, if $\mathbf{a}_\ell$ is not an element of Γ_f, it follows that the vector

$$y_0^T := (-\delta(b_1)\,|b_1|^{q/p},\dots,-\delta(b_n)\,|b_n|^{q/p})$$

with $\mathbf{b}^T = (b_1,\dots,b_n) = c_\ell(\mathbf{a}_\ell)^T$ is a descent direction.

Proof

If $\|c_\ell(\mathbf{a}_\ell)\|_q \leq \frac{\partial g}{\partial z_\ell}\ (\mathbf{d}(\mathbf{a}_\ell))$ we obtain by Hölder's inequality (see [29]) that

$$\begin{aligned} c_\ell(\mathbf{a}_\ell)^T y &\geq -\|c_\ell(\mathbf{a}_\ell)\|_q\ \|y\|_p \\ &\geq -\frac{\partial g}{\partial z_\ell}\ (\mathbf{d}(\mathbf{a}_\ell))\ \|y\|_p \end{aligned}$$

and hence by (4) it follows that $f'(\mathbf{a}_\ell;\, y) \geq 0$ for every $y \in \mathbb{R}^n$. On the other hand, using the definition of y_0, it is easy to verify that

$$c_\ell(\mathbf{a}_\ell)^T y_0 + \frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))\ \|y_0\|_p = \|y_0\|_p\left(\frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell)) - \|c_\ell(\mathbf{a}_\ell)\|_q\right) \tag{5}$$

and so if $\mathbf{a}_\ell \in \Gamma_f$ it must follow by (4) and (5) that either $\|y_0\|_p = 0$ or $\|c_\ell(\mathbf{a}_\ell)\|_q \leq \frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))$. By the definition of y_0 this finally yields $\|c_\ell(\mathbf{a}_\ell)\|_q \leq \frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))$. To verify the last part of the result we obtain by the first part that $\mathbf{a}_\ell$ does not belong to Γ_f if and only if $\|c_\ell(\mathbf{a}_\ell)\|_q > \frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))$. Hence by (4) and (5) it follows that $f'(\mathbf{a}_\ell; y_0) < 0$. ◆

To prove that Γ_f only contains optimal solutions of (P) we need to introduce the following class of functions (see [2]). Observe unless stated otherwise that we always assume $\mathbb{R}^m_+ \subseteq S$ and S open.

Definition 2.2

A function $h : S \rightarrow \mathbb{R}$ is called quasiconvex on $\mathbb{R}^m_+$ if $h(\lambda x+(1-\lambda)y) \leq \max\{h(x), h(y)\}$ for every $x,y \in \mathbb{R}^m_+$ and $0 \leq \lambda \leq 1$.

A function $h : S \rightarrow \mathbb{R}$ is called pseudoconvex on $\mathbb{R}^m_+$ if h is differentiable and $\nabla h(x)^T(y-x) \geq 0$ implies $h(y) \geq h(x)$ for every $x,y \in \mathbb{R}^m_+$.

Finally, the function h is called quasiconcave (pseudoconcave) on $\mathbb{R}^m_+$ if -h is quasiconvex (pseudoconvex) on $\mathbb{R}^m_+$.

It is not difficult to verify (see [2]) that the class of pseudoconvex functions is a proper subset of the class of quasiconvex functions.

In [21] the next result is given. For the sake of completeness we list its proof.

Lemma 2.3

If $g : S \rightarrow \mathbb{R}$ is pseudoconvex on $\mathbb{R}^m_+$ then $\mathbf{x}$ is a solution of (P) if and only if $\mathbf{x}$ belongs to Γ_f.

Proof

If $\mathbf{x}$ is a solution of (P) then clearly $\mathbf{x}$ belongs to Γ_f and so we only have to verify the reverse implication. For this case we only give a proof if $\mathbf{x}$ belongs to $D \cap \Gamma_f$ since the proof for $\mathbf{x}$ belonging to $D^c \cap \Gamma_f$ is similar. Let $\mathbf{x} = \mathbf{a}_\ell$ and consider the function $d_j : y \rightarrow \|y - \mathbf{a}_j\|_p$, $j \neq \ell$. Since d_j is convex it follows by the subgradient inequality that

$$y^T \nabla d_j(\mathbf{a}_\ell) \leq d_j(y + \mathbf{a}_\ell) - d_j(\mathbf{a}_\ell)$$

for every $y \in \mathbb{R}^n$ and $j \neq \ell$. Hence, using $\frac{\partial g}{\partial z_j}(\mathbf{d}(\mathbf{a}_\ell)) \geq 0$, we obtain by (3) that

$$c_\ell(\mathbf{a}_\ell)^T y \leq \sum_{\substack{j=1 \\ j\neq \ell}}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(\mathbf{a}_\ell))\,(d_j(y + \mathbf{a}_\ell) - d_j(\mathbf{a}_\ell))$$

By (4), the above inequality and $\mathbf{a}_\ell \in \Gamma_f$ this yields

$$0 \leq f'(\mathbf{a}_\ell; y) = c_\ell(\mathbf{a}_\ell)^T y + \frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))\,\|y\|_p$$

Furthermore, since

$$\begin{aligned}\|T_c(\mathbf{x}) - \mathbf{a}_j\|_2^2 &= \|T_c(\mathbf{x}) - \mathbf{x} + \mathbf{x} - \mathbf{a}_j\|_2^2 \\ &= \|T_c(\mathbf{x}) - \mathbf{x}\|_2^2 + \|\mathbf{x} - \mathbf{a}_j\|_2^2 + 2(T_c(\mathbf{x}) - \mathbf{x})^T (\mathbf{x} - \mathbf{a}_j)\end{aligned}$$

we obtain by (13) that

$$\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(\mathbf{x}))\, \|\mathbf{x} - \mathbf{a}_j\|_2^{-1} \left[\|T_c(\mathbf{x}) - \mathbf{a}_j\|_2^2 - \|\mathbf{x} - \mathbf{a}_j\|_2^2\right] < 0 \qquad (14)$$

We can now prove that the generalized Weiszfeld procedure is a descent algorithm for $p = 2$. This generalizes a result by Kuhn [15] and Ostresh [27]. Remember for a proper interpretation of the algorithm it is always assumed that $0 < b(\mathbf{x}) < \infty$ for every $\mathbf{x} \in \text{co}(D)\backslash D$ or equivalently $\nabla g(\mathbf{d}(\mathbf{x}))$ contains at least one positive component for every $\mathbf{x} \in \text{co}(D)\backslash D$.

Theorem 2.4

Let $g : S \rightarrow \mathbb{R}$ be a nondecreasing continuously differentiable function and suppose $\varphi : \mathbb{R}_+^m \rightarrow \mathbb{R}$ is given by

$$\varphi(\mathbf{z}) = g\left(z_1^{1/2}, \ldots, z_n^{1/2}\right) \qquad (15)$$

If φ is quasiconcave on $\mathbb{R}_+^m$ and $\mathbf{x}$ belongs to $\text{co}(D)\backslash(D \cup \Gamma_f)$ with $\nabla g(\mathbf{d}(\mathbf{x}))$ containing at least one positive component then for f given by (1) it follows that $f(T_c(\mathbf{x})) < f(\mathbf{x})$ for every $0 < c \leq 1$.

Proof

Since $g : S \rightarrow \mathbb{R}$ is continuously differentiable it follows immediately that φ is continuously differentiable on $\text{int}(\mathbb{R}_+^m)$. Moreover, if $\mathbf{z} \in \text{int}(\mathbb{R}_+^m)$ we obtain that

$$\frac{\partial \varphi}{\partial z_j}(\mathbf{z}) = \frac{1}{2} \frac{\partial g}{\partial z_j}\left(z_1^{1/2}, \ldots, z_n^{1/2}\right) z_j^{-1/2}$$

for every $1 \leq j \leq m$. This implies since $\mathbf{d}(\mathbf{x})$ belongs to $\text{int}(\mathbb{R}_+^m)$ for every $\mathbf{x} \in \text{co}(D)\backslash D$ that the inequality (14) can be simplified to

$$\sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j}(\mathbf{d}^2(\mathbf{x})) \left[\|T_c(\mathbf{x}) - \mathbf{a}_j\|_2^2 - \|\mathbf{x} - \mathbf{a}_j\|_2^2\right] < 0$$

for every $\mathbf{x} \in \text{co}(D)\backslash(D \cup \Gamma_f)$. Hence by the quasiconcavity of φ and Theorem 3.5.4 of [2] we obtain that $\varphi\left(\mathbf{d}^2(T_c(\mathbf{x}))\right) < \varphi(\mathbf{d}^2(\mathbf{x}))$ and so the desired result follows. ◆

For the proof of convergence of the generalized Weiszfeld procedure for $p = 2$ we also need the following result.

Lemma 2.5

If $g : S \rightarrow \mathbb{R}$ is a nondecreasing continuously differentiable function with $\nabla g(\mathbf{d}(\mathbf{x}))$ containing at least one positive component for every $\mathbf{x} \in \text{co}(D)$ then the mapping $T_c : \text{co}(D) \rightarrow \text{co}(D)$ is continuous on $\text{co}(D)$ for every $0 < c \leq 1$. Moreover, for every sequence $\{\mathbf{y}_k : k \geq 0\} \subseteq \text{co}(D)\backslash D$ such that $\lim_{k \to \infty} \mathbf{y}_k = \mathbf{a}_\ell \in D$ and $\|\mathbf{c}_\ell(\mathbf{a}_\ell)\|_2 > 0$ we have for $p = 2$

$$\lim_{k\to\infty} \frac{\|T_1(y_k)-a_\ell\|_2}{\|y_k-a_\ell\|_2} = \frac{\|c_\ell(a_\ell)\|_2}{\frac{\partial g}{\partial z_\ell}(d(a_\ell))} \leq \infty$$

with $c_\ell(x)$ given by (3).

Proof

To prove that the mapping T_c is continuous on co(D) it is sufficient by (11) to verify that T_1 is continuous on co(D). As observed after (10) the mapping $T_1 : \text{co(D)} \to \text{co(D)}$ is indeed continuous and so the first part of the lemma is proved. To verify the last part of this lemma consider now a sequence $\{y_k : k \geq 0\} \subseteq \text{co(D)}\backslash\text{D}$ with $\lim_{k\to\infty} y_k = a_\ell \in D$ and $\|c_\ell(a_\ell)\|_2 > 0$. By (10) and (12) we obtain that

$$T_1(y_k) = y_k - b(y_k)^{-1}\,\nabla f(y_k)$$

$$= y_k - \frac{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}\,(y_k-a_j)}{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}}$$

$$= \frac{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}\,a_j}{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}}$$

$$= a_\ell + \frac{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}\,(a_j-a_\ell)}{\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}}$$

Hence

$$\frac{T_1(y_k)-a_\ell}{\|y_k-a_\ell\|_2} = \frac{-\sum_{\substack{j=1\\ j\neq\ell}}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}\,(a_\ell-a_j)}{\frac{\partial g}{\partial z_\ell}(d(y_k)) + \sum_{\substack{j=1\\ j\neq\ell}}^{m} \frac{\partial g}{\partial z_j}(d(y_k))\,\|y_k-a_j\|_2^{-1}\,\|y_k-a_\ell\|_2}$$

and this implies using $\|c_\ell(a_\ell)\|_2 > 0$ and the continuity of the functions $x \to \frac{\partial g}{\partial z_\ell}(d(x))$ and $y \to \|y\|_2$ that

$$\lim_{k\to\infty} \frac{\|T_1(y_k)-a_\ell\|_2}{\|y_k-a_\ell\|_2} = \frac{\|c_\ell(a_\ell)\|_2}{\frac{\partial g}{\partial z_\ell}(d(a_\ell))} \qquad \blacklozenge$$

Observe the condition $\|c_\ell(a_\ell)\|_2 > 0$ is needed to give a proper interpretation of the above limit $\beta := \dfrac{\|c_\ell(a_\ell)\|_2}{\frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))}$, i.e. this limit is finite if $\frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell)) > 0$ and equals $+\infty$ if $\frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell)) = 0$.

We prove now the most important result of this section.

Theorem 2.6

If $g : S \rightarrow \mathbb{R}$ is a nondecreasing continuously differentiable function with $\nabla g(\mathbf{d}(x))$ containing at least one positive component for every $x \in \mathrm{co(D)}$, $\varphi : \mathbb{R}^m_+ \rightarrow \mathbb{R}$ as given by (15) is quasiconcave on $\mathbb{R}^m_+$ and the sequence $\{x^{(k)} : k \geq 0\}$ generated by the generalized Weiszfeld method is contained in the set co(D)\D then this sequence has a limit point belonging to co(D) and every limit point of this sequence belongs to Γ_f.

Proof

If $x^{(k)} \in \Gamma_f \cap (\mathrm{co(D)\backslash D})$ for some $k \geq 0$ then $f'(x^{(k)}; y) = \nabla f(x^{(k)})^T y$ for every $y \in \mathbb{R}^n$ and hence $\nabla f(x^{(k)}) = \mathbf{0}$. By this observation and Theorem 2.4 we conclude that the sequence $\{f(x^{(k)}) : k \geq 0\}$ is nonincreasing. Moreover, since the continuous function f is bounded from below on the compact set co(D) this implies that the sequence $\{f(x^{(k)}) : k \geq 0\}$ is also convergent and hence

$$\lim_{k\rightarrow\infty} \left[f\left(T_1(x^{(k)})\right) - f(x^{(k)})\right] = 0 \qquad (16)$$

Observe now, using $\{x^{(k)} : k \geq 0\} \subseteq \mathrm{co(D)}$, that by Theorem 2.37 of [30] the sequece $\{x^{(k)} : k \geq 0\}$ has a limit point x^* in co(D) and so there exists by definition a subsequence $\{x^{(k)} : k \in K\} \subseteq \{x^{(k)} : k \geq 0\}$ such that $\lim_{k\in K\rightarrow\infty} x^{(k)} = x^*$. By the continuity of the mapping T_1 as shown in Lemma 2.5 and the continuity of f this yields by (16) that

$$f(T_1(x^*)) - f(x^*) = \lim_{k\in K\rightarrow\infty} [f(T_1(x^{(k)})) - f(x^{(k)})] = 0 \qquad (17)$$

On the other hand, if x^* does not belong to D we obtain by Theorem 2.4 that $f(T_1(x^*)) < f(x^*)$ whenever x^* also does not belong to Γ_f. This contradicts (17) and so $x^* \in \Gamma_f$ if $x^* \notin D$. Let us now consider the case that x^* is a demand point $\mathbf{a}_\ell$ and $\mathbf{a}_\ell \notin \Gamma_f$. By Lemma 2.1 we have $\|c_\ell(a_\ell)\|_2 > \frac{\partial g}{\partial z_\ell}(\mathbf{d}(a_\ell)) \geq 0$ or

$$\infty \geq \beta := \frac{\|c_\ell(a_\ell)\|_2}{\frac{\partial g}{\partial z_\ell}(\mathbf{d}(\mathbf{a}_\ell))} > 1$$

and so by Lemma 2.5 there exists some $\delta > 0$ such that

$$\frac{\|T_1(x) - \mathbf{a}_\ell\|_2}{\|x - \mathbf{a}_\ell\|_2} > 1 \qquad (18)$$

for every $x \in B(\mathbf{a}_\ell; \delta) \cap \mathrm{co(D)\backslash D}$ with $B(\mathbf{a}_\ell; \delta) := \{x \in \mathbb{R}^n : \|x - \mathbf{a}_\ell\|_2 < \delta\}$. Let $K' = \{k_1, k_2, \ldots, k_m\}$ denote a subset of K satisfying

$$k_m := \inf\{k\in K : \|x^{(k)} - \mathbf{a}_\ell\|_2 \le \frac{1}{m}\} \tag{19}$$

Clearly $k_1 \le k_2 \le \ldots \le k_m \le k_{m+1}$ and $\lim_{m\to\infty} k_m = +\infty$. Moreover, if $\|x^{(k_m-1)} - \mathbf{a}_\ell\|_2 < \delta$ it must follow by (18) that $\|x^{(k_m)} - \mathbf{a}_\ell\|_2 = \|T_1(x^{(k_m-1)}) - \mathbf{a}_\ell\|_2 > \|x^{(k_m-1)} - \mathbf{a}_\ell\|_2$ and hence also $\|x^{(k_m-1)} - \mathbf{a}_\ell\|_2 \le \frac{1}{m}$ contradicting (19). As a result of this,

$$\|x^{(k_m-1)} - \mathbf{a}_\ell\|_2 > \delta, \ \forall m \ge 1 \tag{20}$$

and since $\{x^{(k_m-1)} : m \ge 1\}$ is a bounded sequence there must exist by Theorem 2.37 of [30] a convergent subsequence $\{x^{(k-1)} : k\in K''\}$, $K'' \subseteq K'$ for which $\lim_{k\in K''\to\infty} x^{(k-1)} = y$ for some $y\in co(D)$. Clearly by (20) we obtain that $\|y - \mathbf{a}_\ell\|_2 > \delta$ and by the continuity of the mapping T_1 it follows that

$$\mathbf{a}_\ell = \lim_{k\in K''\to\infty} x^{(k)} = \lim_{k\in K''\to\infty} T_1(x^{(k-1)}) = T_1(y) \tag{21}$$

Observe, since $\|y - \mathbf{a}_\ell\|_2 > \delta$ and $\mathbf{a}_\ell = T_1(y)$, that y does not belong to D. Hence y must be a differentiability point of the function f and so $\nabla f(y) = \mathbf{0}$ if y belongs to Γ_f. This implies $y = T_1(y) = \mathbf{a}_\ell$ and we have a contradiction with $\|y - \mathbf{a}_\ell\|_2 > \delta$. The above observation yields that y belongs to $co(D)\backslash(D\cup\Gamma_f)$ and so by Theorem 2.4 it follows that

$$f(\mathbf{a}_\ell) = f(T_1(y)) < f(y) \tag{22}$$

On the other hand we obtain by (16), (21) and the continuity of f and T_1 that

$$\begin{aligned} 0 &= \lim_{k\in K''\to\infty} [f(T_1(x^{(k-1)})) - f(x^{(k-1)})] = f(T_1(y)) - f(y) \\ &= f(\mathbf{a}_\ell) - f(y) \end{aligned}$$

The last result contradicts (22) and so $\mathbf{a}_\ell$ must belong to Γ_f. ♦

The result mentioned in Theorem 2.6 does not imply that every sequence $\{x^{(k)} : k \ge 0\} \subseteq co(D)\backslash D$ generated by the generalized Weiszfeld algorithm converges to a point belonging to Γ_f. In order to achieve convergence we need the following result.

Theorem 2.7

If Γ_f is a finite set and the conditions of Theorem 2.6 are satisfied then the sequence $\{x^{(k)} : k \ge 0\} \subseteq co(D)\backslash D$ converges to an element of Γ_f.

Proof

Let C denote the set of limit points of the sequence $\{x^{(k)} : k \ge 0\} \subseteq co(D)\backslash D$. By Theorem 2.6 this set is nonempty and $C \subseteq \Gamma_f$. Since Γ_f is finite this implies that C is finite, i.e. $|C| < \infty$ with $|.|$ denoting the cardinality of a set. To verify the above result it is sufficient by Theorem 2.37 of [30] to check that $|C| = 1$. Suppose therefore that $|C| \ge 2$ and choose some $\mathbf{c}_i \in C$. Since $|C| < \infty$, there exists some $\delta > 0$ such that $B(\mathbf{c}_i; \delta)\cap C = \{\mathbf{c}_i\}$. By the definition of C, $|C| \ge 2$ and $\mathbb{N}\backslash K$ and K countable with $K = \{k \ge 0 : \|x^{(k)} - \mathbf{c}_i\|_2 < \delta\}$, there exists again by Theorem 2.37 of [30] a subsequence $K' \subseteq K$ satisfying $\lim_{k\in K'\to\infty} x^{(k)} = \mathbf{c}_i$ and $\lim_{k\in K'\to\infty} x^{(k+1)} \ne \mathbf{c}_i$. Hence by the continuity of T_1 this yields $T_1(\mathbf{c}_i) \ne \mathbf{c}_i$ and so $\mathbf{c}_i$ belongs to $co(D)\backslash(D\cup\Gamma_f)$.

However, by Theorem 2.6 we obtain that $c_j \in C \subseteq \Gamma_f$ and a contradiction is derived. Hence $|C| = 1$ and the desired result is proved. ♦

In general one can verify for p = 2 whether the Weiszfeld algorithm must be started and if so, whether the sequence $\{x^{(k)} : k \geq 0\}$ is contained in the set co(D)\D. This can be done as follows. First one checks, using Lemma 2.1, whether there exists a demand point belonging to Γ_f. If not, start the Weiszfeld algorithm in the point generated by the next procedure. First, pick that demand point $\mathbf{a}_\ell$ with lowest function value and use Armijo's rule (see [6]) or line minimization in the descent direction given by Lemma 2.1. This yields a new point $\mathbf{x}^{(1)}$ and start now the Weiszfeld algorithm in this point. Clearly $f(\mathbf{x}^{(1)}) < f(\mathbf{a}_\ell) = \min_{1 \leq j \leq m} f(\mathbf{a}_j)$ and by Theorem 2.4 it is now easy to verify that $\{x^{(k)} : k \geq 0\} \subseteq \text{co}(D) \setminus D$.

In order to prove that Γ_f is a finite set we introduce the next definition.

Definition 2.8

The set D of demand points is called not collinear if this set contains three demand points which are not lying on one straight line. Moreover, the set D is called strongly not collinear if $D\setminus\{\mathbf{a}_j\}$ is not collinear for every j = 1,..., m.

Theorem 2.9

If the set D of demand points is not collinear and the nondecreasing continuously differentiable function $g : S \to \mathbb{R}$ is quasiconvex on $\mathbb{R}^m_+$ and satisfies $\nabla g(\mathbf{d}(x)) \in \text{int}(\mathbb{R}^m_+)$ for every x belonging to co(D)\D then Γ_f is a finite set. Moreover, if additionally D is strongly not collinear and $\nabla g(\mathbf{d}(x)) \in \text{int}(\mathbb{R}^m_+)$ for every $x \in \text{co}(D)$ then Γ_f only contains the unique optimal solution of (P).

Proof

To verify the first result it is sufficient to show that $\Gamma_f \cap D^c$ is finite. Suppose therefore that $\overline{\mathbf{x}}$ belongs to $\Gamma_f \cap D^c$. If D is not collinear it is shown in [9] and [11] that the Weber function $\mathbf{x} \to \sum_{j=1}^{m} d_j(\mathbf{x})$ with $d_j(\mathbf{x}) := \|\mathbf{x} - \mathbf{a}_j\|_2$ is strictly convex. This implies by Theorem 3.3.3 of [2] that the subgradient inequality

$$d_j(\overline{\mathbf{x}} + \mathbf{y}) - d_j(\overline{\mathbf{x}}) \geq \mathbf{y}^T \nabla d_j(\overline{\mathbf{x}}),\ j = 1,\dots, m$$

holds for every $\mathbf{y} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ with a strict inequality for at least one $j \in \{1,\dots, m\}$. Using now $\nabla g(\mathbf{d}(\mathbf{x})) > \mathbf{0}$ for every $\mathbf{x} \in \text{co}(D)\setminus D$ we obtain

$$\begin{aligned} 0 \leq f'(\overline{\mathbf{x}}; \mathbf{y}) &= \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(\overline{\mathbf{x}}))\, \mathbf{y}^T \nabla d_j(\overline{\mathbf{x}}) \\ &< \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(\overline{\mathbf{x}}))\, (d_j(\overline{\mathbf{x}} + \mathbf{y}) - d_j(\overline{\mathbf{x}})) \end{aligned}$$

This implies by the quasiconvexity of g and Theorem 3.5.4 of [2] that $f(\overline{x} + y) > f(\overline{x})$ for every $y \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ and so $\overline{x}$ is a unique optimal solution of (P) if $\overline{x}$ belongs to $\Gamma_f \cap D^c$. Hence $|\Gamma_f \cap D^c| \leq 1$ and the first part of this theorem is proved. To prove the second part we observe, if D is strongly not collinear, that by a similar proof one can show that $\overline{x}$ is a unique optimal solution of (P) if $\overline{x}$ belongs to $\Gamma_f \cap D$. This implies using also the first part that $\overline{x}$ is a unique optimal solution if $\overline{x} \in \Gamma_f$ and so $|\Gamma_f| = 1$. ◆

Assuming $\varphi : \mathbb{R}^m_+ \to \mathbb{R}$ given by (15) and $g : S \to \mathbb{R}$ are respectively quasiconcave and quasiconvex functions on $\mathbb{R}^m_+$, it is now immediate by Theorems 2.9 and 2.7 that the sequence $\{x^{(k)} : k \geq 0\} \subseteq co(D) \backslash D$ generated by the generalized Weiszfeld method converges to the unique optimal point x^* if some additional conditions on D and g are also satisfied. It is also possible to obtain convergence of the Weiszfeld method in a neighbourhood of x^* for a class of objective functions for which the corresponding functions φ are not quasiconcave (see [21]).

To conclude this section we observe that the domain of the function g given by (1) was taken for convenience to be an open set containing $\mathbb{R}^m_+$. This enabled us to define the function φ given by (15) on the convex set $\mathbb{R}^m_+$. However, if the function g represents the ratio of two functions this function might only be defined on a (convex) subset of $\mathbb{R}^m_+$. An example of such a function is discussed in Section 5. Although these functions do not fit within the framework discussed in this section it does not mean that the results of this section can not be applied to those functions. By checking the proofs of the convergence results it turns out that the same results also hold if the domain S of the function g is a convex subset of $\mathbb{R}^m_+$ satisfying

i) $\mathbf{d}(x) \in int(S)$ for every $x \in co(D) \backslash D$

ii) $S^{1/2} \subseteq S$ with $S^{1/2} := \{(z_1^{1/2}, \ldots, z_n^{1/2}) : z \in S\}$

iii) The function $g : S \to \mathbb{R}$ is continuously differentiable on int(S) and the vector-valued function $z \to \nabla g(z)$ can be continuously extended to the boundary of S.

As a consequence of these conditions we may define the function φ on the same domain S as the function g. Observe some of the test problems to be discussed in Section 5 fall within the above category.

3. The Weiszfeld Algorithm and the Hyperbolic Approximation

As observed in the previous section, the nondifferentiability of the objective function (1) at each demand point is caused by the nondifferentiability of the L_p-norm, $1 < p < \infty$ in $\mathbf{0}$. A simple approach, suggested by Eyster et al. [10], to avoid this problem is to replace the L_p-norm $\|v\|_p$ of the vector $v^T = (v_1, \ldots, v_n)$ by the so-called hyperbolic approximation $\|\xi^\varepsilon(v)\|_p$, $\varepsilon > 0$ fixed with $\xi^\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$\xi^\varepsilon(v) := ((v_1^2 + \varepsilon^2)^{1/2}, \ldots, (v_n^2 + \varepsilon^2)^{1/2}) \tag{23}$$

The original optimization problem (P) is then uniformly approximated by

$$\min\{f_\varepsilon(x) : x \in \mathbb{R}^n\} \tag{P_ε}$$

with $f_\varepsilon(x) := g(\mathbf{d}_\varepsilon(x))$ and $\mathbf{d}_\varepsilon(x)^T := (\|\xi^\varepsilon(x - \mathbf{a}_1)\|_p, \ldots, \|\xi^\varepsilon(x - \mathbf{a}_m)\|_p)$.

Since it is not difficult to compute an upperbound on the error caused by replacing (P) by (P_ε) (see [14]), the optimal solution of the perturbed problem can be used as an approximation of the optimal solution of the original problem. Moreover, the smoothing procedure has the advantage that the objective function is everywhere differentiable and so the perturbed optimization problem (P_ε) can be solved by classical nonlinear optimization algorithms (see [2]). Based on the necessary condition $\nabla f_\varepsilon(\mathbf{x}_\varepsilon^*) = \mathbf{0}$ for $\mathbf{x}_\varepsilon^*$ an optimal solution of (P_ε) one can construct an iterative procedure similarly as carried out in the previous section for the unperturbed problem. To construct this procedure observe for $p > 1$ fixed that the i-th component $\nabla_i f_\varepsilon(\mathbf{x})$ of the gradient vector $\nabla f_\varepsilon(\mathbf{x})$ equals

$$\nabla_i f_\varepsilon(\mathbf{x}) = \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(\mathbf{x}))\ \|\xi^\varepsilon(\mathbf{x} - \mathbf{a}_j)\|_p^{1-p}\ \xi_i^\varepsilon(\mathbf{x} - \mathbf{a}_j)^{p-2}\ (x_i - a_{ij}) \qquad (24)$$

where $\xi_i^\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1,\dots, n$, is given by $\xi_i^\varepsilon(\mathbf{v}) := (v_i^2 + \varepsilon^2)^{1/2}$ and so

$$0 = \nabla_i f_\varepsilon(\mathbf{x}_\varepsilon^*) = \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(\mathbf{x}_\varepsilon^*))\ \|\xi^\varepsilon(\mathbf{x}_\varepsilon^* - \mathbf{a}_j)\|_p^{1-p}\ \xi_i^\varepsilon(\mathbf{x}_\varepsilon^* - \mathbf{a}_j)^{p-2}\ (x_{i\varepsilon}^* - a_{ij}) \qquad (25)$$

Isolating the components of $\mathbf{x}_\varepsilon^*$ in (25) yields for $1 \le i \le n$ that

$$x_{i\varepsilon}^* = \sum_{j=1}^{m} \bar{\lambda}_{ij}(\mathbf{x}_\varepsilon^*) a_{ij}$$

with

$$\bar{\lambda}_{ij}(\mathbf{x}) := \frac{\frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(\mathbf{x}))\ \|\xi^\varepsilon(\mathbf{x}-\mathbf{a}_j)\|_p^{1-p}\ \xi_i^\varepsilon(\mathbf{x}-\mathbf{a}_j)^{p-2}}{\sum_{\ell=1}^{m} \frac{\partial g}{\partial z_\ell}(\mathbf{d}_\varepsilon(\mathbf{x}))\ \|\xi^\varepsilon(\mathbf{x}-\mathbf{a}_\ell)\|_p^{1-p}\ \xi_i^\varepsilon(\mathbf{x}-\mathbf{a}_\ell)^{p-2}} \qquad (26)$$

Clearly for a proper interpretation of the function $\bar{\lambda}_{ij} : \mathbb{R}^n \to \mathbb{R}$ we assume that $\nabla g(\mathbf{d}_\varepsilon(\mathbf{x}))$ has at least one strictly positive component for every $\mathbf{x}$ belonging to a prespecified compact subset of $\mathbb{R}^n$.

A natural iterative approach to determine $\mathbf{x}_\varepsilon^*$ is now given by

$$x_i^{(k+1)} = \sum_{j=1}^{m} \bar{\lambda}_{ij}(\mathbf{x}^{(k)}) a_{ij},\ 1 \le i \le n,\ k \ge 0 \qquad (27)$$

Expressing (27) in matrix notation in a similar way as reported in [22] and [23] for the special case of the Weber function and defining the sets

$$K := \prod_{i=1}^{n} \left[\min_{1\le j\le m} a_{ij},\ \max_{1\le j\le m} a_{ij}\right]$$

and

$$U := \begin{cases} K & \text{if } p > 1 \text{ and } p \neq 2 \\ \mathrm{co}(D) & \text{if } p = 2 \end{cases} \qquad (28)$$

yields the following algorithm:

Adapted Weiszfeld Method

Step 0 Choose $x^{(0)} \in U$ arbitrarily and set $k \leftarrow 0$.

Step 1 Compute $x^{(k+1)} = M_1(x^{(k)})$ with $M_1 : U \to U$ a mapping given by

$$M_1(x) := x - B(x)^{-1}\nabla f_\varepsilon(x) \tag{29}$$

and

$B(x)$ a $n \times n$ diagonal matrix with positive diagonal elements $b_i(x)$, $i = 1,\dots, n$, equal to

$$b_i(x) := \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(x))\, \|\xi^\varepsilon(x-a_j)\|_p^{1-p}\, \xi_i^\varepsilon(x-a_j)^{p-2} \tag{30}$$

Step 2 Stop if a given stopping criterion is satisfied, otherwise set $k \leftarrow k + 1$ and return to Step 1.

Observe that M_1 is a special case of the set of mappings $M_c : U \to M_c(U)$ given by

$$M_c(x) := x - cB(x)^{-1}\nabla f_\varepsilon(x) \tag{31}$$

with $0 < c \leq 1$ some fixed parameter. Moreover, since $M_c(x) = cM_1(x) + (1 - c)x$ and U is convex this implies that $M_c(U) \subseteq U$.

While for the generalized Weiszfeld algorithm we can only guarantee global convergence under certain conditions for $p = 2$, a similar result holds for the above algorithm whenever $1 < p \leq 2$. Denoting by $M_{ci}(x)$ the i-th component of M_c, it follows by (31) that

$$\begin{aligned}(M_{ci}(x) - x_i)^2\, b_i(x) &+ 2(M_{ci}(x) - x_i)\nabla_i f_\varepsilon(x)\\ &= -c(2 - c)\, b_i(x)^{-1}\nabla_i f_\varepsilon(x)^2 \leq 0\end{aligned}$$

for every $0 < c \leq 1$ and $i = 1, 2,\dots, n$. Hence by (24) and (30) we obtain that

$$\begin{aligned}0 &\geq (M_{ci}(x) - x_i)^2\, b_i(x) + 2(M_{ci}(x) - x_i)\nabla_i f_\varepsilon(x)\\ &= \sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(x))\, \|\xi^\varepsilon(x-a_j)\|_p^{1-p}\, \xi_i^\varepsilon(x-a_j)^{p-2} \big((M_{ci}(x) - x_i)^2 +\\ &\qquad 2(M_{ci}(x) - x_i)\,(x_i - a_{ij})\big)\end{aligned} \tag{32}$$

for every $i = 1,\dots, n$.

Observe now

$$(M_{ci}(x) - x_i)^2 + 2(M_{ci}(x) - x_i)\,(x_i - a_{ij}) = (M_{ci}(x) - a_{ij})^2 - (x_i - a_{ij})^2$$

and by the definition of $\xi_i^\varepsilon : \mathbb{R}^n \to \mathbb{R}$ given below (24) this implies

$$(M_{ci}(x) - x_i)^2 + 2(M_{ci}(x) - x_i)\,(x_i - a_{ij}) = \xi_i^\varepsilon(M_c(x) - a_j)^2 - \xi_i^\varepsilon(x - a_j)^2$$

Substituting this expression into (32) finally yields

$$\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}_\varepsilon(x))\, \|\xi^\varepsilon(x-a_j)\|_p^{1-p}\, \xi_i^\varepsilon(x-a_j)^{p-2} \left[\xi_i^\varepsilon(M_c(x) - a_j)^2 - \xi_i^\varepsilon(x - a_j)^2\right] \leq 0 \tag{33}$$

Introducing now the function $\varphi : \mathbb{R}_+^m \to \mathbb{R}$ with

$$\varphi(z) := g\left(z_1^{1/p},\dots, z_m^{1/p}\right) \tag{34}$$

relation (33) reduces for every fixed $x \in U$ to

$$h_i(M_c(x)) \leq h_i(x),\ i = 1, 2,\dots, n \tag{35}$$

where $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function given by

$$h_i(y) := \sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j} (\mathbf{d}_\varepsilon(x)^p)\, \xi_i^\varepsilon(x-a_j)^{p-2}\, \xi_i^\varepsilon(y-a_j)^2 \tag{36}$$

To prove that the sequence of points $\{M_c(x^{(k)}) : k \geq 0\}$ generated by the adapted Weiszfeld method is strictly decreasing whenever $1 < p \leq 2$, we need the following result due to Beckenbach and Bellman [3]:

Lemma 3.1
If $a,b > 0$, $u < 1$ $(u \neq 0)$ and $\frac{1}{u} + \frac{1}{v} = 1$ then $a^{1/u}\, b^{1/v} \geq \frac{a}{u} + \frac{b}{v}$.

Theorem 3.2
Let $1 < p \leq 2$ and suppose $\varphi : \mathbb{R}_+^m \rightarrow \mathbb{R}$ given by (34) is a quasiconcave function on $\mathbb{R}_+^m$. Then $f_\varepsilon(M_c(x)) < f_\varepsilon(x)$ for every $0 < c \leq 1$ and $x \in U$ satisfying $\nabla f_\varepsilon(x) \neq \mathbf{0}$.

Proof
Consider for fixed $x \in U$ with $\nabla f_\varepsilon(x) \neq \mathbf{0}$ and $1 < p < 2$ the functions $s_i : \mathbb{R}^n \rightarrow \mathbb{R}$ given by

$$s_i(y) := \sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j} (\mathbf{d}_\varepsilon(x)^p)\, \xi_i^\varepsilon(y-a_j)^p,\ i = 1, 2, \ldots, n \tag{37}$$

Clearly $h_i(x) = s_i(x)$ for every $1 \leq i \leq n$ with h_i given by (36). Furthemore, since $\nabla f_\varepsilon(x) \neq \mathbf{0}$, at least one of the inequalities in (35) is strict. Taking $a := \xi_i^\varepsilon(M_c(x)-a_j)^p$, $b := \xi_i^\varepsilon(x-a_j)^p$ and $\frac{1}{u} := \frac{2}{p}$ it follows by Lemma 3.1 that

$$h_i(M_c(x)) \geq \frac{2}{p}\, s_i(M_c(x)) + (1 - \frac{2}{p})\, s_i(x) \tag{38}$$

Inequalities (35) and (38) together with $s_i(x) = h_i(x)$ yield $s_i(M_c(x)) \leq s_i(x)$ with a strict inequality for al least one value of i. Hence, $\sum_{i=1}^{n} s_i(M_c(x)) < \sum_{i=1}^{n} s_i(x)$ and so by the definition of s_i we obtain that

$$\sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j} (\mathbf{d}_\varepsilon(x)^p) \left[\|\xi^\varepsilon(M_c(x) - a_j)\|_p^p - \|\xi^\varepsilon(x - a_j)\|_p^p\right] < 0$$

By the above inequality and the quasiconcavity of φ the desired result now follows for $1 < p < 2$.

To verify the above result for $p = 2$ we first observe that the functions h_i and s_i are identical and so relation (38) is trivially satisfied. Replacing now p in the first part of this proof everywhere by 2 yields the desired result for $p = 2$. ♦

If Γ_{f_ε} denotes the set of stationary points of f_ε, i.e. $\Gamma_{f_\varepsilon} := \{x \in \mathbb{R}^n : \nabla f_\varepsilon(x) = \mathbf{0}\}$ it is clear since (P_ε) is solvable that Γ_{f_ε} is not empty. Moreover, we obtain by (26) that $\Gamma_{f_\varepsilon} \subseteq U$. Observe now for $1 < p \leq 2$, since $M_1 : U \rightarrow U$ is continuous on the compact set U and $\{x^{(k)} :$

$k \geq 0\} \subseteq U$, that by Theorem 7.2.3 of [2] and Theorem 3.2 the sequence $\{x^{(k)} : k \geq 0\} \subseteq U$ contains a limit point belonging to Γ_{f_ε}. In addition, if Γ_{f_ε} is a finite set, this sequence converges to a point $x \in \Gamma_{f_\varepsilon}$. Finally, to conclude the analysis for $1 < p \leq 2$, if the functions $g : \mathbb{R}^m_+ \to \mathbb{R}$ and $\varphi : \mathbb{R}^m_+ \to \mathbb{R}$ are quasiconvex, respectively quasiconcave on $\mathbb{R}^m_+$ and $\nabla g(\mathbf{d}_\varepsilon(x)) \in \text{int}(\mathbb{R}^m_+)$ for every $x \in U$, it can be shown (by adapting the proof of Theorem 2.9) that the sequence $\{x^{(k)} : k \geq 0\} \subseteq U$ converges to the unique optimal solution x^*_ε.

As for the unperturbed optimization problem and $p = 2$ these convergence results also hold for each fixed $1 < p \leq 2$ if the domain of the function g is replaced by a convex subset $S \subseteq \mathbb{R}^m_+$ satisfying

i) $\mathbf{d}_\varepsilon(x) \in \text{int}(S)$ for every $x \in U$

ii) $S^{1/p} \subseteq S$ with $S^{1/p} := \left\{ (z_1^{1/p}, \ldots, z_n^{1/p}) : z \in S \right\}$

iii) The function g is continuously differentiable on int(S) and the vector-valued function $z \to \nabla g(z)$ can be continuously extended to the boundary of S.

As a consequence the function φ given by (34) can be defined on the same domain S.

Let us examine now the behaviour of the adapted Weiszfeld method for $p > 2$. Consider again the four demand points defined in the example presented in Section 2 and set $w_1 = w_2 = w > 0$ and $w_3 = w_4 = \overline{w} \neq w$, $\overline{w} > 0$. It is not difficult to prove that the global minimizer of the perturbed Weber function $f_\varepsilon(x) = \sum_{j=1}^{4} w_i \, \|\xi^\varepsilon(x - \mathbf{a}_j)\|_p$ is on the line $x_1 = 0$. For $k \in \mathbb{R}_+$ sufficiently large and $0 < \varepsilon \leq \frac{a}{k}$, it is shown in [21] that starting at the point $x^{(0)} = (0,0)$ the adapted Weiszfeld procedure generates in the first iteration a point (0,y) such that $f_\varepsilon((0,y)) > f_\varepsilon((0,0))$. This means, contrary to the case $1 < p \leq 2$, that the descent property of the algorithm does not hold for $p > 2$. However, it does not imply that there exists no limit point belonging to Γ_{f_ε} of the sequence $\{x^{(k)} : k \geq 0\}$. At present, the existence of a subsequence of $\{x^{(k)} : k \geq 0\}$ converging to an element of Γ_{f_ε} is still an open problem for $p > 2$.

4. Rate of Convergence Analysis

Before discussing the rate of convergence of the algorithms derived in Sections 2 and 3 we need the following framework to solve unconstrained optimization problems proposed by Voss and Eckhardt [31]. Let us consider the optimization problem $\min\{h(x) : x \in \mathbb{R}^n\}$ with $h : \mathbb{R}^n \to \mathbb{R}$ a twice continuously differentiable function. For this problem Voss and Eckhardt [31] proposed a so-called quadratic approximation method.

If $x^{(k)}$ is the present iteration point approximate h by a strongly convex (see [1]) quadratic function $\Psi_k : \mathbb{R}^n \to \mathbb{R}$ satisfying $\Psi_k(x^{(k)}) = h(x^{(k)})$ and $\nabla\Psi_k(x^{(k)}) = \nabla h(x^{(k)})$. Clearly by these assumptions the function $\Psi_k : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\Psi_k(x) = \frac{1}{2}(x - x^{(k)})^T C_k(x - x^{(k)}) + \nabla h(x^{(k)})^T (x - x^{(k)}) + h(x^{(k)})$$

with C_k a symmetric positive definite $n \times n$ matrix.

Take now as the next iteration point

$$x^{(k+1)} := \operatorname{argmin}\{\Psi_k(x) : x \in \mathbb{R}^n\} \tag{39}$$

and continue this procedure until a given stopping rule is satisfied. For this algorithm the following result is proved in [31] (see also [21]).

Theorem 4.1

Let x^* denote an optimal solution of the optimization problem $\min\{h(x) : x \in \mathbb{R}^n\}$ and suppose the Hessian $\nabla^2 h(x^*)$ at this point x^* is positive definite. If the sequence $\{x^{(k)} : k \geq 0\}$ generated by the above algorithm without applying a stopping rule converges to x^* and satisfies

i) $\Psi_k(x^{(k+1)}) \geq h(x^{(k+1)})$ for every $k \geq 0$

ii) $\sup_{k\geq 0} \rho(C_k) < \infty$ with $\rho(C_k)$ the spectral radius of the matrix C_k

iii) $\inf_{k\geq 0} \lambda_k > 0$ with λ_k the smallest eigenvalue of the matrix C_k

then there exists some K and $0 < \Lambda < 1$ such that

$$0 \leq h(x^{(k+1)}) - h(x^*) \leq \Lambda[h(x^{(k)}) - h(x^*)]$$

for every $k \geq K$.

Moreover, if additionally h is strongly convex on some compact convex set $A \subseteq \mathbb{R}^n$ and $\{x^{(k)} : k \geq 0\} \subseteq A$ then the above constant K can be taken equal to zero.

Clearly, since Ψ_k is strongly convex, it follows by (39) that $x^{(k+1)}$ is the unique optimal solution of the set of nonlinear equations

$$0 = \nabla\Psi_k(x) = C_k(x - x^{(k)}) + \nabla h(x^{(k)})$$

and so $x^{(k+1)} = x^{(k)} - C_k^{-1}\nabla h(x^{(k)})$.

Observe by (10) that the Weiszfeld algorithm applied to (P) for Euclidean distances satisfies this iterative formula whenever $x^{(k)}$ belongs to co(D)\D. In this case replace C_k by $b(x^{(k)})I$ and $\nabla h(x^{(k)})$ by $\nabla f(x^{(k)})$. Also it is not difficult to verify that the corresponding quadratic function is given by

$$\Psi_k(x) = \frac{1}{2}\sum_{j=1}^{m} \frac{\partial g}{\partial z_j}(\mathbf{d}(x^{(k)}))\, \|x^{(k)} - a_j\|_2^{-1}\, (\|x - x^{(k)}\|_2^2 + 2(x^{(k)} - a_j)^T (x - x^{(k)})) + f(x^{(k)}) \tag{40}$$

Rewriting Ψ_k yields by (15) that

$$\Psi_k(x) = \sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j}(\mathbf{d}^2(x^{(k)}))\, (\|x - x^{(k)}\|_2^2 + 2(x^{(k)} - a_j)^T (x - x^{(k)})) + f(x^{(k)}) \tag{41}$$

An application of Theorems 2.9 and 4.1 is now presented by the following result for the Weiszfeld algorithm applied to (P) for p = 2.

Theorem 4.2

Let $g : S \to \mathbb{R}$ be a nondecreasing twice continuously differentiable convex function satisfying $\nabla g(\mathbf{d}(x)) \in \operatorname{int}(\mathbb{R}_+^m)$ for every $x \in$ co(D) and $\varphi : \mathbb{R}_+^m \to \mathbb{R}$, given by (15), a concave function. If this holds and the set D of demand points is strongly not collinear then there exists a unique optimal solution x^* of (P). Moreover, if x^* does not belong to D and the Weiszfeld method starts in the point $x^{(1)}$ given by the remark after Theorem 2.7 the sequence $\{x^{(k)} : k \geq 0\}$ generated by the Weiszfeld algorithm satisfies the conditions of Theorem 4.1.

Proof

The first part of this result is already presented by Theorem 2.9. Moreover, by the construction of the starting point $x^{(1)} \notin D$ it follows that $\{x^{(k)} : k \geq 0\} \subseteq \text{co}(D) \setminus D$ and so by Theorem 2.7 we obtain that $\lim_{k \to \infty} x^{(k)} = x^* \notin D$. Also by this observation it is clear that $\inf_{k \geq 0}(\min_{1 \leq j \leq m}(\|x^{(k)} - a_j\|_2)) > 0$ and hence, since C_k equals $b(x^{(k)})I$, the conditions (ii) and (iii) of Theorem 4.1 follow immediately. To verify condition (i) of Theorem 4.1 we notice by the concavity of φ that

$$f(x^{(k+1)}) - f(x^{(k)}) \leq \sum_{j=1}^{m} \frac{\partial \varphi}{\partial z_j} (\mathbf{d}^2(x^{(k)}))\, (\|x^{(k+1)} - a_j\|_2^2 - \|x^{(k)} - a_j\|_2^2)$$

This implies by (41) and using

$$\|x^{(k+1)} - a_j\|_2^2 - \|x^{(k)} - a_j\|_2^2 = \|x^{(k+1)} - x^{(k)}\|_2^2 + 2(x^{(k+1)} - x^{(k)})^T (x^{(k)} - a_j)$$

that

$$f(x^{(k+1)}) - f(x^{(k)}) \leq \Psi_k(x^{(k+1)}) - f(x^{(k)})$$

Finally, for the verification of the result that $\nabla^2 f(x^*)$ is positive definite we need that g is convex, $\nabla g(\mathbf{d}(x)) \in \text{int}(\mathbb{R}_+^m)$ for every $x \in \text{co}(D)$ and D is strongly not collinear. However, the proof is rather long and technical and so we refer the reader to similar proofs discussed in [21] or [13]. ◆

To conclude this section we observe that a similar rate of convergence result can be proved for the perturbed problem (P_ε) if $1 \leq p \leq 2$. However, for a detailed discussion of these results the reader is referred to [14].

5. Computational Results

The algorithms derived in Sections 2 and 3 were applied for $n = 2$ to two sets of test problems. The first set is given by $g_\alpha : \mathbb{R}_+^m \to \mathbb{R}$ with $g_\alpha(z) = \sum_{j=1}^{m} w_j z_j^\alpha$, $w_j > 0$ for every $j = 1,\ldots, m$ and $\sum_{j=1}^{m} w_j = 1$. The corresponding function $f_\alpha : \mathbb{R}^2 \to \mathbb{R}$ has the form

$$f_\alpha(x) := \sum_{j=1}^{m} w_j \|x - a_j\|_p^\alpha \qquad (42)$$

and the perturbed version is denoted by $f_{\varepsilon,\alpha}(x)$.

In (42) the parameters α and p satisfy $1 < p \leq 2$ and $1 \leq \alpha \leq p$. Clearly for this choice the function g_α is convex while the function φ_α given by (15) and(34) is concave.

Introduce now for fixed $1 < p \leq 2$ the constant $M_p := \max\{\sum_{j=1}^{m} w_j \|x - a_j\|_p, x \in U\} + \beta$ with $\beta > 1$ and consider on the convex set $S_p := \{z \in \mathbb{R}_+^m : M_p - \sum_{j=1}^{m} w_j z_j > 0\}$ the function

$$g(z) := \frac{\sum_{j=1}^{m} w_j z_j^{\alpha}}{M_p - \sum_{j=1}^{m} w_j z_j}$$

The second set of test problems associated with g is now given by

$$\bar{f}_\alpha(x) := \frac{\sum_{j=1}^{m} w_j \|x - a_j\|_p^{\alpha}}{M_p - \sum_{j=1}^{m} w_j \|x - a_j\|_p} \tag{43}$$

with $1 \leq \alpha \leq p$.

Moreover, denote the perturbed version by $\bar{f}_{\varepsilon,\alpha}(x)$. Clearly for every $1 \leq \alpha \leq 2$ the function $g : S_p \to \mathbb{R}$ being the ratio of a nonnegative convex function and a positive concave function is quasiconvex on S_p. Also for $\varepsilon > 0$ small enough it is easy to verify by the definition of M_p that for fixed $1 < p \leq 2$ the vector $\mathbf{d}_\varepsilon(x)$ belongs to $\mathrm{int}(S_p)$ for every $x \in U$ while for $p = 2$ the vector $\mathbf{d}(x)$ belongs to $\mathrm{int}(S_2)$ for every $x \in \mathrm{co}(D) \backslash D$. Finally, we observe by Theorem 62.A of [29] that $\left(\sum_{j=1}^{m} w_j z_j^{1/p}\right)^p \leq \sum_{j=1}^{m} w_j z_j$ for every $z \in \mathbb{R}_+^m$ and $1 < p \leq 2$. Hence using $M_p > 1$ it follows that

$$\begin{aligned} M_p - \sum_{j=1}^{m} w_j z_j^{1/p} &\geq \min\left(M_p - 1, M_p - \left(\sum_{j=1}^{m} w_j z_j^{1/p}\right)^p\right) \\ &\geq \min\left(M_p - 1, M_p - \sum_{j=1}^{m} w_j z_j\right) > 0 \end{aligned}$$

for every $z \in S_p$. This means that $S_p^{1/p} \subseteq S_p$ and so the function φ given by (15) and (34) can be defined on S_p. Also by the first part of Theorem 5.15 of [1] it follows that φ is quasiconcave on S_p for $1 \leq \alpha \leq p$. By these observations and the remarks at the end of Sections 2 and 3 we obtain that both test sets satisfy the conditions of the convergence results discussed in Sections 2 and 3.

Both algorithms were now applied to the two types of functions given above and 20 sets with clusters of demand points randomly generated as follows: for m = 50, 100, 200 and 300, five data sets were generated each time at random. First we draw two numbers m_1 and m_2 ranging between 1 and 20 and then we divide the square $[0,250] \times [0,250]$ into $(m_1+1)\ (m_2+1)$

subsquares by randomly generating m_1 coordinates in the x-axis and m_2 coordinates in the y-axis. The subsquares thus obtained are labelled from 1 to (m_1+1) (m_2+1) (see Figure 1). Afterwards, a random number of subsquares is chosen (each one corresponding to a cluster) and in each subsquare a given number of demand points in uniformly draw. Finally, the remaining demand points are randomly generated in the original square [0,250] × [0,250] and added to the already existing set, in a total of m points.

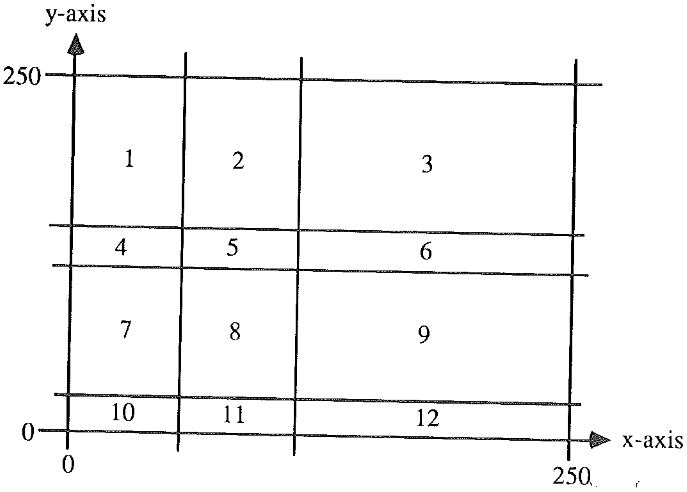


Figure 1 - Clustered problem with $m_1 = 2$, $m_2 = 3$

In order to obtain each weight w_j associated with the j-th demand point $\mathbf{a}_j$, m numbers $\overline{w}_j$ are drawn from a uniform distribution in [0,1]. Each weight w_j is then set equal to the ratio between $\overline{w}_j$ and $\sum_{j=1}^{m} \overline{w}_j$.

Based on the computation of an upper bound for the relative error RE := $[f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)]/f(\mathbf{x}^*)$ (with f denoting the functions f_α, $\overline{f}_\alpha$, $f_{\varepsilon,\alpha}$ and $\overline{f}_{\varepsilon,\alpha}$) obtained during the k-th iteration, a stopping rule for the two algorithms can be derived. If f is a convex function it follows that

$$LB(\mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k)})^T \mathbf{x}^{(k)} + \min_{\mathbf{y}\in\Omega} \{\nabla f(\mathbf{x}^{(k)})^T \mathbf{y}\} \tag{44}$$

is a valid lower bound on the minimum value of the objective function at the k-th iteration (see [20]). As a result, if

$$\frac{\nabla f(\mathbf{x}^{(k)})^T \mathbf{x}^{(k)} - \min_{\mathbf{y}\in\Omega}\{\nabla f(\mathbf{x}^{(k)})^T \mathbf{y}\}}{\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k)})^T \mathbf{x}^{(k)} + \min_{\mathbf{y}\in\Omega}\{\nabla f(\mathbf{x}^{(k)})^T \mathbf{y}\}} \le \delta \tag{45}$$

the iterative procedure stops and the current point $\mathbf{x}^{(k)}$ is accepted as an adequate solution for the problem under consideration, with $\delta > 0$ a prespecified tolerance.

For the quasiconvex functions $\overline{f}_{\alpha}$ and $\overline{f}_{\varepsilon,\alpha}$ there is no guarantee that (44) is still a valid lower bound and so the stopping rule (45) may not be correct. Nevertheless, (45) was also applied in this case and in all the computational tests performed we observed that the value of (44) was always below the corresponding objective function value.

The two algorithms were coded in Pascal and compiled with Turbo Pascal 5.0 in a Unisys PC/AT (80286) with mathematical co-processor. In all tests performed the tolerance δ used in the stopping rule and the constant ε in the hyperbolic approximation of the L_p-norm were both set to 5×10^{-6}. Moreover, the initial point $x^{(0)}$ was chosen at random from the domain Ω in all tests. Before applying the generalized Weiszfeld method, we also checked for each problem if any demand point was optimal by using the result given in Lemma 2.1.

In Table I we summarize the results obtained for the convex functions f_{α} and $f_{\varepsilon,\alpha}$ for $p = 2.0$ and $\alpha = 1.3$. The first column indicates the size of each data set and is followed by five columns with the results of the generalized Weiszfeld procedure. The remaining five columns contain the results for the Weiszfeld procedure with the ξ^{ε}-approximation. For each method we present the total number of iterations performed (columns 2 and 7), the coordinates of the points obtained (columns 3 and 8), the corresponding objective function values (columns 4 and 9), the lower bounds based on the solutions found (columns 5 and 10) and finally the execution times in seconds (columns 6 and 11).

	Generalized Weiszfeld Method					Adapted Weiszfeld Method				
m	k	$x^{(k)}$	$f_{\alpha}(x^{(k)})$	$LB(x^{(k)})$	CPU	k	$x^{(k)}$	$f_{\epsilon,\alpha}(x^{(k)})$	$LB(x^{(k)})$	CPU
50	10	(46.630, 176.860)	241.698	241.697	1.70	11	(46.630, 176.860)	241.698	241.698	3.00
50	16	(85.033, 103.492)	318.369	318.368	2.64	19	(85.033, 103.492)	318.369	318.368	5.34
50	22	(179.102, 84.407)	247.691	247.690	3.63	22	(179.102, 84.407)	247.691	247.690	6.10
50	14	(141.703, 95.533)	377.340	377.340	2.25	12	(141.703, 95.533)	377.340	377.340	3.35
50	14	(145.318, 114.948)	314.853	314.851	2.08	15	(145.318, 114.948)	314.853	314.851	4.13
100	15	(106.883, 126.627)	378.052	378.051	4.44	16	(106.883, 126.627)	378.052	378.050	8.46
100	11	(123.490, 144.328)	329.918	329.917	3.24	10	(123.490, 144.328)	329.918	329.917	5.28
100	12	(97.789, 94.355)	272.553	272.552	3.57	11	(97.789, 94.355)	272.553	272.552	5.76
100	11	(103.348, 101.993)	308.108	308.107	3.29	10	(103.349, 101.994)	308.108	308.107	5.38
100	11	(78.707, 141.615)	340.584	340.584	3.35	10	(78.707, 141.615)	340.584	340.583	5.38
200	12	(147.110, 85.496)	303.317	303.317	6.69	11	(147.110, 85.496)	303.317	303.317	11.27
200	17	(124.198, 109.904)	325.931	325.930	9.56	16	(124.198, 109.904)	325.931	325.930	16.36
200	12	(115.017, 130.026)	412.076	412.075	6.75	12	(115.017, 130.026)	412.076	412.075	12.36
200	15	(168.990, 143.376)	185.581	185.580	8.62	14	(168.990, 143.376)	185.581	185.580	14.45
200	13	(71.017, 180.777)	291.318	291.317	7.31	13	(71.017, 180.777)	291.318	291.317	13.28
300	16	(93.670, 54.871)	310.811	310.810	13.17	16	(93.670, 54.871)	310.811	310.810	24.44
300	12	(121.539, 94.117)	232.443	232.442	9.88	13	(121.539, 94.117)	232.443	232.443	19.89
300	10	(134.543, 134.090)	447.012	447,011	8.30	9	(134.543, 134.090)	447.012	447.010	13.67
300	10	(91.189, 167.436)	337.192	337.191	8.35	11	(91.189, 167.436)	337.192	337.191	16.80
300	13	(115.133, 107,280)	256.261	256.260	10.88	12	(115.133, 107,280)	256.261	256.260	18.34

Table I – Computational results for $p = 2.0$, $\alpha = 1.3$ and the convex functions f_{α} and $f_{\varepsilon,\alpha}$

Observe that both methods give the same solutions in 18 out of 20 problems and that the corresponding values of the objective functions are the same in all cases. The total number of iterations performed is also similar in both algorithms. However, the adapted Weiszfeld procedure requires larger execution time since more calculations are involved when the hyperbolic approximation of the norm is used.

Table II contains the results for the quasiconvex functions $\overline{f}_{\alpha}$ and $\overline{f}_{\varepsilon,\alpha}$ for $p = 2.0$ and $\alpha = 1.3$. The information included in this table is of the same type as in Table I. In this case, the columns representing the lower bounds are not included due to the already mentioned reason. Again we observe that both methods produce the same solutions and the same objective function values. Moreover, the results suggest that the problems with quasiconvex objectives are more difficult to solve since the number of iterations required is larger in comparison with the convex case (Table I). As a result, the CPU times increase twice on average but still remain under 45 seconds which seems reasonable taking into account the size of the data when this occurs.

Finally, Tabel III presents the results of the adapted Weiszfeld method with the functions $f_{\varepsilon,\alpha}$ and $\overline{f}_{\varepsilon,\alpha}$ and the set of parameters p = 1.5, α = 1.3. The first column gives the size of each test problem and the next five columns contain the results for the convex function $f_{\varepsilon,\alpha}$. The last four columns show the results for the quasiconvex function $\overline{f}_{\varepsilon,\alpha}$. Observe once more that the function $f_{\varepsilon,\alpha}$ requires a smaller number of iterations in order to satisfy the prespecified tolerance. On average, 20.35 iterations need to be performed while in the quasiconvex case 27.6 iterations are required. Consequently, the computational times are obviously larger for $\overline{f}_{\varepsilon,\alpha}$ and in two problems more than 1 minute of CPU time was spent.

	Generalized Weiszfeld Method				Adapted Weiszfeld Method			
m	k	$x^{(k)}$	$\overline{f}_\alpha(x^{(k)})$	CPU	k	$x^{(k)}$	$\overline{f}_{\epsilon,\alpha}(x^{(k)})$	CPU
50	12	(45.154, 178.431)	1.736	2.64	12	(45.154, 178.431)	1.736	4.99
50	25	(76.422, 107.838)	5.247	5.06	24	(76.422, 107.838)	5.247	9.56
50	32	(179.318, 78.564)	3.659	6.26	33	(179.318, 78.564)	3.659	13.08
50	19	(143.455, 94.728)	3.658	3.80	19	(143.455, 94.728)	3.658	7.64
50	21	(149.749, 116.778)	3.707	4.25	21	(149.749, 116.778)	3.707	8.34
100	15	(105.708, 123.495)	4.445	5.72	15	(105.708, 123.495)	4.445	11.75
100	15	(124.656, 145.182)	3.946	5.78	16	(124.656, 145.182)	3.946	12.37
100	16	(97.342, 92.479)	3.096	6.09	14	(97.342, 92.479)	3.096	10.82
100	13	(103.297, 102.265)	3.349	4.78	14	(103.297, 102.265)	3.349	10.82
100	12	(79.117, 145.252)	3.208	4.55	16	(79.117, 145.252)	3.208	12.41
200	16	(147.702, 85.329)	2.608	11.32	15	(147.702, 85.329)	2.608	22.69
200	23	(126.914, 110.852)	2.715	16.03	25	(126.914, 110.852)	2.715	36.63
200	16	(113.644, 129.232)	4.852	11.44	13	(113.645, 129.232)	4.852	19.93
200	18	(169.597, 144.155)	1.424	12.74	18	(169.597, 144.155)	1.424	26.86
200	15	(68.783, 181.830)	2.129	10.61	15	(68.783, 181.830)	2.129	22.68
300	20	(89.792, 51.827)	2.249	20.71	20	(89.792, 51.827)	2.249	44.31
300	17	(120.722, 93.660)	1.884	17.74	15	(120.722, 93.660)	1.884	33.67
300	11	(135.618, 133.425)	5.932	11.48	13	(135.618, 133.425)	5.932	29.49
300	13	(90.400, 168.544)	2.819	13.79	15	(90.400, 168.544)	2.819	33.74
300	17	(115.750, 106.836)	1.932	17.74	17	(115.750, 106.836)	1.932	37.79

Table II – Computational results for p = 2.0, α = 1.3 and the convex functions $\overline{f}_\alpha$ and $\overline{f}_{\varepsilon,\alpha}$

	$p = 1.5,\ \alpha = 1.3$					$p = 1.5,\ \alpha = 1.3$			
m	k	$x^{(k)}$	$f_{\epsilon,\alpha}(x^{(k)})$	$LB(x^{(k)})$	CPU	k	$x^{(k)}$	$\overline{f}_{\epsilon,\alpha}(x^{(k)})$	CPU
50	19	(49.802, 178.111)	263.374	263.374	5.51	20	(48.586, 180.046)	2.919	8.95
50	24	(86.182, 105.089)	350.182	350.181	6.99	27	(74.947, 107.204)	7.889	12.04
50	24	(181.738, 83.375)	262.444	262.443	7.08	36	(181.535, 78.191)	2.419	16.00
50	22	(143.254, 97.583)	423.943	423.941	6.43	32	(146.129, 96.543)	5.131	14.23
50	21	(136.141, 115.098)	347.926	347.926	6.20	33	(140.858, 116.470)	5.599	14.66
100	20	(110.062, 121.666)	429.307	429.305	11.09	26	(107.193, 120.324)	6.244	22.25
100	18	(129.846, 144.665)	346.632	346.630	9.98	39	(132.568, 145.523)	26.782	32.95
100	17	(94.569, 95.248)	300.623	300.621	9.50	27	(93.848, 93.014)	4.850	23.14
100	19	(105.041, 104.928)	330.283	330.282	10.50	22	(104.790, 104.173)	9.327	18.96
100	20	(76.421, 148.616)	378.202	378.200	10.99	29	(76.161, 153.205)	8.217	24.71
200	21	(152.816, 85.242)	327.095	327.094	22.84	32	(154.194, 85.337)	4.993	53.03
200	23	(125.900, 109.511)	355.473	355.472	24.67	32	(129.902, 110.266)	5.268	52.62
200	19	(106.107, 118.173)	454.525	454.523	20.55	26	(102.526, 115.941)	11.234	43.39
200	20	(167.071, 143.493)	201.984	201.983	21.75	26	(168.795, 145.743)	5.954	43.32
200	20	(69.286, 184.018)	321.302	321.301	21.70	27	(64.999, 185.367)	6.153	45.15
300	21	(95.510, 49.462)	339.780	339.780	33.93	26	(93.638, 47.955)	1.431	64.82
300	22	(123.153, 91.302)	254.244	254.243	35.31	22	(122.228, 91.288)	1.379	54.97
300	17	(139.394, 138.234)	492.258	492.256	27.42	22	(140.699, 137.843)	4.937	54.93
300	19	(83.638, 173.188)	367.431	367.429	30.66	23	(82.476, 174.342)	2.217	57.43
300	21	(114.976, 103.373)	285.564	285.563	33.96	25	(115.433, 102.875)	1.396	62.13

Table III – Computational results for the adapted Weiszfeld procedure and the functions $f_{\varepsilon,\alpha}$ and $\overline{f}_{\varepsilon,\alpha}$

More details about the computational experience with the same objective functions but other sets of values for the parameters p, α, δ and ε can be found in [14] and [21]. In [21] results for data sets without clusters of demand points are also presented.

In general, when the data sets contain clusters of demand points it seems that the corresponding problems are more difficult to solve especially if the optimal point is in a neighbourhood of a cluster or between two clusters. In those cases slow convergence may be observed probably due to the flatness of the gradient in the neighbourhood of the optimal solution. However, the so-called clustered problems can better approximate real-life situations and hence are more interesting to investigate.

6. Conclusions

Table IV summarizes the results derived in Sections 2, 3 and 4 for the two algorithms based on the Weiszfeld method and here proposed for solving the single facility continuous space location problem with distances modelled by some L_p-norm.

We can conclude that the methods presented are gradient methods which do not require line minimization. They are simple to implement and produce good solutions in a reasonable amount of time. However, the conditions under which global convergence can be established may not be easy to verify in practice.

Problem	Weiszfeld Method	Lp-norm	Behaviour
without approximation	generalization	$1 < p < 2$ $p = 2$ $p > 2$	not convergent linearly convergent* not convergent
with the ξ^{ε}-approximation	adaptation	$1 < p \leq 2$ $p > 2$	linearly convergent* unknown

Table IV – * under certain conditions (Theorems 2.9 and 4.2)

Acknowledgments

The authors would like to thank the anonymous referee for his useful comments and Frank Plastria for sending them his recent excellent survey paper on continuous location.

7. References

[1] Avriel, M., Diewert, W.E., Schaible, S. and Zang, I., Generalized Concavity, Plenum Press, New York, 1988.

[2] Bazaraa, M.S. and Shetty, C.M., Nonlinear Programming (Theory and Algorithms), Wiley, New York, 1979.

[3] Beckenbach, E.F. and Bellman, R., Inequalities, Springer Verlag, New York, 1967.

[4] Berens, W., The Suitability of the Weighted L_p-norm in Estimating Actual Road Distances, European Journal of Operational Research 34 (1988) 39-43.

[5] Berens, W. and Körling, F.-J., On Estimating Road Distances by Mathematical Functions - a Rejoinder, European Journal of Operational Research 36 (1988) 254-255.

[6] Bertsekas, D.P., Constrained Optimization and Lagrange Multiplier Methods, Academic Press, New York, 1982.

[7] Brimberg, J. and Love, R.F., Estimating Travel Distances by the Weighted L_p-norm, Naval Research Logistics 38 (1991) 241-259.

[8] Brimberg, J. and Love R.F., A New Distance Function for Modelling Travel Distances in a Transportation Network, Transportation Science 26 (1992) 129-137.

[9] Dax, A., The Use of Newton's Method for Solving Euclidean Multifacility Location Problems, Technical Report Hydrological Service, Jerusalem, 1983.

[10] Eyster, J.W., White, J.A. and Wierwille, W.W., On Solving Multifacility Location Problems Using a Hyperbolic Approximation Procedure, AIIE Transactions 5 (1973) 1-6.

[11] Francis, R.L. and Cabot, A.V., Properties of a Multifacility Location Problem Involving Euclidian Distances, Naval Research Logistics Quarterly 19 (1972) 335-353.

[12] Francis, R.L. and White, J.A., Facility Layout and Location: An Analytical Approach, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1974.

[13] Frenk, J.B.G., Labbé, M., Visscher, R.J. and Zhang, S., The Stochastic Queue Location Problem in the Plane, Report 8948/A, Econometric Institute, Erasmus University, Rotterdam, 1989.

[14] Frenk, J.B.G., Melo, M.T. and Zhang, S., A Weiszfeld Method for a Generalized Lp Distance Minisum Location Model in Continuous Space, to appear in Location Science, 1994.

[15] Kuhn, H.W., A Note on Fermat's Problem, Mathematical Programming 4 (1973) 98-107.

[16] Love, R.F. and Morris, J.G., Mathematical Models of Road Travel Distances, Management Science 25 (1979) 130-139.

[17] Love, R.F. and Morris, J.G., Modelling Inter-City Road Distances by Mathematical Functions, Operational Research Quarterly 23 (1972) 61-71.

[18] Love, R.F. and Morris, J.G., On Estimating Road Distances by Mathematical Functions, European Journal of Operational Research 36 (1988) 251-253.

[19] Love, R.F., Morris J.G. and Wesolowsky, G.O., Facilities Location (Models and Methods), North Holland, New York, 1988.

[20] Love, R.F. and Yeong, W.Y., A Stopping Rule for Facilities Location Algorithms, AIIE Transactions 13 (1981) 357-362.

[21] Melo, M.T., Solving the Single Facility Continuous Space Location Problem Using the Weiszfeld Method - Advantages and Disadvantages, Master Thesis, DEIO-Faculty of Sciences University of Lisbon, 1991.

[22] Morris, J.G., Convergence of the Weiszfeld Algorithm for Weber Problems Using a Generalized Distance Function, Operations Research 29 (1981) 37-48.

[23] Morris, J.G. and Verdini, W.A., Minisum L_p Distance Location Problems solved via a Perturbed Problem and Weiszfeld's Algorithm, Operations Research 27 (1979) 1180-1188.

[24] Muller, J.C., La Cartographie d'une Metrique Non Euclidienne: Les Distance-temps, L'Espace Géographique 3 (1979) 215-227.

[25] Muller, J.C., Non Euclidean Geographic Spaces: Mapping Functional Distances, Geographical Analysis 14 (1982) 189-203.

[26] Nemhauser, G.L., Rinnooy Kan, A.H.G. and Todd, M.J. Editors, Handbooks in Operations Research and Management Science, Volume 1-Optimization (1989), Chapter VII, 529-572, North-Holland.

[27] Ostresh, L.M. Jr., On the Convergence of a Class of Iterative Methods for Solving the Weber Location Problem, Operations Research 26 (1978) 597-609.

[28] Plastria, F., Continuous Location Anno 1992: A Progress Report, Studies in Locational Analysis 5 (1993) 85-127.

[29] Roberts, A.W. and Varberg, D.E., Convex Functions, Academic Press, New York, 1973.

[30] Rudin, W., Principles of Mathematical Analysis, Third Edition, International Student Edition , McGraw-Hill International Book Company, 1976.

[31] Voss, H. and Eckhardt, U., Linear Convergence of the Generalized Weiszfeld's Method, Computing 25 (1980) 243-251.

[32] Weiszfeld, E., Sur le Point sur lequel la Somme des Distances de n Points donnés est Minimum, Tôhoku Mathematical Journal 43 (1937) 355-386.

[33] Zhang, S., Stochastic Queue Location Problems, Ph. D. Thesis, Tinbergen Institute, Research Series 14, Erasmus University, Rotterdam, 1991.

INVESTIGAÇÃO OPERACIONAL, vol. 14, n.° 1 (Junho 1994) 61-75

SOBRE A SOLUÇÃO ADMISSÍVEL INICIAL EM PROGRAMAÇÃO LINEAR

Domingos M. Cardoso
Departamento de Matemática
Universidade de Aveiro/INESC

João C.N. Clímaco
Faculdade de Economia
Universidade de Coimbra/INESC

Abstract

In most approaches, the resolution of a linear programming problem needs a starting feasible solution. Among the techniques commonly used to overcome this difficulty, the construction of an artifical linear program having a known feasible solution and then the application of the big M method is certainly the most popular.

In this paper we study some criteria that provide an estimation of M, in order to get an optimal solution of the original linear program from the feasible solution of the constructed artificial linear program. We also provide some techniques for updating M, which allow to avoid the undesirable consequences that arise from an initial unsafe estinate of M. Although the fact that the main discussed results are oriented to the simplex method at the end of the paper we extend some of the results to interior point methods.

Resumo

A resolução de um programa linear, em geral, obriga à determinação de uma solução admissível inicial, seja ela um ponto interior ou uma solução básica. As técnicas vulgarmente utilizadas para se ultrapassar esta questão consistem na modificação do programa original, introduzindo variáveis artificiais, de modo a que o novo programa aceite como solução admissível um ponto previamente conhecido. De entre os métodos que recorrem a esta técnica destaca-se o, usualmente designado, método do "big M".

Como esta apresentação estudam-se alguns critérios que, não só ajudam a estimar um valor a atribuir a M, de modo a que a partir da solução óptima do programa transformado se obtenha a solução óptima do programa original, como também permitem detectar e prevenir escolhas insuficientemente elevadas para os objectivos pretendidos. Embora os principais resultados obtidos estejam directamente relacionados com o método simplex, a sua aplicação aos métodos de ponto interior é também abordada.

Keywords

Linear Programming, simplex method, big M, interior point algorithms.

1. Introdução

A resolução de um programa linear, por aplicação da generalidade dos principais métodos conhecidos (excepção feita ao método do elipsoide desenvolvido por Shor (1970) e por Yudin e Nemrivoskii (1976) e que, de acordo com uma prova dada por Khachiyan (1979), constitui o primeiro algoritmo polinomial para a resolução de programa lineares), obriga à determinação de uma solução admissível inicial, seja ela um ponto interior ou uma solução básica[1]. As técnicas vulgarmente utilizadas para se ultrapassar esta questão consistem na modificação do programa

[1] Existem ainda outras aproximações que dispensam o conhecimento de uma solução admissível inicial como é o caso do "parametric self-dual method" Dantzig (1963) e do "criss-cross method" Terlaky (1985, 1987).

original, introduzindo variáveis artificiais, de modo a que o novo programa aceite como solução admissível um ponto previamente conhecido. De entre os métodos que recorrem a esta técnica destacam-se naturalmente o método das duas fases e o usualmente designado método do "big *M*". Porém, embora o primeiro destes dois métodos determine sempre uma solução admissível, no caso dela existir, o facto de obrigar à resolução de dois programas distintos, leva a que, em geral, se adopte o segundo.

Todavia, a escolha de um valor suficientemente grande para *M*, que garanta que a solução óptima do programa modificado apresenta com valor nulo todas as componentes relativas às variáveis artificiais, garantindo assim a determinação de uma solução óptima para o programa original, continua a ser um tema de grande actualidade. Com efeito, embora teoricamente a atribuição a *M* de um valor de ordem $O(2^L)$ (com L identificado o "tamanho" dos dados de entrada do programa linear) implique que o programa modificado determine uma solução óptima nas condições anteriormente referidas, os problemas numéricos provocados pela sua sobreavaliação tornam-no impraticável.

Numa abordagem recente desta questão, Kojima, Mizuno e Yoshise (1993) e Ishiara e Kojima (1993) desenvolveram resultados que permitem verificar, durante a aplicação de alguns métodos de ponto interior, se o valor adoptado para *M* ainda é demasiado pequeno, possibilitando nestas condições a sua correcção.

Com o que se segue analisam-se alguns critérios que, não só ajudam a estimar o valor a atribuir a M no programa transformado e a partir do qual se pretende obter uma solução óptima para o programa original, como também permitem detectar e prevenir escolhas de M insuficientemente elevadas para o objectivo pretendido.

Ao longo de todo este texto A identificará uma matriz com m linhas e n colunas ($A \in \mathbf{R}^{m \times n}$) cuja característica é m (logo $m \leq n$) e S o poliedro convexo $\{x \in \mathbf{R}^n : Ax = b, x \geq 0\}$ onde $b \in \mathbf{R}^m$ e x designa o n-uplo de variáveis reais relativamente ao qual a condição $x \geq 0$ determina que todas as variáveis que o compõem só possam tomar valores não negativos.

Dada uma matriz B, Ker(B), denota o subespaço nulo de B, ou seja, o subespaço das soluções do sistema homogéneo associado a B.

Sem perda de generalidade e de modo a simplificar a abordagem que se pretende fazer, vamos considerar que o poliedro S é não vazio e limitado.

Definição 1.1

Seja c um vector de $\mathbf{R}^n$ que não é ortogonal ao subespaço vectorial associado à variedade linear de menor dimensão que contém S [2]. O programa linear:

$$\text{determinar } x^* \in S \text{ tal que } c^t x^* = \max\{c^t x : x \in S\}$$

será identificado por (P).

[2] Deve observar-se que esta condição é equivalente a dizer que c não é ortogonal ao subespaço vectorial gerado pelos vectores definidos por quaisquer dois pontos de S e, no caso de ela não se verificar, então qualquer ponto de S é solução óptima para (P).

Seja $T = \{y \in \mathbb{R}^m : y^t A \geq c^t\}$. O dual de (P) será o seguinte programa linear identificado por (D):

$$\text{determinar } y^* \in T \text{ tal que} y^{*t}b = \max\{y^t b : y \in T\}$$

Nesta definição, x^* e y^* são soluções óptimas, respectivamente, para os programas lineares (P) e (D).

Definição 1.2

Dado um programa linear consistente (PL), υ(PL) identifica o valor óptimo de (PL) e X^*(PL) identifica o conjunto de soluções óptimas para (PL). No caso da função objectivo de (PL), para problemas de maximização (problemas de minimização), não admitir qualquer majorante (minorante), para os pontos da respectiva região admissível, por convenção, dizemos que $\upsilon(\text{PL}) = +\infty$ ($\upsilon(\text{PL}) = -\infty$), ou seja, que υ(PL) não é finito e $X^*(\text{PL}) = \varnothing$.

Uma vez que S é não vazio e compacto [3] (P) tem solução óptima finita, logo, pela teoria da dualidade, existe um par de soluções (x^*, y^*) respectivamente óptimas para (P) e (D) tais que $c^t x^* = \upsilon(P) = \upsilon(D) = y^{*t}b$.

2. Modificação do Programa Original

Sendo x^k um ponto arbitrário de $\mathbb{R}^n$, cujas componentes são não negativas, vamos considerar o subconjunto S^k de $\mathbb{R}^{n+1}$ defenido por

$$S^k = \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{n+1} : [A, b - Ax^k] \begin{bmatrix} x \\ z \end{bmatrix} = b,\ x \geq 0,\ z \in \mathbb{R}_0^+ \right\}$$ [4]

e o subconjunto T^k de $\mathbb{R}^m$ dado por

$$T^k = \left\{ y \in \mathbb{R}^m : y^t [A, b - Ax^k] \geq [c^t, -M] \right\},$$

com M representando um escalar estritamente positivo. A estes subconjuntos vamos associar o par de programas primal dual

$$(P^k) \quad \max \quad c^t x - Mz \qquad\qquad (D^k) \quad \min \quad y^t b$$
$$\text{s.a} \quad \begin{bmatrix} x \\ z \end{bmatrix} S^k \qquad\qquad \text{s.a} \quad y \in T^k.$$

Nestas condições, dado que $\begin{bmatrix} x^k \\ 1 \end{bmatrix}$ é uma solução admissível para (P^k), podemos concluir que (P^k) é consistente e, se (D^k) também é consistente $(T^k = \varnothing)$, então, da teoria da dualidade da programação linear (Dantzig (1963)), vem que ambos os programas admitem solução óptima finita e $\upsilon(P^k) = \upsilon(D^k)$. Reciprocamente, se (P^k) tem solução óptima finita $(\upsilon(P^k) < +\infty)$, então (D^k) é consistente e, tal como anteriormente, também (D^k) tem solução óptima finita com $\upsilon(P^k) = \upsilon(D^k)$.

Note-se que esta técnica, de transformação do programa original, vulgarmente utilizada nos métodos de ponto interior (introduzidos por Karmarkar (1984)) nos quais, em geral, x^k corresponde a um n-uplo de componentes todas iguais a um, poderá não só aplicar-se ao

[3] Note-se que um subconjunto de $\mathbb{R}^n$ é compacto se e somente se é um subconjunto fechado e limitado.

[4] $\mathbb{R}_0^+$ identifica o conjunto de escalares não negativos.

método simplex, com x^k apresentando não mais do que m-1 componentes não nulas (desde que as colunas a que se referem estas componentes não nulas conjuntamente com a coluna $b\text{-}Ax^k$ sejam linearmente independentes) como também se pode a aplicar qualquer dos métodos intermédios para os quais a solução admissível corrente possa ter um número p de componentes não nulas, com $m \leq p \leq n$ (Cardoso e Clímaco (1992)).

Como é sabido para M suficientemente grande não só (D^k) é consistente como, consequentemente, se verifica que $\upsilon(P^k) = \upsilon(P) = \upsilon(D) = \upsilon(D^k)$.

Proposição 2.1

Seja y^* uma solução óptima para (D). Se $M > y^{*t}(Ax^k\text{-}b)$ então $\upsilon(P^k)$ é finito e para todo $\begin{bmatrix} x^{k*} \\ z^{k*} \end{bmatrix} \in X^*(P^k)$ tem-se que $z^{k*} = 0$.

Prova:

Uma vez que $y^* \in T$ e $y^{*t}(b\text{-}Ax^k) > \text{-}M$, segue-se que

$$y^{*t}[A, b\text{-}Ax^k] \geq [c^t, \text{-}M] \Leftrightarrow y^* \in T^k$$

e, uma vez que $\begin{bmatrix} x^k \\ 1 \end{bmatrix} \in S^k$ vem que tanto (P^k) como (D^k) são consistentes, pelo que ambos admitem solução óptima finita tal que $\upsilon(P^k) = \upsilon(D^k)$.

Sendo $\begin{bmatrix} x^{k*} \\ z^{k*} \end{bmatrix}$ uma solução óptima para (P^k) (e por conseguinte adimissível), vem que $Ax^{k*} + (b\text{-}Ax^k)z^{k*} = b$ e, uma vez que $T \supseteq T^k \Rightarrow \upsilon(P) = \upsilon(D) \leq \upsilon(D^k) = \upsilon(P^k)$, conclui-se que

$$y^*Ax^{k*} + y^*(b\text{-}Ax^k)\,z^{k*} = y^*b \leq c^tx^{k*} - Mz^{k*} \qquad (1)$$

Por outro lado, dado que $z^{k*} \geq 0$, da hipótese tira-se que $\text{-}Mz^{k*} \leq y^{*t}(b\text{-}Ax^k)z^{k*}$. Logo de (1) vem que

$$y^{*t}Ax^{k*} - Mz^{k*} \leq c^tx^{k*} - Mz^{k*} \Leftrightarrow y^{*t}Ax^{k*} \leq c^tx^{k*} \qquad (2)$$

Dado que $y^{*t}A \geq c^t$ e $x^{k*} \geq 0$, tira-se que $y^{*t}Ax^{k*} = c^tx^{k*}$. (3)

Assim, de (2) 2 (3) conclui-se que $y^{*t}Ax^{k*} = c^tx^{k*}$. Desta igualdade e de (1) vem finalmente que:

$$y^{*t}(b\text{-}Ax^k)z^{k*} \leq \text{-}Mz^{k*} \Rightarrow z^{k*} = 0 \text{ (uma vez que por hipótese } y^{*t}(b\text{-}Ax^k) > \text{-}M) \ \blacklozenge$$

A partir desta proposição é possível estimar um valor para M e bem ainda, conforme veremos, eventualmente adequá-lo, durante a aplicação do método, no caso desse valor ser inicialmente insuficiente.

Assim, supondo que, uma vez escolhido um dado ponto x^k cujas componentes são todas não negativas, aplicamos, para k = 0, 1,..., K, uma ou várias iterações de um dado método a cada um dos programas lineares (P^k), de modo a obter, a partir de $\begin{bmatrix} x^k \\ 1 \end{bmatrix}$, um novo ponto admissível para (P^k), $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix}$, tal que $0 \leq z^{k+1} < 1$, vem que:

$$[A, b\text{-}Ax^k]\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix} = b \Leftrightarrow Ax^{k+1} + z^{k+1}(b\text{-}Ax^k) = b$$

Definindo-se sucessivamente (P^{k+1}) a partir do ponto x^{k+1}, anteriormente determinado, obtém-se $Ax^{k+1} - b = z^{k+1}(Ax^k - b)$, para $k = 0, 1, \ldots, K$, donde se tira que

$$Ax^{K+1} - b = \left(\prod_{k=1}^{K+1} z^k\right)(Ax^0 - b) \Rightarrow \|Ax^{K+1} - b\| = \left(\prod_{k=1}^{K+1} z^k\right)\|Ax^0 - b\|.$$

Deste modo (supondo que $|z^k| \leq \varepsilon < 1, \forall k \in \mathbb{N}$) conclui-se que $\|Ax^{K+1} - b\| \to 0$ quando $K \to \infty$.

Sendo y^* uma solução óptima para (D) se $M > \|y^*\| \, \|Ax^0 - b\| \geq y^{*t}(Ax^0 - b)$, claramente que $M > y^{*t}(Ax^{k+1} - b)$. Por outro lado, no caso de $\|y^*\|$ ter sido mal avaliado, uma vez que $\|Ax^{k+1} - b\| \to 0$, a partir de determinado valor de K, M terá um valor suficiente grande para satisfazer as condições da proposição 2.1.

Um processo de acelarar a convergência de $\|Ax^k - b\|$ consiste em determinar, para cada k, de entre os múltiplos escalares positivos de x^k, Δx^k, aquele que minimiza $\|b - \Delta Ax^k\|$. Assim, sendo $\varphi_k(\Delta) = \|b - \Delta Ax^k\|^2 = \|b\|^2 - 2\Delta b^t Ax^k + \Delta^2 \|Ax^k\|^2$ e dado que $\varphi_k''(\Delta) = 2\|Ax^k\|^2 \geq 0$, claramente se tem que $\varphi_k(\Delta)$ é convexa, pelo que atinge o seu mínimo global para $\Delta = \Delta^k$ tal que $\varphi_k'(\Delta^k) = 0$, ou seja, para $\Delta^k = \dfrac{b^t Ax^k}{\|Ax^k\|^2}$. Deste modo, se após cada iteração, uma vez determinado x^k, se substituir x^k por $\Delta^k x^k (x^k \leftarrow \Delta^k x^k)$, conforme se esquematiza na figura 2.1, acelera-se a convergência do método e $b - A(\Delta^k x^k)$ vem ortogonal a $A(\Delta^k x^k)$.

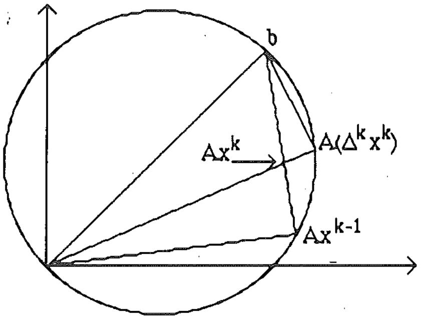


Fig. 2.1

Todavia, para que se possa substituir x^k por $\Delta^k x^k$, é necessário garantir que Δ^k seja positivo. A proposição 2.3, a seguir, garante-nos essa condição, desde que a primeira escolha, x^0, seja tal que $b^t Ax^0 > 0$.

Antes porém, deve observar-se que, sendo S não vazio e limitado, se $b = 0$ então $x = 0$ é a única solução admissível para (P) (se $\exists x \geq 0$ tal que $Ax = 0$ então $A(\mu x) = 0 \; \forall \mu > 0$) e toda esta análise deixa de fazer sentido. No caso de se ter $b \neq 0$, a determinação de $x^0 \geq 0$ tal que $b^t Ax^0 > 0$ é imediata, uma vez que, de acordo com a proposição 2.2, o vector $b^t A$ tem pelo menos uma componente positiva.

Proposição 2.2

Tendo em conta a definição de S, se $b^t A \leq 0$ então $b = 0$.

Prova:

Suponha-se que a condição $b^tA \leq 0$ (f) se verifica. Dado que $S \neq \varnothing$, existe $x \geq 0$ tal que $Ax = b$ (ff). Nestas condições de (f) obtém-se $b^tAx \leq 0$ e de (ff) vem que $b^tAx = \|b\|^2$, donde se conclui que $\|b\|^2 \leq 0$. Logo $b = 0$. ♦

Proposição 2.3

Se $x^k \in S^k$ é tal que $b^tAx^k > 0$ e $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix} \in S^k$ é tal que $0 < z^{k+1} \leq 1$, então $b^tAx^{k+1} > 0$.

Prova:

Dado que $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix} \in S^k$ vem que $Ax^{k+1} = b + (Ax^k - b)z^{k+1}$, logo tem-se que $b^tAx^{k+1} = b^tb + b^t(Ax^k - b)z^{k+1}$, ou seja, $b^tAx^{k+1} = \|b\|^2(1 - z^{k+1}) + b^tAx^kz^{k+1} > 0$, uma vez que $0 < z^{k+1} \leq 1$. ♦

3. Aplicação ao Método Simplex

Vamos supôr que se aplica o método simplex clássico à resoluçãão de (P^k) e que o quadro simplex reduzido associado à solução básica admissível corrente tem o aspecto:

	x_N	
x_B	$B_k^{-1}(x^k)N_k$	x_B^k
z		1
	$[c_B^t, -M]B_k^{-1}(x^k)N_k - c_N^t$	

Fig. 3.1

onde $B_k(x^k) = [\bar{B}_k, b - Ax^k]$ corresponde à submatriz de $[A, b - Ax^k]$ constituída pelas colunas associadas às variáveis básicas de $\begin{bmatrix} x^k \\ z^k \end{bmatrix}$ e N_k corresponde à submatriz de $[A, b - Ax^k]$ constituída pelas colunas associadas às variáveis não básicas de x^k.

Para se garantir que o ponto $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix}$ a iterar seja tal que $z^{k+1} < 1$ não só é necessário que haja uma efectiva mudança de vértice em S^k como também que a m-ésima componente da coluna pivotal $B_k^{-1}(x^k)N_ke_j$ (onde e_j identifica o j-ésimo vector da base canónica de $\mathbf{R}^q$ com $q=n-m+1$) seja positiva, i. é, se verifique a desigualdade

$$e_m^tB_k^{-1}(x^k)N_ke_j > 0$$

(onde e_m corresponde ao m-ésimo vector da base canónica de $\mathbf{R}^m$).

As proposições a seguir, não só nos garantem a existência de um índice j para o qual a desigualdade $e_m^tB_k^{-1}(x^k)N_ke_j > 0$ se verifica, como também nos indicam qual o mínimo valor a atribuir a M para que a coluna definida por esse índice j seja candidata a pivotal.

No que segue vamos identificar as componentes básicas de $\begin{bmatrix} x^k \\ z^k \end{bmatrix}$ por $\begin{bmatrix} x^k_B \\ z^k \end{bmatrix}$ e as não básicas por x^k_N e supor (para uma simples caracterização dos quadros simplex) que estas componentes estão ordenadas de modo a obter-se o ponto $\begin{bmatrix} x^k_B \\ z^k \\ x^k_N \end{bmatrix}$.

Proposição 3.1

Para o programa linear (P^k), dado o quadro simplex reduzido associado à solução básica admissível corrente, $\begin{bmatrix} x^k_B \\ 1 \\ 0 \end{bmatrix}$, existe j ($1 \le j \le n-m+1$) para o qual se verifica a inequação $e^t_m B^{-1}_k(x^k)N_k e_j > 0$, com e_m e e_j representado, respectivamente, o m-ésimo vector da base canónica de $\mathbf{R}^m$ e o j-ésimo vector da base canónica de $\mathbf{R}^q$ (onde $q=n-m+1$).

Prova:

Dado que as colunas da matriz $\begin{bmatrix} -B^{-1}_k(x^k)N_k \\ I_q \end{bmatrix}$ (onde I_q identifica a matriz identidade de ordem q) geram Ker($[A,b-Ax^k]$), para todo o ponto $\begin{bmatrix} x' \\ z \\ x'' \end{bmatrix}$ de S^k existe um q-uplo de escalares não negativos λ tal que

$$\begin{bmatrix} x' \\ z \\ x'' \end{bmatrix} = \begin{bmatrix} x^k_B \\ x^k \\ x^k_N \end{bmatrix} + \begin{bmatrix} -B^{-1}_k(x^k)N_k \\ I_q \end{bmatrix} \lambda. \qquad (f)$$

Suponha-se que não existe $j(1 \le j \le q)$ para o qual se tenha $e^t_m B^{-1}_k(x^k)N_k e_j > 0$, ou seja, as componentes da m-ésima linha de $B^{-1}_k(x^k)N_k$ são não positivas, ou ainda,

$$e^t_m B^{-1}_k(x^k)N_k \le 0 \qquad (ff)$$

Como consequência, de (f), tira-se que a m-ésima componente (componente associada á variável z) de qualquer dos pontos admissíveis para (P^k) verifica a condição

$$z = 1 - e^t_m B^{-1}_k(x^k)N_k \lambda > 0,$$

o que contraria a hipótese de $S \neq \varnothing$ (ou seja, de que existe $x \ge 0$ tal que $[A,b-Ax^k] \begin{bmatrix} x \\ 0 \end{bmatrix} = b$).

Logo, pode concluir-se que o sistema de inequações (ff) é impossível e que, por conseguinte, existe pelo menos um índice j para o qual $e^t_m B^{-1}_k(x^k)N_k e_j > 0$. ♦

Proposição 3.2

Seja o quadro simplex reduzido associado à solução básica admissível corrente $\begin{bmatrix} x^k_B \\ 1 \\ 0 \end{bmatrix}$, o representado na fig 3.1. Seja $\beta_j = e^t_m B^{-1}_k(x^k)N_k e_j > 0$ (com e_m e e_j identificando o m-ésimo e o j-ésimo vectores, respectivamente, da base canónica de $\mathbf{R}^m$ e $\mathbf{R}^q$ onde $q=n-m+1$), c^t_B o subvector de c^t associado às componentes bàsicas x^k_B e c^t_N o subvector de c^t associado às correspondentes componentes não básicas. Então a j-ésima coluna do referido quadro simplex é candidata a coluna pivotal sse $M \geq \frac{\alpha_j}{\beta_j}$ onde $\alpha_j = ([c_B^t\ 0]B_k^{-1}(x^k)N_k - c_N^t)e_j$.

Prova:

Uma vez que $[c_B^t, -M]$ e c_N^t são os subvectores do gradiente da f.o. associados, respectivamente, às componentes básicas e não básicas de $\begin{bmatrix} x_B^k \\ 1 \\ 0 \end{bmatrix}$, a linha de custos reduzidos vem definida pela equação

$$[c_B^t, -M]B_k^{-1}(x^k)N_k - c_N^t = [c_B^t\ 0]B_k^{-1}(x^k)N_k - c_N^t - Me_m^tB_k^{-1}(x^k)N_k$$

Nestas condições, sendo $\alpha_j = ([c_B^t\ 0]B_k^{-1}(x^k)N_k - c_N^t)e_j$ e $\beta_j = e_m^tB_k^{-1}(x^k)N_ke_j$, para que a j-ésima componente da linha de custos reduzidos do quadro simplex reduzido determine a j-ésima coluna deste mesmo quadro como coluna candidata a pivotal, é necessário e sufeciente que se verifique a desigualdade $\alpha_j - M\beta_j \leq 0 \Leftrightarrow M \geq \frac{\alpha_j}{\beta_j}$, uma vez que, por hipótese, $\beta_j > 0$. ♦

Embora a estratégia sugerida pelas proposições anteriores, por si só, nos permita chegar a uma solução óptima $\begin{bmatrix} x^{k*} \\ z^{k*} \end{bmatrix}$ de (P^k), tal que $z^{k*} = 0$,a actualização de $[A,b\text{-}Ax^k]$ para $[A,b\text{-}Ax^{k+1}]$, como se vair ver, não oferece qualquer dificuldade.

Com efeito, suponha-se que $B_{k+1}(x^k) = [\bar{B}_{k+1},b\text{-}Ax^k]$ e $B_{k+1}(x^{k+1}) = [\bar{B}_{k+1},b\text{-}Ax^k]$ correspondem, respectivamente, às submatrizes de $[A,b\text{-}Ax^k]$ e $[A,b\text{-}Ax^{k+1}]$ constituídas pelas colunas associadas às variáveis básicas de $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix}$ com $z^{k+1} > 0$. Uma vez que se verifica a igualdade $b\text{-}Ax^{k+1} = z^{k+1}(b\text{-}Ax^k)$, sendo D_{k+1} a matriz diagonal de ordem m

$$D_{k+1} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & \ldots & 0 & z^{k+1} \end{bmatrix}$$

vem que $B_{k+1}(x^{k+1}) = [\bar{B}_{k+1},b\text{-}Ax^{k+1}] = [\bar{B}_{k+1},b\text{-}Ax^k]D_{k+1}$, donde se pode concluir que

$$B_{k+1}^{-1}(x^{k+1}) = [\bar{B}_{k+1},b\text{-}Ax^{k+1}]^{-1} = D_{k+1}^{-1}\,[\bar{B}_{k+1},b\text{-}Ax^k]^{-1} = D_{k+1}^{-1}B_{k+1}^{-1}(x^k)$$

Por outro lado, relativamente à linha de custos reduzidos, sendo c_B^t o subvector de c^t associado às variáveis básicas de x^{k+1} (ou seja, c_B^t não considera o coeficiente -M, relativo à variável artificial z), obtém-se

$$[c_B^t,-M]B_{k+1}^{-1}(x^{k+1})N_{k+1}-c_N =$$

$$= [c_B^t,-M]D_{k+1}^{-1}B_{k+1}^{-1}(x^k)N_{k+1}-c_N$$

$$= [c_B^t,0]B_{k+1}^{-1}(x^k)N_{k+1} - \frac{M}{z^{k+1}}\, e_m^t B_{k+1}^{-1}(x^k)N_{k+1}c_N$$

$$= [c_B^t,0]B_{k+1}^{-1}(x^k)N_{k+1} - Me_m^t B_{k+1}^{-1}(x^k)N_{k+1} - c_N - (\frac{M}{z^{k+1}} - M)e_m^t B_{k+1}^{-1}(x^k)N_{k+1}$$

$$= [c_B^t,-M]B_{k+1}^{-1}(x^k)N_{k+1} - c_N - M\,\frac{1-z^{k+1}}{z^{k+1}}\, e_m^t B_{k+1}^{-1}(x^k)N_{k+1},$$

com e_m identificando o m-ésimo vector da base canónica de $\mathbb{R}^m$, pelo que $e_m^t B_{k+1}^{-1}(x^k)N_{k+1}$ corresponde à m-ésima linha do quadro simplex reduzido (1) a seguir representado.

Assim, supondo que se dispõe do quadro simplex reduzido associado a $B_{k+1}(x^k)$, a determinação do quadro simplex reduzido associado a $B_{k+1}(x^{k+1})$ faz-se dividindo por z^{k+1} a linha correspondente à variável artificial z e adicionando a linha assim obtida multiplicada por $M(z^{k+1} - 1)$ à respectiva linha de custos reduzidos.

	x_N	
x_B	$B_{k+1}^{-1}(x^k)N_{k+1}$	x_B^{k+1}
z		z^{k+1}
	$[c_B^t, -M]B_{k+1}^{-1}(x^k)N_{k+1} - c_N^t$	

(1)

$\longrightarrow$

	x_N	
x_B	$B_{k+1}^{-1}(x^{k+1})N_{k+1}$	x_B^{k+1}
z		1
	$[c_B^t, -M]B_{k+1}^{-1}(x^{k+1})N_{k+1} - c_N^t$	

(2)

Fig. 3.2

No que se refere à substituição de x^{k+1} por $\Delta^{k+1}x^{k+1}$ (com Δ^{k+1} determinado como anteriormente se referiu), dado que

$$B_{k+1}(\Delta^{k+1}x^{k+1}) = B_{k+1}(x^k)\, D_{k+1} - \delta Ax^{k+1}e_m^t,$$

onde $\delta = \Delta^{k+1} - 1$ [5], a actualização da inversa de $B_{k+1}(\Delta^{k+1}x^{k+1})$ pode fazer-se, sem grande esforço computacional, com recurso à fórmula de Sherman-Morrison-Woodbury (veja-se Goldfarb e Todd (1989)), com a qual, sendo Q uma matriz invertível e u e v dois vectores arbitrários tais que $1+v^tQ^{-1}u\neq0$, se obtém $[Q+uv^t]^{-1} = Q^{-1} - \frac{1}{1+v^tQ^{-1}u}\,Q^{-1}uv^tQ^{-1}$.

Com efeito, sendo $B = B_{k+1}(x^k)$, $D = D_{k+1}$, $u = Ax^{k+1}$ e $v = -\delta e_m$, vem que $B_{k+1}(\Delta^{k+1}x^{k+1}) = BD+uv^t$. Desta equação matricial decorre que

$$B_{k+1}^{-1}(\Delta^{k+1}x^{k+1}) = D^{-1}B^{-1} - D^{-1}B^{-1}uv^tD^{-1}B^{-1} \qquad (fff)$$

uma vez que, conforme a seguir se prova, $1+v^tD^{-1}B^{-1}u = 1-\delta e_m^tD^{-1}B^{-1}Ax^{k+1} = 1$. De facto, $e_m^tD^{-1}B^{-1}Ax^{k+1} = e_m^tD^{-1}B^{-1}(b-z^{k+1}(b-Ax^k))$ (uma vez que $Ax^{k+1}=b-z^{k+1}(b-Ax^k)$)

[5] Note-se que, para $\Delta^{k+1}=1+\delta$, $b-A(\Delta^{k+1}x^{k+1})=b-Ax^{k+1}-\delta Ax^{k+1}=z^{k+1}(b-Ax^k)-\delta Ax^{k+1}$.

$$= e_m^t\left(\begin{bmatrix} x_B^{k+1} \\ 1 \end{bmatrix} - e_m\right)$$

$$\text{(note-se que } D^{-1}B^{-1}b = \begin{bmatrix} x_B^{k+1} \\ 1 \end{bmatrix} \text{ e } D^{-1}B^{-1}(b-Ax^k) = \frac{1}{z^{k+1}}\, e_m))$$

$$= 0.$$

Por outro lado, desenvolvendo o segundo membro da equação (fff) vem que

$$B_{k+1}^{-1}(\Delta^{k+1}x^{k+1}) = D^{-1}B^{-1} + \delta D^{-1}B^{-1}Ax^{k+1}e_m^tD^{-1}B^{-1}$$

$$= D^{-1}B^{-1} + \delta D^{-1}B^{-1}(b\text{-}z^{k+1}(b\text{-}Ax^k))e_m^tD^{-1}B^{-1}$$

$$= D^{-1}B^{-1} + \delta\left(\begin{bmatrix} x_B^{k+1} \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)e_m^tD^{-1}B^{-1}$$

$$= (I+\delta\begin{bmatrix} x_B^{k+1} \\ 0 \end{bmatrix} e_m^t)D^{-1}B^{-1}$$

$$= \begin{bmatrix} 1 & 0 & \dots & 0 & \delta x_{B_1}^{k+1} \\ 0 & 1 & \dots & 0 & \delta x_{B_2}^{k+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \delta x_{B_{m-1}}^{k+1} \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} D^{-1}B^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & \frac{\delta}{z^{k+1}}x_{B_1}^{k+1} \\ 0 & 1 & \dots & 0 & \frac{\delta}{z^{k+1}}x_{B_2}^{k+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \frac{\delta}{z^{k+1}}x_{B_{m-1}}^{k+1} \\ 0 & 0 & \dots & 0 & \frac{1}{z^{k+1}} \end{bmatrix} B^{-1}$$

Assim, quando se passa de x^{k+1} para $\Delta^{k+1}x^{k+1}$, a i-ésima linha da inversa da matriz $\bar{B}_{k+1}(\Delta^{k+1}x^{k+1})$, para i=1, 2,..., m-1, vem dada por

$$e_i^tB_{k+1}^{-1}(\Delta^{k+1}x^{k+1}) = e_i^tB^{-1} + \frac{\delta}{z^{k+1}}\, x_{B_i}^{k+1}e_m^tB^{-1}$$

e a m-ésima por $e_m^tB_{k+1}^{-1}(\Delta^{k+1}x^{k+1}) = \frac{1}{z^{k+1}}\, e_m^tB^{-1}$.

Como consequência, a actualização do quadro simplex reduzido, no que diz respeito à determinação de $B_{k+1}^{-1}(\Delta^{k+1}x^{k+1})N_k$, faz-se acrescentando a cada uma das i-ésimas linhas (i≠m) de $B^{-1}N_k$ a m-ésima multiplicada por $\frac{\delta}{z^{k+1}}\, x_{B_i}^{k+1}$ e dividindo a m-ésima linha por z^{k+1}.

Relativamente à linha de custos reduzidos obtém-se

$$[c_B^t,\text{-}M]B_{k+1}^{-1}(\Delta^{k+1}x^{k+1})N_{k+1}-c_N^t = [c_B^t,\text{-}M](I+\delta\begin{bmatrix} x_B^{k+1} \\ 0 \end{bmatrix}e_m^t)D^{-1}B^{-1}N_{k+1}-c_N^t$$

$$= [c_B^t, \frac{\delta}{z_{B_1}^{k+1}}\, c_B^tx_B^{k+1} - \frac{M}{z^{k+1}}]B^{-1}N_{k+1}-c_N^t$$

$$= [c_B^t, -M + \frac{\delta}{z_{B_1}^{k+1}} c_B^t x_B^{k+1} - M \frac{1-z^{k+1}}{z^{k+1}}] B^{-1} N_{k+1} c_N^t$$

$$= [c_B^t, -M] B^{-1} N_{k+1} - c_N^t + (\frac{\delta}{z^{k+1}} c_B^t x_B^{k+1} - M \frac{1-z^{k+1}}{z^{k+1}}) e_m^t B^{-1} N_{k+1},$$

pelo que, a sua actualização é feita pela adição da m-ésima linha de $B^{-1}N_{k+1}$, previamente multiplicada por $\frac{\delta}{z^{k+1}} c_B^t x_B^{k+1} - M \frac{1-z^{k+1}}{z^{k+1}}$, à linha de custos reduzidos $[c_B^t, -M]B^{-1}N_{k+1} - c_N^t$.

4. Exemplo Numérico

Segue-se um exemplo numérico muito simples, no qual se pretende ilustrar a aplicação das técnicas de correcção de M associadas aos resultados apresentados.

Considere-se o programa linear

$$\max\{-2x_1+x_2 : x_1, x_2 \in \mathbf{R}, x_1-2x_2 \leq 5, x_2 \leq 5, x_1+x_2 \geq 1, x_1 \geq 0, x_2 \geq 0\}.$$

Após a introdução das variáveis de desvio e da variável artificial z (à qual vamos associar, na função objectivo, o coeficiente -M = -2), tomando-se para solução básica admissível inicial o ponto $\begin{bmatrix} x^0 \\ z^0 \end{bmatrix}^t = [1,0,0,0,1,1]$ obtém-se

$$\max\{c^t x - 2z : \begin{bmatrix} x \\ z \end{bmatrix} \in S^0\},$$

onde $S^0 = \{\begin{bmatrix} x \\ z \end{bmatrix} \in \mathbf{R}^6, [A, b-Ax^0] \begin{bmatrix} x \\ z \end{bmatrix} = b, x \geq 0, z \geq 0\}$,

$$c^t = [-2,1,0,0,0], [A, b-Ax^0] = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & -1 & 1 \end{bmatrix} \text{ e } b = \begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix}.$$

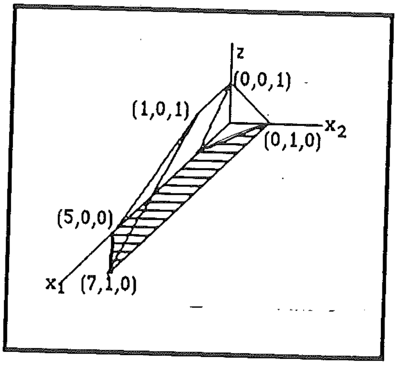


Fig. 4.1

Na figura 4.1 para além da região admissível do problema inicial (convenientemente assinalada) representa-se a região admissível obtida, para o problema modificado, com a introdução da variável artificial z e a correspondente coluna $b-Ax^0$.

Nos quadros simplex reduzidos a seguir representados (os quais, para cada solução básica $\begin{bmatrix} x^k \\ z^k \end{bmatrix}$ e cada valor de M, denotaremos por $Q^k(M)$) vamos utilizar duas linhas de custos reduzidos associadas, respectivamente, aos gradientes $[c^t,0]$ e $[c^t,-M]$ das funções objectivo original e modificada.

Embora a solução básica admissível inicial, $\begin{bmatrix} x^0 \\ z^0 \end{bmatrix}^t = [1,0,0,0,1,1]$, tenha associado o quadro simplex reduzido:

Q^0 (2)	x_2	x_3	x_4	
x_1	-6	1	-4	1
x_5	-6	1	-4	1
z	1	0	1	1
$[c_B,0]B^{-1}$ N-c_N	11	-2	8	-2
$[C_B,-M]B^{-1}$ N-c_N	9	-2	6	-4

a determinação de Δ^0 (e consequentemente de Δ^0x^0) de modo a minimizar-se $\|b-\Delta Ax^0\|$ implica a obtenção do valor $\Delta^0 = 5$ [6] e a correspondente actualização do quadro $Q^0(2)$ que, de acordo com os desenvolvimentos referidos anteriormente, toma o aspecto:

Q^0 (2)	x_2	x_3	x_4	
x_1	-2	1	0	5
x_5	-2	1	1	5
z	1	0	1	1
$[c_B,0]B^{-1}$ N-c_N	3	-2	0	-10
$[C_B,-M]B^{-1}$ N-c_N	1	-2	-2	-12

Da análise da linha de $Q^0(2)$ associada à variável básica z decorre que $\beta_2=1$ (o coeficiente correspondente à coluna não básica associada a x_2 é positivo) e $\beta_4=1$ (o coeficiente correspondente à coluna não básica associada a x_4 é positivo) pelo que, tendo em vista a redução da variável artificial z, se torna conveniente que a coluna pivotal seja ou a coluna associada a x_2 ou a coluna associada a x_4.

Embora com actual valor de M, a coluna associada a x_4 seja candidata a pivotal, para exemplificação da aplicação dos procedimentos de actualização de M, vamos supor que, por algum motivo, estamos interessados que a coluna associada à variável não básica x_2 seja a coluna pivotal.

[6] Note-se que $\Delta^0 = \dfrac{b^tAx^0}{\|Ax^0\|^2}$.

De acordo com os resultados introduzidos (tendo em conta que $\frac{\alpha_2}{\beta_2} = \frac{3}{1}$) basta fazer M = 4 ($M > \frac{\alpha_2}{\beta_2}$) para que a coluna associada a x_2 passe a ser a candidata a pivotal. Deste modo obtém-se $[c_B,-M]B^{-1}N-c_N = [\alpha_2,-2,0]-M[\beta_2,0,1] = [-1,-2,-4]$, ou seja, obtém-se o quadro simplex reduzido:

Q^0 (4)	x_2	x_3	x_4	
x_1	-2	1	0	5
x_5	-2	1	1	5
z	**1**	0	1	1
$[c_B,0]B^{-1}$ N-c_N	3	-2	0	-10
$[C_B,-M]B^{-1}$ N-c_N	-1	-2	-4	-14

Tomando-se para coluna pivotal a coluna associada a x_2, determina-se como pivô a entrada do quadro assinalada a negro, o que obriga a que seja a variável z a sair da base. Nestas condições, a presença da variável artificial z, nos quadros subsequentes, torna-se desnecessária e, após duas iterações (contadas a partir de $Q^0(4)$), encontra-se a solução óptima $\begin{bmatrix} x_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

5. Aplicação aos Métodos de Ponto Interior

No que se refere aos métodos de ponto interior a actualização de $b-Ax^k$ e a substituição de x^k por $\Delta^k x^k$ exigem praticamente o mesmo esforço computacional. Com efeito, sendo $D(x^k,z^k)$ a matriz diagonal cujos elementos principais são as componentes de $\begin{bmatrix} x^k \\ z^k \end{bmatrix} > 0$, a operação de "scaling", inerente a qualquer dos métodos de ponto interior, obriga, de iteração para iteração, à determinação da matriz de projecção sobre o subespaço nulo de $[A,b-Ax^k]D(x^k,z^k)$, ou seja, sendo $B^k = [A,b-Ax^k]D(x^k,z^k)$, a k-ésima iteração obriga à determinação da matriz de projecção

$$P_k = I - B_k^t(B_kB_k^t)^{-1}B_k$$

determinação essa que é, claramente, a que mais contribui para o elevado volume de cálculo associado, nestes métodos, a cada iteração. De um modo alternativo, a determinação de P_k pode ser feita a partir de uma matrix U_k cujas colunas constituem uma base para o subespaço vectorial $Ker([A,b-Ax^k])$. Com efeito, uma vez que $B_kU_k = 0$ e $[B_k^t,U_k]$ é uma matriz quadrada não singular, a matriz de projecção em $Ker([A,b-Ax^k])$ pode ser dada por

$$P_k = U_k(U_k^tU_k)^{-1}U_k^t$$

Nestas condições a determinação da projecção em $Ker([A,b-Ax^k])$ do gradiente da função objectivo de (P^k) $([c^t,-M])$ obtém-se fazendo

$$P_k\begin{bmatrix} c \\ -M \end{bmatrix} = U_k(U_k^tU_k)^{-1}U_k^t\begin{bmatrix} c \\ -M \end{bmatrix},$$

ou então, determina-se y = $(U_k^tU_k)^{-1}U_k^t\begin{bmatrix}c\\-M\end{bmatrix}$ (o que equivale a resolver o sistema $U_k^tU_ky = U_k^t\begin{bmatrix}c\\-M\end{bmatrix}$) e faz-se $P_k = \begin{bmatrix}c\\-M\end{bmatrix}U_ky$.

Relativamente à determinação de uma base para Ker([A,b-Ax^k]) cujos vectores formem as colunas de U_k, uma vez que, a matriz A pode tomar a forma A = [A',I_m], com A'$\in \mathbb{R}^{m\times p}$ (p=n-m) e I_m identificando a matriz identidade de ordem m, fazendo-se

$$u_j = \begin{bmatrix}e_j\\a_j\end{bmatrix} \text{ para } j=1, 2,\ldots, p$$

onde e_j representa o j-ésimo vector da base canónica de $\mathbb{R}^p$ e a_j a j-ésima coluna de A (a qual coincide com a j-ésima coluna de A' para $1\leq j\leq 0$), claramente se verifica que, não só estes p vectores pertencem ao subespaço nulo de A (note-se que $\forall j\in\{1, 2,\ldots, p\}$ $Au_j = a_j - a_j = 0$) como são linearmente independentes, pelo que constituem uma base para Ker(A). Deste modo, representando-se por u_j' (para j=1, 2,..., p+1) os vectores coluna de U_k, os quais constituem uma base para Ker([A,b-Ax^k]) (note-se que o subespaço nulo desta matriz tem dimensão p+1), a sua determinação obtém-se fazendo

$$u_j' = \begin{bmatrix}u_j\\0\end{bmatrix} \text{ para } j=1, 2,\ldots, p$$

e ainda, tendo em conta que, para A = [A',I_m], tanto $\begin{bmatrix}x^k\\1\end{bmatrix}$ como $\begin{bmatrix}\mathbf{0}\\b\\0\end{bmatrix}$, com $\mathbf{0}\in\mathbb{R}^p$, são soluções do sistema $[A,b-Ax^k]\begin{bmatrix}x\\z\end{bmatrix} = b$, fazendo-se

$$u_{p+1}' = \begin{bmatrix}x^k\\1\end{bmatrix} - \begin{bmatrix}\mathbf{0}\\b\\0\end{bmatrix}$$

Embora, para o caso do método simplex, já se tenha referido um critério que garante o decréscimo, em cada iteração, da variável artificial z, o qual é suficiente para a obtenção de uma sucessão de pontos convergentes para uma solução óptima de (P), de um modo mais geral, independentemente do método que se pretenda aplicar, para se garantir esta convergência, basta complementar a actualização de [A,b-Ax^k], anteriormente referida, (o que nos métodos de ponto interior equivale a actualizar u_{p+1}') com uma eventual correcção de M baseada na seguinte proposição:

Proposição 5.1

Seja $\begin{bmatrix}x\\z\end{bmatrix}$ uma solução admissível para (P^k) ($k\in\mathbb{N}$) tal que $z\geq 1$ e seja $\tau\in\mathbb{R}$ tal que $\upsilon(P) + \tau \geq 0$. Se o número positivo M é tal que $-M + \tau < -c^tx$, então $\begin{bmatrix}x\\z\end{bmatrix}$ não é solução óptima para (P^k).

Prova:

Uma vez que sendo x* uma solução óptima para (P), $\begin{bmatrix}x^*\\0\end{bmatrix}$ é admissível para (P^k), sabe-se que $\upsilon(P^k) + \tau \geq \upsilon(P) + \tau \geq 0$.

Dado que $z \geq 1$, vem que $-Mz+\tau \leq -M+\tau < -c^t x \Rightarrow c^t x - Mz + \tau < 0$. Logo $\begin{bmatrix} x \\ z \end{bmatrix}$ não é solução óptima para (P^k). ♦

A partir do momento que se obtenha uma solução $\begin{bmatrix} x^{k+1} \\ z^{k+1} \end{bmatrix}$ admissível para (P^k) tal que $z^{k+1} = 0$, é claro que a variável artificial z pode passar a ser ignorada e tudo se resume, a partir daí, à determinação de uma solução óptima para (P).

Agradecimentos

Os autores desejam agradecer as correcções e sugestões efectuadas por um "referee" anónimo, por as considerarem de grande utilidade para a melhoria da versão inicial deste artigo.

Referências

[1] Cardoso, D.M., Clímaco, J.N., The Generalized Simplex Method, Operations Research Letters 12 (1992) 337-348.

[2] Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, New Jersey, 1963.

[3] Goldfarb, D. e Todd, M.J., Linear Programming, em Handbooks in Operations Research and Management Science, Vol. 1 Optimization (1989) 73-170, ed. G.LK. Nemhauser, A.H.Rinnoy Kan, M.J.Todd.

[4] Karmarkar, N.K., A New Polynomial Time Algorithm for Linear Programming, Combinatorica 4 (1984) 373-395.

[5] Khachiyan, L.G., A polynomial algorithm in linear programming, Soviet Mathematics Doklady 20 (1979) 191-194.

[6] Kojima, M., Mizuno, S. e Yoshise, A., A Litle Theorem of the Big M in Interior Point Algorithms, Mathematical Programming 59 (1993) 361-375.

[7] Ishihara, T., Kojima, M., On the big M in the affine scaling algorithm, Mathematical Programming 62 (1993) 85-93.

[8] Shor, N.Z., Utilization of the operation of the space dilatation in the minimization of convex functions, (em Russo) Kibernetica 1 (1970) 6-12, Tradução Inglesa: Cybernetics 6, 7-15.

[9] Terlaky, T., A Convergent Criss-Cross Method, Optimization 16 (1985) 683-690.

[10] Terlaky, T., A Finite Criss-Cross Method for Oriented Matroids, Journal of Combinatorial Theory (Ser. B) 42 (1987) 319-327.

[11] Yudin, D.B. e Nemirovskii, A.S., Informational complexity and efficient methods for the solution of convex extremal problems, (em Russo) Ékonomika i Mathematicheskie Metody 12 (1976) 357-369; Tradução Inglesa: Matekon 13 3-25.

INVESTIGAÇÃO OPERACIONAL, vol. 14, n.° 1 (Junho 1994) 77-90

HEURÍSTICAS PARA A PROGRAMAÇÃO DE OPERAÇÕES EM AMBIENTE DE "JOB-SHOP"

Manuel Pina Marques
José Fernando Gonçalves
Faculdade de Engenharia da Universidade do Porto
GEIN-Departamento de Engenharia Mecânica e Gestão Industrial
Rua dos Bragas
4099 Porto Codex

Abstract

The scheduling of operations in industrial environment is often modelled as a job-shop problem. However, for this type of problems there are no efficient algorithms to obtain the optimal solution, unless the shop has only one or two machines. The absence of optimization tools often leads to the use of heuristics in the search of the problem solution.

In this paper two new heuristics are presented for the generation of schedules. Both heuristics aim to minimize the largest tardiness, and allow the user to define the sequencing of the operations processed in bottleneck machines. The heuristics automatically schedule all the others operations. The job-shop problem is decomposed in single machine scheduling problems, wich are solved by the Cartier algorithm or by the Schrage heuristic.

Tests performed on 100 problems and in a pilot industrial company demonstrated that the purposed heuristics provides good schedules and flexibility, and can substitute with success the manual and time consuming processes that are still used for sheduling industrial operations.

Resumo

O "Job-Shop" é um problema do tipo NP-difícil, pelo que o recurso a algoritmos de optimização para a sua resolução é bastante limitado. Com efeito, a sua aplicação a problemas de dimensão industrial é praticamente nula devido, por um lado, ao reduzido número de máquinas consideradas e, por outro, ao facto da função objectivo adoptada ser frequentemente diferente da real.

Neste artigo apresentam-se duas heurísticas para a programação de operações em ambiente de "job-shop". Ambas as heurísticas procuram minimizar o maior dos atrasos das ordens de fabrico relativamente às datas devidas de entrega respectivas, apresentando a particulariedade de permitirem ao agente de planeamento definir um conjunto de máquinas críticas (gargalos). A sequência de processamento das operações realizadas nestas máquinas pode ser fixada pelo agente de planeamento.

A programação das operações processadas nas restantes máquinas é feita pelas heurísticas de uma forma automática. As heurísticas decompõem o problema global em sub-problemas de sequenciamento de uma só máquina, utilizando para a sua resolução o algoritmo de Carlier ou a heurística de Schrage. As máquinas são sucessivamente tratadas, seleccionando-se em primeiro lugar a máquina mais crítica. As heurísticas diferem no procedimento adoptado para a determinação dessa máquina.

Apresentam-se os resultados obtidos pelas heurísticas na resolução de um conjunto de problemas gerados aleatoriamente. Faz-se a comparação desses resultados com os que se obtêm pela aplicação de um conjunto de regras de sequenciamento mais frequentemente utilizadas na indústria.

Keywords

Job-shop, scheduling, heuristics

Notação

$O_{i,j}$ - Operação *j* da ordem de fabrico *i*.

a_i - Data a partir da qual a ordem de fabrico *i* pode ser lançada em fabrico

$a_{i,j}$ - Data a partir da qual a operação $O_{i,j}$ está disponível para processamento.

$d_{i,j}$ - Duração da operação $O_{i,j}$.

$t_{i,j}$ - Data de início de processamento da operação $O_{i,j}$.

$q_{i,j}$ - Tempo de permanência na oficina da ordem de fabrico no sistema após o processamento da operação $O_{i,j}$.

D_i - Data devida de entrega para a ordem de fabrico *i*.

$D^k_{i_{min}}$ - Data mínima possível de entrega para a ordem de fabrico *i*, calculada a partir da calendarização das operações processadas na máquina m_k.

I_1 - Conjunto das máquinas para as quais já foi fixada a sequência de processamento das operações.

I_0 - Conjunto das máquinas para as quais ainda não foi fixada a sequência de processamento das operações.

N - Número total de ordens de fabrico.

M - Número total de máquinas. Quando utilizado como índice, representa a última operação da ordem de fabrico. Por exemplo, $t_{i,M}$ representa a data de início de processamento da última operação da ordem de fabrico *i*.

T_{max} - Valor do maior dos atrasos associado a um determinado plano de fabrico.

Introdução

As necessidades de fabrico numa empresa industrial são frequentemente traduzidas através de um conjunto de ordens de fabrico. Cada ordem de fabrico diz respeito a um produto e tem associada uma data devida de entrega. A cada produto corresponde uma rota distinta na oficina, isto é, a ordem pela qual o produto visita as máquinas é diferente de produto para produto. Os tempos de processamento em cada posto de trabalho e para cada um dos produtos são conhecidos e considerados determinísticos.

O problema do "job-shop" consiste na definição de um plano para o fabrico de um conjunto de ordens de fabrico cujas operações são processadas nas diferentes máquinas da oficina. Esse plano deverá ser óptimo relativamente a uma função objectivo e verificar as restrições de precedência entre operações da mesma ordem de fabrico.

O "job-shop" é um problema de resolução reconhecidamente complexa dada a sua natureza combinatória. Diferentes autores demonstraram que determinados tipos de problemas de programação de operações são NP- dificeis como, por exemplo, Lenstra et al.[1977], Graham et al.[1979] e Garey e Johnson [1979]. A maioria destes trabalhos foi revista por Rinnooy

[1976]. A inexistência de métodos de optimização eficientes para o tratamento de problemas de dimensão industrial leva a concluir que é praticamente inevitável a utilização de métodos heurísticos na resolução deste tipo de problemas.

Neste artigo apresentam-se duas novas heurísticas para a programação de operações em ambiente de "job-shop", designadas por PRO-I e PRO-II (PRO-Partial Resourse Optimization). Estas heurísticas decompõem o problema global em sub-problemas de sequenciamento de operações numa só máquina, utilizando para a sua resolução o algoritmo de Carlier [1982] (até à optimalidade) ou, alternativamente, a heurística de Schrage (de uma forma aproximada). A resolução de cada sub-problema consiste na definição de uma sequência de processamento para um conjunto de operações (ou tarefas) numa única máquina, sendo definidas para cada operação uma data mínima para o início do seu processamento e uma data devida de entrega.

O objectivo de ambas as heurísticas é o de minimizar a maior diferença positiva entre as datas de conclusão das ordens de fabrico e as respectivas datas devidas de entrega (isto é, minimizar o maior dos atrasos). As heurísticas resolvem sucessivamente o problema do sequenciamento das operações realizadas em cada uma das máquinas, começando pela máquina mais crítica. As heurísticas PRO-I e PRO-II diferem somente no procedimento de selecção da próxima máquina a sequenciar (ou seja, na determinação da máquina mais crítica).

Por vezes, os agentes de planeamento são capazes de identificar as máquinas que se comportam como gargalos ao processo produtivo (máquinas críticas). Quando tal acontece, e dado o conhecimento específico que aqueles agentes têm do problema em estudo, seria vantajoso que concentrassem a sua atenção na definição da sequência de processamento das operações realizadas nessas máquinas. Surge assim a necessidade de dispôr de heurísticas que, tendo em conta as sequências pré-definidas pelos utilizadores para as máquinas críticas, possam gerar o sequenciamento das operações processadas nas máquinas não críticas. Como se verá, as heurísticas PRO-I e PRO-II obedecem a esta filosofia de utilização.

Sequenciamento de tarefas numa só máquina

Os problemas de sequenciàmento de um conjunto de N tarefas independentes numa oficina composta por uma só máquina constituem um conjunto particular de problemas. A máquina só pode processar uma tarefa de cada vez . A tarefa *i* está disponivel para ser processada a partir do instante a_i ("release date"), tem um tempo de processamento d_i e um determinado tempo de permanência no sistema, após ter sido processada, q_i ("tail"). Supõe-se que o processamento de uma tarefa, depois de iniciado, não pode ser interrompido.

Se a função objectivo for a minimização do "makespan", o problema é NP-dificil (Garey e Johnson [1979]). Para a sua resolução, vários autores utilizam a técnica do "branch and bound", tais como Baker e Su [1974], McMahon e Florian [1975], Larson et al.[1985], Grabowski et al.[1986] e Cartier [1982]. Todos eles utilizam a estrutura do caminho crítico no procedimento de "branching" adoptado.

de Carlier (ou em alternativa a heurística de Schrage), a fim de se obter um plano provisório. Para a aplicação do algoritmo (ou da heurística de Schrage), é necessário definir um conjunto de parâmetros $a_{i,j}$ e $q_{i,j}$, para cada operação $O_{i,j}$ processada na máquina $m_k \in I_0$.

Determinação dos valores $a_{i,j}$

Como já foi referido, $a_{i,j}$ representa a data a partir da qual a operação $O_{i,j}$ está disponível para ser processada. O procedimento para o seu cálculo é o seguinte:

Para cada máquina $m_k \in I_0$:

Para cada operação $O_{i,j}$ processada na máquina m_k:

Se já foi fixada a data de início de alguma operação da ordem de fabrico *i* com índice inferior a *j*, então:

(seja *p* o maior dos índices das operações da ordem de fabrico *i* com data de início já fixada, com p<j)

$$a_{i,j} = t_{i,p} + \sum_{z=p}^{j-1} d_{i,z} \qquad (1)$$

Senão:

$$a_{i,j} = a_i + \sum_{z=1}^{j-1} d_{i,z} \qquad (2)$$

Determinação dos valores $q_{i,j}$

O algoritmo de Carlier utiliza tempos de permanência no sistemaa das diferentes ordens de fabrico, q_i, e minimiza o valor do "makespan". Nos problemas industriais, no entanto, definem-se geralmente as datas devidas de entrega para as várias ordens de fabrico, e utiliza-se como medida de desempenho os desvios verificados relativamente àquelas datas. A fim de transformar o problema em estudo num problema com a forma utilizada pelo algoritmo de Carlier [1982], basta converter as datas devidas de entrega das diferentes ordens de fabrico, D_i, em tempos de permanência no sistema após processamento, q_i, da seguinte forma (pode-se encontrar em Carlier [1987] a justificação para esta transformação):

$$q_i = D_{max} - D_i, \quad \text{em que}$$
$$D_{max} = \max_{i \in I} D_i \qquad (3)$$

em que *I* representa o conjunto de ordens de fabrico com operações a serem processadas na máquina em estudo.

Como se pretendem tratar problemas envolvendo oficinas com mais do que uma máquina, é necessário adaptar a expressão (3) a esta nova situação. Vai-se designar por D^k_{max} o valor de D_{max} associado à máquina m_k. Para a determinação de D^k_{max} é necessário ter em conta as datas mínimas possíveis de entrega das diferentes ordens de fabrico, D^M_i, tendo em conta as operações já sequenciadas noutras máquinas. O procedimento para a determinação de D^k_{max} será então:

Para cada operação $O_{i,j}$ processada na máquina m_k:

Se já foi fixada a data de início de alguma operação da ordem de fabrico *i* com índice inferior a *j*, então:

(seja *p* o maior dos índices das operações da ordem de fabrico *i* com data de início já fixada, com p<j)

$$D_i^M = \max[D_i, t_{i,p} + \sum_{z=p}^{M} d_{i,z}] \tag{4}$$

Caso contrário:

$$D_i^M = \max[D_i, a_i + \sum_{z=1}^{M} d_{i,z}] \tag{5}$$

O valor de D_{max}^k para a máquina m_k será então dado pelo maior valor de D_i^M, isto é:

$$D_{max}^k = \max_{i \in I} D_i^M \tag{6}$$

Após a determinação do valor de D_{max}^k, podem-se calcular os valores de $q_{i,j}$ do seguinte modo:

Para cada máquina $m_k \in I_0$:

Para cada operação $O_{i,j}$ processada nessa máquina;

Se já fixada a data de início de alguma operação da ordem de fabrico *i* com índice superior a *j*, então:

(Seja *p* o menor dos índices das operações da ordem de fabrico *i* com data de início já fixada, com p>j)

$$q_{i,j} = D_{max}^k - \min\,[D_i - \sum_{z=j+1}^{M} d_{i,z}, t_{i,p} - \sum_{z=j+1}^{p-1} d_{i,z}] \tag{7}$$

Caso contrário:

$$q_{i,j} = D_{max}^k - (D_i - \sum_{z=j+1}^{M} d_{i,z}) \tag{8}$$

Após a determinação dos valores de $a_{i,j}$ e $q_{i,j}$ para a totalidade das operações realizadas numa determinada máquina m_k, pode aplicar-se o algoritmo de Carlier, obtendo-se um plano provisório. A partir das datas de início provisórias definidas por este algoritmo, determina-se as datas mínimas de entrega possíveis das diferentes ordens de fabrico, $D_{i_{min}}^k$, da seguinte forma:

Se já foi fixada a data de início de alguma operação da ordem de fabrico *i* com índice superior a *j*, então:

(seja *p* o maior dos índices das operações da ordem de fabrico *i* com data de início já fixada, com p>j)

$$D_{i_{min}}^k = t_{i,p} + \sum_{z=p}^{M} d_{i,z} \tag{9}$$

Caso contrário:

$$D^k_{i_{min}} = t_{i,j} + \sum_{z=j}^{M} d_{i,z} \qquad (10)$$

Conhecidos os valores de $D^k_{i_{min}}$, determina-se o maior dos atrasos, T^k_{max}, induzido pela máquina m_k.

$$T^k_{max} = \max\,(0, D^k_{i_{min}} - D_i) \qquad (11)$$

Após a aplicação do passo (iii) da heurística a todas as máquinas $m_k \in I_0$, torna-se definitiva a sequência de processamento das operações realizadas na máquina que induz o maior dos atrasos , m* - passo (iv). Seguidamente, actualizam-se os conjuntos $I_1 = I_1 \cup \{m^*\}$ e $I_0 = I_0 \backslash \{m^*\}$.

Finalmente, no passo (v), são actualizadas as datas de início das operações realizadas em máquinas com sequenciamento já fixado a fim de que as restrições de precedência das operações de cada ordem de fabrico sejam respeitadas.Essa necessidade é seguidamente exemplificada com a ajuda de um problema de programação de operações envolvendo sete ordens de fabrico e sete máquinas (Marques [1993]).

Na figura 1 representa-se o plano de fabrico obtido para as máquinas 1 e 2, através da aplicação do algoritmo de Carlier. Pode-se verificar a existência de duas situações distintas:

(i) As restrições de precedência não são respeitadas. Por exemplo, a 4ª e a 7ª operações da ordem de fabrico 6 (operações 6/4 e 6/7 na figura) são processadas simultaneamente; a 6ª operação da ordem de fabrico 7 é processada antes da 5ª operação da mesma ordem. Nos exemplos referidos há uma evidente violação das restrições de precedência das operações de uma mesma ordem de fabrico.

(ii) A folga entre duas operações não consecutivas da mesma ordem de fabrico é inferior ao somatório dos tempos de processamento das operações intermédias.É o que acontece, por exemplo, relativamente à ordem de fabrico 5, em que a folga entre a 3ª e a 7ª operação é de 20 unidades de tempo (77-57), e em que a soma dos tempos de processamemto das operações 4, 5 e 6 é, no presente problema, de 42 unidades de tempo.

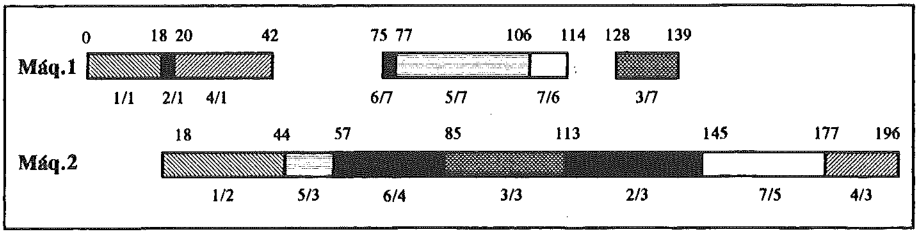


Figura 1 - Gantt após o sequenciamento das máquinas 1 e 2 (antes da actualização das datas de início)

Na actualização das datas de início das operações deve-se ter em conta, por um lado, a possibilidade de existirem outras operações da mesma ordem de fabrico realizadas em máquinas ainda não sequenciadas e, por outro, o facto de uma máquina não poder processar simultaneamente mais do que uma ordem de fabrico.

A determinação da menor data possível para o início de processamento da operação $O_{i,j}$, processada numa máquina m_k já sequenciada, é feita do seguinte modo:

A determinação da menor data possível para o início de processamento da operação $O_{i,j}$, processada numa máquina m_k já sequenciada, é feita do seguinte modo

- Se já foi fixada a data de início de alguma operação da ordem de fabrico *i* com índice inferior a *j*:
  - Seja *p* o maior dos índices das operações da ordem de fabrico *i* com data de início já fixada, com p<j
  - Seja $S_{(k,n)}$ o par (i,j) (ordem de fabrico, operação) processado na máquina *k* em n-ésimo lugar

$$t_{i,j}^{min} = \max[t_{i,j} + \sum_{s=p}^{j-1} d_{i,s},\ t_{S(k,n-1)} + d_{S(k,n-1)}] \qquad (12)$$

- Se não:

$$t_{i,j}^{min} = \max[\sum_{s=1}^{j-1} d_{i,s},\ t_{S(k,n-1)} + d_{S(k,n-1)}] \qquad (13)$$

Após a aplicação das expressões (12) e (13) ao exemplo, obtem-se o plano para as máquinas 1 e 2 representado pelo Gantt da figura 2.

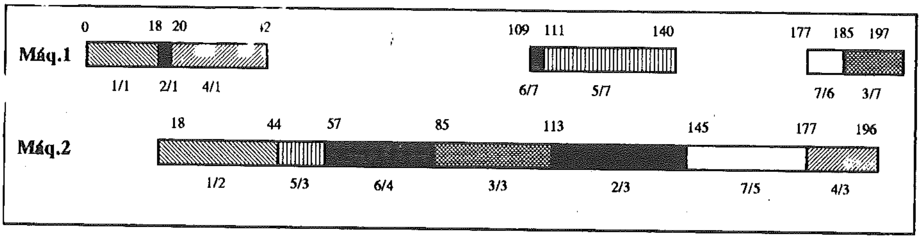


Figura 2 - Gantt para as máquinas 1 e 2 (após a actualização das datas de início)

Heurística PRO-II

É possível melhorar os resultados obtidos pela heurística PRO-II alterando o critério utilizado na selecção da próxima máquina a sequenciar.

A regra utilizada para essa selecção - passo (iv) da heurística - limita-se à análise individual do maior dos atrasos associado a cada uma das máquinas, partindo do princípio que as operações realizadas em máquinas ainda não sequenciadas não agravam o valor desses atrasos.Este prossuposto não é realista, pois pode ser impossível, após a definição do sequenciamento e calendarização das operações realizadas nas restantes máquinas, obterem-se os atrasos calculados pela regra utilizada. Esta regra não tem em conta a interacção existente entre as diferentes máquinas.

Foi dentro desta linha de raciocínio, e tendo em vista a melhoria dos resultados obtidos, que se construiu, a partir da heurística PRO-I, uma outra mais elaborada, designada por PRO-II. A única diferença entre estas heurísticas reside no critério utilizado para a selecção da próxima máquina a sequenciar.

Vai-se designar por M_0 o conjunto das máquinas para as quais ainda não foi fixada a sequência das operações a realizar, e por M_1 o conjunto das máquinas para as quais já foi definida essa sequência.

A heurística PRO-II compreende os seguintes passos:

(i) Inicializar os conjuntos $M_1 = I_1 = \varnothing$; $M_0 = I_0 = \{m_1, m_2, m_3,\ldots, m_M\}$

(ii) Se $M_0 = \varnothing$, terminar; se não, continuar.

(iii) Para cada máquina $m_k \in M_0$, associar um problema a ser resolvido pela heurística PRO-I do seguinte modo:

- (a) Aplicar o algoritmo de Carlier (ou Schrage) à máquina m_k;
- (b) Inicializar os conjuntos $I_1 = M_1 \cup \{m_k\}$ e $I_0 = M_0 \backslash \{m_k\}$;
- (c) Aplicar a heurística PRO-I a este problema;
- (d) Determinar o maior dos atrasos, T^k_{MAX}, associado ao plano assim obtido;

(iv) Seleccionar, para próxima máquina a sequenciar, a máquina m_k que conduzir ao menor valor de T^k_{MAX}. Seja m* essa máquina.
Actualizar os conjuntos M_1 e M_0:
$M_1 = M_1 \cup \{m^*\}$
$M_0 = M_0 \backslash \{m^*\}$

(v) Actualizar as datas de início de todas as operações realizadas em máquinas $m_k \in M_1$ com sequência já fixada.

(vi) Voltar ao passo (ii).

Repare-se que agora o critério adoptado para a selecção da próxima máquina a sequenciar baseia-se nos valores do maior dos atrasos associados aos planos que se obêm sequenciando em primeiro lugar cada uma das máquinas $m_k \in M_0$ e aplicando seguidamente a heurística PRO-I.

Esta nova regra tem em conta a interdependência do sequenciamento das várias máquinas, sendo o maior dos atrasos, T^k_{MAX}, um valor com aderência à realidade.

A heurística PRO-II implica a aplicação da heurística PRO-I (m/2) (m+1) vezes. O tempo de computação necessário para a obtenção da solução é assim maior do que para o caso da heurística PRO-I. No entanto, a melhoria dos resultados obtido justifica o acréscimo de tempo de computação.

3. Incorporação de máquinas pré-sequenciadas

Os agentes de planeamento conseguem frequentemente identificar os recursos que se comportam como gargalos ao processo de fabrico (máquinas críticas). O sequenciamento destas máquinas obedece muitas vezes a regras muito específicas de cada processo de fabrico e de natureza pouco estruturada, sendo por isso a sua generalização e codificação tarefa bastante difícil.

Nestes casos a utilização de sistemas interactivos de sequenciamento como, por exemplo, o sistema INGANTT de Marques e Gonçalves [1991] parece ser a melhor solução. O utilizador pode então concentrar os seus esforços na definição do sequenciamento desses recursos, deixando para a heurística a tarefa de sequenciar automaticamente todas as outras máquinas (não críticas).

As heurísticas PRO-I E PRO-II permitem ao utilizador definir as sequências de processamento das operações realizadas nas máquinas críticas.Depois de o utilizador fixar essas sequências, a heurística não as altera em nenhuma circunstância. Para o caso de se definirem previamente as sequências de processamento das operações de alguma (ou algumas) máquina(s), a heurística descrita só é alterada no passo relativo à inicialização dos conjuntos I_1 e I_0 (passo (i)), que passa a ser feita do seguinte modo:

$I_1 = \{m_1, m_2, \ldots, m_c\}$, sendo c o número de máquinas críticas definidas pelo utilizador;

$I_0 = \{m_{c+1}, \ldots, m_M\}$, sendo M o número total de máquinas.

Tal como a heurística PRO-I, também a heurística PRO-II permite a definição de um conjunto de máquinas críticas, cujo sequenciamento é fixado pelo agente de planeamento. O procedimento a utilizar nesse caso é em tudo análogo ao que foi descrito, com excepção do passo (ii) em que se faz a inicialização do conjunto M_1. No caso de existirem máquinas pré-sequenciadas, o conjunto M_1 deverá incluir inicialmente essas máquinas críticas.

4. Reoptimização de máquinas já sequenciadas

Após se ter tornado definitiva a sequência de processamento da máquina crítica (ver passo (iv) da heurística PRO-I), esta nunca mais é alterada. No entanto, é possivél obter-se melhores soluções se, sempre que se sequencia uma nova máquina, se corrijam as sequências definidas para as máquinas anteriormente programadas.

Introduziu-se assim um novo passo nas heurísticas desenvolvidas, que consiste na reoptimização das máquinas já programadas, sempre que uma máquina é sequenciada. Nesse novo passo, aplica-se o algoritmo de Carlier a cada máquina com sequência já defenida, recalculando previamente para esse efeito os valores de $a_{i,j}$ e $q_{i,j}$ tendo em conta a sequência fixada para a nova máquina sequenciada.

Exemplificando relativamente à heurística PRO-I, seria introduzido um novo passo, a executar depois do passo (v), que consistiria no seguinte:

Par cada máquina $m_k \in I_1$:

- Determinar os parâmetros $a_{i,j}$ e $q_{i,j}$ necessários à aplicação do algoritmo de Carlier, tendo em conta as datas de início das operações da mesma ordem de fabrico já fixadas e realizadas noutras máquinas $m_k \in I_1$ (incluindo a máquina m* acabada de sequenciar);
- Aplicar o algoritmo de Carlier a cada máquina $m_k \in I_1$, a fim de se obter um novo plano para cada uma delas.

De uma forma análoga, para o caso da heurística PRO-II, introduz-se o novo passo de reoptimização após o passo (v) desta heurística, sendo agora tratadas as máquinas $m_k \in M_1$. Em ambos os casós , isto é, quer para a heurística PRO-I, quer para a heurística PRO-II, obtém-se uma melhoria significativa nos resultados, quando se incorpora a reoptimização descrita. Adams et al. [1988] apresentam uma heurística para a resolução do problema do "job-shop", tendo por objectivo a minimização do "makespan". Nessa heurística é utilizado um procedimento de reoptimização semelhante ao descrito.

5. Resultados computacionais

As heurísticas apresentadas foram codificadas e incorporadas no sistema INGANTT referido anteriormente. A fim de testar o seu desempenho foram gerados aleatoriamente 100 problemas, para os quais se conhece uma solução óptima (sem nenhum ataso).

Para cada problema é definido o número de máquinas, M, e o número de ordens de fabrico, N, sendo o número total de operações dado por N×M. Uma ordem de fabrico pode ter mais do que uma operação processada na mesma máquina. O número de operações de cada ordem de fabrico é, em média, igual ao número de máquinas. A probabilidade de uma determinada operação $O_{i,j}$ ser processada em qualquer uma das máquinas é idêntica. Os tempos de processamento das operações foram gerados aleatoriamente segundo uma distribuição uniforme no intervalo [5,50]. Para cada combinação N×M foram gerados 4 problemas diferentes.

Cada problema gerado foi resolvido através das duas heurísticas PRO-I e PRO-II, incluíndo ambas o procedimento de reoptimização. A fim de se avaliarem as soluções obtidas pela heurística, foram utilizadas, na resolução do mesmo conjunto de problemas, as regras de sequenciamento EDD ("Earliest Due Date"), MDD ("Modified Due Date"), FIFO ("First In First Out"), Min. SLACK ("Minimum Slack"), SLACK/RPT ("Slack per Remaining Processing Time") e SPT ("Shortest Processing Time"). Estas regras foram escolhidas por serem algumas das mais frequentemente utilizadas na resolução de problemas industriais.

Os resultados obtidos pelas heurísticas são apresentados na tabela I. Na primeira coluna da tabela indica-se o número total de operações, fazendo-se ainda referência ao número de ordens de fabrico e de máquinas (OFs×Máq). Para cada combinação N×M, foram gerados quatro problemas diferentes. Na segunda coluna (Regras) apresenta-se o maior dos atrasos associado à melhor solução obtida pelo conjunto de regras de sequenciamento referidas anteriormente.

Nas duas últimas colunas apresenta-se o maior dos atrasos obtido respectivamente pela heurística PRO-I e pela heurística PRO-II. Em ambas foi utilizado o algoritmo de Carlier para a

Tabela I - Resultados obtidos

Nº Op.	Regras	Heurística	
		PRO-I	PRO-II
9(3x3)	22	0	0
	42	0	0
	0	0	0
	56	25	25
15(5x3)	0	0	0
	22	0	0
	34	0	0
	13	0	0
18(6x3)	31	0	0
	11	0	0
	35	9	0
	18	0	0
27(9x3)	25	0	0
	3	0	0
	0	0	0
	49	10	0
25(5x5)	32	0	0
	33	36	0
	49	0	0
	93	0	0
40(8x5)	61	3	36
	15	20	11
	40	0	3
	0	0	0
50(10x3)	45	0	0
	15	0	0
	14	0	0
	15	25	0
75(15x5)	78	12	0
	91	6	0
	31	0	0
	44	1	24
100(20x5)	21	39	34
	90	8	6
	45	0	0
	47	0	18
49(7x7)	16	23	0
	19	1	0
	0	13	0
	4	32	0
77(11x7)	20	24	2
	18	14	5
	38	0	12
	14	35	18
98(14x7)	27	0	0
	61	2	23
	50	0	0
	30	9	0
147(21x7)	211	17	16
	14	0	0
	19	10	21
	49	0	8
189(27x7)	102	8	0
	196	26	6
	117	0	0
	50	1	11
81(9x9)	0	13	0
	80	29	38
	21	23	0
	79	22	2
117(13x9)	99	40	32
	66	18	18
	47	20	14
	0	14	18
162(18x9)	0	7	6
	26	13	25
	46	11	7
	52	60	20
189(21x9)	0	33	3
	11	12	9
	35	0	0
	0	18	12
243(17x9)	48	13	0
	55	0	0
	0	0	2
	40	39	8
100(10x10)	29	31	0
	22	9	0
	12	27	0
	48	39	21
150(15x10)	47	11	10
	5	15	7
	0	0	11
	52	36	45
200(20x10)	62	51	78
	21	0	29
	68	9	0
	0	0	16
250(25x10)	19	3	3
	0	13	0
	17	10	45
	161	29	21
300(30x10)	190	76	62
	16	0	1
	96	32	5
	0	0	16
165(15x15)	42	48	40
	59	3	12
	7	3	36
	68	25	15
36(6x6)	6	5	5
100(10x10)	199	88	123
100(20x5)	201	65	65

resolução do problema do sequenciamento e calendarização das operações numa única máquina. Para o conjunto dos 100 problemas gerados verificou-se que a heurística PRO-I obteve a solução óptima em 38 problemas (38%), enquanto a heurística PRO-II a encontrou em 48 (48%). (Note-se que a solução óptima é encontrada sempre que o maior atraso seja nulo). A heurística PRO-I obteve uma solução melhor ou igual à melhor obtida pelo conjunto das outras regras de sequenciamento em 79 problemas (79%), enquanto a heurística PRO-II a obteve em 84 problemas (84%).

No final da tabela I apresentam-se os resultados obtidos para os problemas de Muth et al.[1963], utilizando-se para as datas devidas de entrega os valores óptimos do "makespan" de cada um dos problemas.

Bibliografia

[1] Adams,J., Balas, E., Zawack D., The Shifting Bottleneck Procedure for Job Shop Scheduling, Management Science (1988) 391-401.

[2] Baker, K.R., Introduction to Sequencing and Scheduling, John Wiley, New York, 1974.

[3] Baker, K.R., Su,Z.S., Sequencing with Due-dates and Early Start Times to Minimize Tardiness, Naval Research Logistics Quartely 21 (1974) 171-176.

[4] Carlier, J., The One-machine Sequencing Problem, European Journal of Operacional Research 11 (1982) 42-47.

[5 Carlier, J., Scheduling Jobs With Release Dates and Tails on Identical Machines to Minimize Makespan, European Journal of Operational Research 29 (1987) 298-306.

[6] Conway , R.W., Maxwell, W.L., Miller, L.W., Theory of Scheduling, Adison-Wesley, Mass., 1967.

[7] French, S., Sequencing and Scheduling: An introduction to the Mathematics of the Job-Shop, Ellis Horwood, ltd., Southampton, U.K., 1982.

[8] Garey, M.R., Johnson, D.S., Computers and Intractability: a Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[9] Grabowski, J., Nowicki, E., Zdrzalka, S., A Block Approach for Single Machine Scheduling with Release and Due Dates, European Jornal of Operational Research 26 (1986) 278-285.

[10] Graham, R.L., Lawer, E.L., Lenstra, J.K., Optimisation and Approximation in Deterministic Sequencing and Scheduling: A Survey, Annals of Discrete Mathematics 5 (1979) 287 - 326.

[11] Lenstra, J.K., Rinnooy, K., Brucker, P., Complexity of Machine Scheduling Problems, Annals of Discrete Mathematics 1 (1977) 343 - 362.

[12] Marques, M.P., Programação de Operações Fabris em Ambiente de Job-Shop: Nova Abordagem, Tese de Doutoramento, FEUP,1993.

[13] Marques, M.P., Gonçalves, J.F., INGANTT - A Decision Support System for Sequencing Operations, comunicação apresentada na conferência IFORS -SPC1, Bruges 1991.

[14] McMahon, G., Florian, M., On Scheduling with Ready Times and Due Dates to Minimize Lateness, Operations Research 23 (1975) 475 - 482.

[15] Rinnooy, K., Machine Scheduling Problems: Classification, Complexity and Computations, Martinus Nijhoff, The Hague, Holland, 1976.

INVESTIGAÇÃO OPERACIONAL, vol. 14, n.° 1 (Junho 1994) 91-102

RISK ANALYSIS USING TIME PETRI NETS

Wim Deceuninck and Gerrit K. Janssens
Faculty of Applied Economics, University of Antwerp (RUCA)
Middelheimlaan 1, B-2020 Antwerpen, Belgium

Abstract

This paper discusses some possible uses of Petri nets in risk analysis. After mentioning how to avoid risk, how to recover from a risky state, and how to detect how fault tolerant a modeled system is, risk in project planning is studied in detail. After having calculated all possible state classes in which the system can arrive, global clock timing information is added to each class. Timings for the best case and the worst case for the global project and each of its subcomponents are obtained. A worked out example of a chemical plant is given.

Keywords

Accidents, networks, Petri net, project management, safety.

1. Introduction

According to C.B. Chapman (1992, p.37) successful risk management requires: "(1) a flexible and general set of verbal, graphical and mathematical models, supported by appropriate computer software; (2) a family of related methods, designed to suit the models, which link the model and the circumstances in which they are to be used; (3) a wide range of relevant expertise and specialist skills; (4) the experience and leadership to design and integrate models, methods and software for specific risk management tasks, to organize and manage risk analysis study teams, and to execute successfully analyses for projects."

This paper studies the opportunities Petri nets offer to improve the first of Chapman's requirements.

As a general but powerful and versatile graphical and mathematical tool, Petri nets can be used for the description and analysis of many systems. The historical roots of Petri nets date back to Carl Adam Petri's dissertation (1962). Since then, a lot of extensions have been carried out, including stochastic Petri nets, timed and time Petri nets, coloured Petri nets, various kinds of transitions and places. A number of excellent books are available to the new-comer in the field (e.g. Peterson (1981), Reisig (1985) and Brams (1983)). Introdutctory and more in-depth review articles can also be found (e.g. Murata (1989) and Peterson (1977)).

Due to its simple representation of concurrency and synchronization, i.e. properties that are difficult to express in traditional formalisms, it is worthful to investigate the opportunities and limitations in the area of risk analysis. By definition, risk analysis operates in an environment of uncertainty : an unwanted event leading to some damage may take place. Nobody knows whether or when it will occur. By this, ordinary Petri nets are not sufficient to study risk. Some of the previously mentioned Petri net extensions dealing with uncertainty and timing information will be required.

Petri nets not only serve as a specification language of concurrent systems. They also allow for a graphical representation easily comprehensible for the non-expert user. Furthermore they are base on a solid mathematical background that allows to develop a theory for analysis and synthesis of the models.

Properties such as the reachability of certain atates in the net (the so called token machine or reachability graph), the boundedness or liveness of the nets have been described thoroughly, enabling the analysis of Petri net under study. In this paper we will not go into a detailed presentation of these properties. For the interested reader, we refer to the supra mentioned reference books and articles. It is, however, required to point out that the theory is not fully developed. Although a great set of results is obtained, a lot of open problems still exist.

The rest of this section is devoted to the formal definition of Petri nets (see e.g. Peterson (1977)) and to an introduction to its extensions of relevance to discuss risk analysis problems.

A Petri net is a 5-tuple (P,T,B,F,M_0) where :

- P is a finite nonempty set of places p_i;
- T is a finite nonempty set of transitions t_i;
- B is the backward incidence function $B : T \times P \to \mathbb{N}$;
- F is the forward incidence function $F : T \times P \to \mathbb{N}$;
- M_0 is the initial marking function $M_0 : P \to \mathbb{N}$ that assigns a nonnegative number of tokens to each place of a net.

One of the many extensions to classical Petri nets is the addition of time features. Two basic models for handling time have been defined: time Petri nets and timed Petri nets. The time Petri net is more general (merlin and Farber (1976)). It can formally be defined as (Berthomieu and Diaz (1991)):

A **time** Petri net is a 6-uple (P,T,B,F,M_0,SIM) where:

- (P,T,B,F,M_0) is defined as a Petri net;
- SIM is a mapping called static interval SIM : $T \to Q^+ \times (Q^+ U \infty)$ and

$$SIM(t_i) = (\alpha_i^s,\beta_i^s) \text{ where } \begin{cases} 0 \leq \alpha_i^s < \infty \\ 0 \leq \beta_i^s \leq \infty \\ \begin{cases} \alpha_i^s \leq \beta_i^s \text{ if } \beta_i^s \neq \infty \\ \text{or} \\ \alpha_i^s < \beta_i^s \text{ if } \beta_i^s = \infty \end{cases} \end{cases}$$

where:

- the interval of numbers (α_i^s,β_i^s) is called the static firing interval of transition t_i;

- α_i^s and β_i^s are called respectively the static earliest firing (EFT) and the static latest firing time (LFT).

The sequence of events in any system modelled by a Petri net can vary due to inherent randomness or due to an external hazard, but this is of no importance in the analysis of the Petri net. All sequences are considered, either explicitly or implicitly.

In Petri nets without timing information, transitions which are enabled can fire any moment but will not necessarily do. It is not specified when, if ever, transitions will fire, and why. The occurrence of transitions is not planned. The causal structure of transitions - pre and post conditions - is described, but the control over the occurrence transitions is not specified by the design. Rather, it is assumed to lie in the environment in which the system is acting. As the system's environment is postulated but not described, transitions represent local spontaneous happenings.

The **time** Petri net case is slightly different. As each transition has a static timing interval (α_i^s,β_i^s) a transition t_i *may not* fire for a period of time of at last α_i^s after it has been enabled, and it *has to* fire if it has been enabled for a period of time of β_i^s.

As our aim is to focus on the opportunities of using Petri nets in risk analysis we will not digress on additional definitions, nor on an exhaustive overview of properties and features of Petri nets.

2. Petri nets and risk

In this paper, three different ways of using Petri nets to model risk are presented. The mutual difference between these implementations of Petri nets lies in the definition that is given to risk. In any case, the possibility of some special, extraordinary or unwanted situation is required to be able to talk about risk. Physical as well as logical situations (e.g. example working procedures) can be considered.

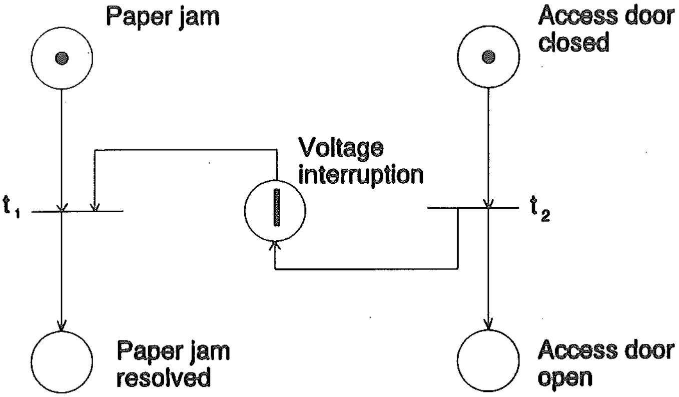
We show how the Petri net representation of a system can be used in designing and analyzing procedures to avoid risk, to make the operations fault tolerant, and to avoid delay of operations. Especially this last type of risk will be studied in further detail.

The three types of risk given here are clearly distinct, but certainly not exhaustive.

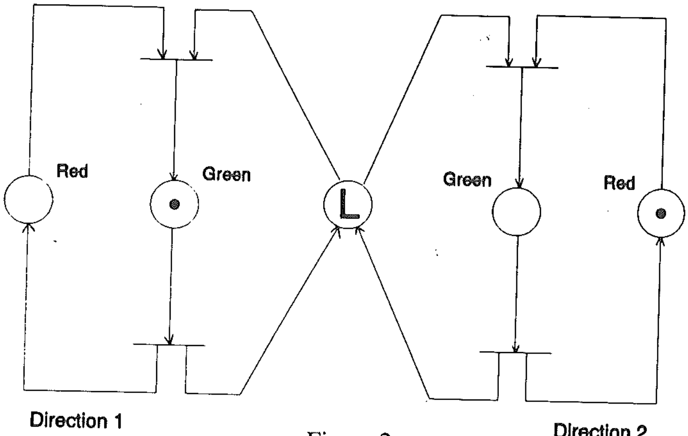
2.1. Avoiding risk or recovering from risk

The main goal in designing safe systems is to eliminate risk from the design. If this is not possible, it is desirable that the system is designed in such a way that only an acceptable level of risk remains. To avoid more hazardous states firing sequences of transitions that could lead to higher-risk situations should be eliminated. This can be done by enforcing timing constraints, or by using interlocks, locking places or watchdog timers. An interlock enables control on the sequence of events. Locking places ensure that specific events do not occur at the same time though the sequence of the events is undecided. Watchdog timers ensure reoccurrence of an event until acknowledgements of all required results of that event are received. It is realised by a place in which a token is placed if a first transition fires. If everything works successfully, another transition later removes the token and stops the timer. If a failure occurs on the path between both transitions, an extra transition should fire after some time and deliver a token to the original placé, allowing the latter to restart the process all over again.

As an example of an interlock an access panel or a door to high-voltage equipment can serve, e.g. if one wants to resolve a paper jam in a photocopier, a voltage interruption at all devices is realised automatically by opening the access door. This Petri net is shown in Figure 1. Examples of locking places are: (1) avoiding the situation that a train is approaching a crossing while the gate is up, or (2) avoiding that a traffic light shows green to both directions at a crossroads. The latter example is shown in figure 2. They are similar as critical sections in software for parallel processing. An example of a watchdog timer can be found in communications protocols retransmitting a packet after a time out interval, if no acknowledgment was received and, by this, assuming that the packet is lost.



Example of an interlock Figure 1



Example of a locking place Figure 2

If the design allows for a certain level of risk to be reached, we want to ensure that the system will not remain in that situation. Assume a system is modeled as a Petri net and risk defined as some combination of conditions resulting in an unwanted situation. If at least one situation exists without the possibility of recovering towards an acceptable situation using designed transitions from one state to another, then the system needs to be redesigned in such a way that recoverability from the risk state to an acceptable state is accomplished. Modeling the system by means of a Petri net enables the generation of the reachability graph, which comprises all possible situations reached from a certain initial state. Analysis of the reachability graph enables the detection of risk or unrecoverable states.

If the state conditions of a system allow for an unwanted (hazardous) event to occur, it is important that the system is redesigned in such a way that controlling actions start before the damaging event can take place. By this, a specification of a system pretending to be "safe" may require the inclusion of timing information.

Examples of this use of Petri nets can be found in Merlin and Farber (1976) or Leveson and Stolzy (1987). Risk being defined in this way, as a unique combination of conditions, can be used e.g. in storage of dangerous chemicals, in systems where time-space coexistence of two elements in the systems can lead to unwanted physical collisions (guarded or unguarded level crossings, traffic, automatic guided vehicles, etc.) and many others.

2.2. Risk as fault tolerance

In the former view of risk, one still assumes that the system behaves as it was designed, with knowledge of all possible states of the system or its components. However, risk could also be defined as the misbehaviour of a designed system. After having found a modeled system that behaves without producing more than acceptable risk, a new kind of Petri net is superimposed on the modeled one. This new Petri net allows a behaviour not foreseen by the original one. However, with some slight and easy performable additions, that new Petri net can be remodeled and analyzed as discussed in 2.1. Misbehaviour is implemented using fault-generating transitions. The generated faults can be seen as the unforeseen occurrence of an event, or as a desired event that does not occur. In the former case a token is added to a certain place in the Petri net, in the latter a token will disappear. An example of using Petri nets with this view on risk can be found in Leveson and Stolzy (1987). Such systems can be transformed into a system discussed in the previous paragraph. Therefore no further comments will be given here.

2.3. Time-risk

When focusing primarily on time, risk can be seen from another viewpoint. In the former cases, risk is defined as some special state. Now we consider the total time period required to finish a set of activities. Delays can have important physical or financial consequences, but also unnecessary early termination of a project might imply some kind of risk. This so-called project planning risk is discussed (without the use of Petri nets) in Chapman (1992). The ability of

Petri nets to model shared resources makes them superior to PERT, CPM or GERT networks as project management representations.

Continuity, a very specific nature of time implies that an exhaustive enumeration of possible states (with time-information) is not possible, as it will almost always be an infinite enumeration. Instead of handling individual states, states can be grouped in state classes.

Formally, the *i*th state class is defined by its marking M_i and its firing domain D_i, where D_i contains all enabled transitions $t_{i,j}$ with their relative firing interval ($\alpha_j < t_{j,j} < \beta_j$). Classes are pairs (M,D) in which M is a marking and D is a firing domain. Firing rules for state classes and ways to compare classes for equality have been defined (Berthomieu and Menasche, 1983). Using these state classes, a finite representation of an infinite number of reachable states can be generated, which will be mentioned further by the *class reachability graph*, similar to the reachability graph of states in (non-time) Petri nets.

As an example in Figure 3 a part of a chemical plant is modelled. The plant transfers two types of raw materials (R_A and R_B) into two kinds of finished products (F_A and F_B) through the production of three semi-manufactured products (S_A, S_B and S_C). Assume that the processes transforming the raw materials into the semi-manufactured products, and the processes transforming the semi-manufactured products into finished products, have certain mutual constraints. Further let product R_B be transformed into a product S_C in reactor 1, and product R_A be transformed in reactor 2 into S_A and S_B by different catalysts resulting in different product chemical reactions. Note that the latter two reactions are performed in the same reactor, so it is very important for safety reasons that never both reactions are going on at the same time. It should be avoided that both kinds of catalysts are put in the reactor at the same time. To realise this a special catalyst controller is added. The addition of the catalyst controller ensures that reaction 1 or reaction 2 is enabled without the possibility of enabling both simultaneously.

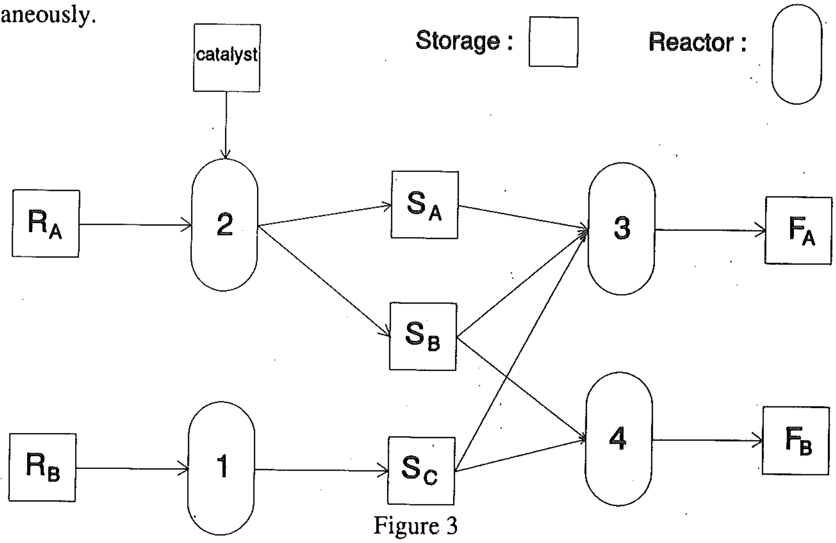


Figure 3

Further assume that the chemical reaction transforming S_A, S_B and S_C into F_A (in reactor 3), and the reaction transforming S_B and S_C into F_B (in reactor 4), are of a specific dangerous nature, requiring the attendance of a higher educated operator. The fact than an operator cannot control both reactors three and four at the same time is called a *shared resources* condition.

To complete the chemical plant model, we add the amounts of chemical products required and produced by each reaction, and a minimum and maximum time that is needed to perform the reaction. To finalise our model, two artificial Petri net places A_1 and A_2 are added in order to prevent each reaction to be executed more than once.

The graph structure used for illustration of the Petri net of the chemical plant is shown in figure 4. In Petri nets, circles represent places, and vertical bars represent transitions (with their static firing intervals between brackets).

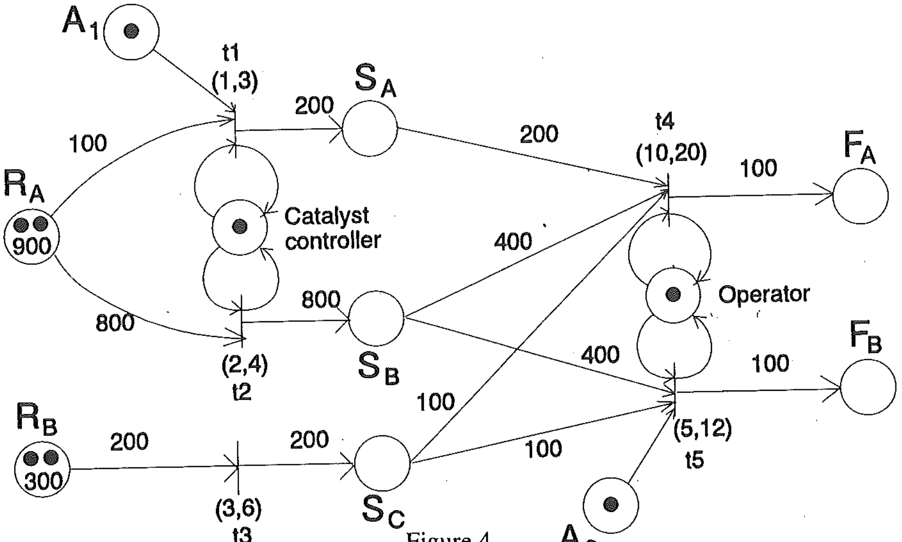


Figure 4

As can be seen, the sequence of events (chemical reactions in this example) is not deterministic. Finished products can be manufactured only when their semi-manufactured products are ready. Which semi-manufactured product is manufactured first, or the sequence in which the finished products are manufactured is left undecided. This can be a random sequence, but we assume that the management of the plant wants to find the best timing schedule to avoid the risk of having delayed production.

As a first step the class reachability graph of the time Petri net is generated. An enumerative procedure to obtain the class reachability graph can be found in Berthomieu and Menasche (1983) or in Menasche and Berthomieu (1983). Their approach assumes the existence of *state classes*. For each firing sequence *s*, the set of all states reachable from an initial state by firing schedules with this firing sequence is considered. This set of states is called the State Class.

In the following paragraphs we briefly discuss the enumerative method for analyzing Time Petri nets.

The global clock

Once the state class reachability graph has been generated including markings and relative firing domains for all enabled transitions, an absolute or global clock timing interval is assigned to each state class in order to find the absolute timing interval of the desired final class.

The initial class C_0 by definition has (absolute) class timing interval (0,0). By firing a transition t_k with static firing interval (α_k^s,β_k^s) which is enabled in a class C_i with timing interval (α_i,β_i), another class C_j with yet unknown absolute timing interval (α_j,β_j) is reached. To identify the according absolute class timing interval, special attention should be given to the absolute time epoch at which the fired transition became enabled. If transition t_k became "newly enabled" in class C_i, the absolute timing interval of class C_j is $(\alpha_j,\beta_j) = (\alpha_i+\alpha_k^s,\beta_i+\beta_k^s)$. In the case the enabled transition is a self-loop transition, it looses the required markings to fire but becomes enabled again immediately. Firing this self-loop transition implies that it did not remain enabled, but becomes a "newly enabled" transition (with a new relative firing interval equal to the static timing interval).

However, if t_k became enabled at a previous class C_n with absolute timing interval (α_n,β_n), and if a sequence of firings of other transitions resulting in a move from class C_n to class C_i retained t_k enabled, then

$$(\alpha_j,\beta_j) = (\max(\alpha_i,\alpha_n+\alpha_k^s), \max(\beta_i,\beta_n+\beta_k^s))$$

As for a newly enabled fired transition $C_n = C_i$, we have

$$(\alpha_i+\alpha_k^s,\beta_i+\beta_k^s) = (\max(\alpha_i,\alpha_n+\alpha_k^s), \max(\beta_i,\beta_n+\beta_k^s))$$

so it is sufficient to retain the second more general formula.

The OR function

Depending on the different sequences of transition firings chosen, classes are reached at different moments in time. Identical classes can be reached following different firing sequences. To perform our project planning risk analysis in this case, we take for each class the best case and the worst case by calculating a kind of OR function of all time windows through which the class is reachable. If the marking of a class C_k is reachable by firing transitions t_i and t_j with relative firing intervals (α_i,β_i) and (α_j,β_j) resp., the relative firing interval of class C_k is defined by $(\min(\alpha_i,\alpha_j), \max(\beta_i,\beta_j))$. This definition differs from the logical OR function due to the absence of an overlap. The interval (β_i,α_j) is always included, even if $\beta_i < \alpha_j$.

The individual and final absolute class timing intervals

Following the rules above, we can calculate absolute timing intervals for all classes. To illustrate the operation of these rules, the state class reachability graph is given in Figure 5. Circles denote state classes, numbered from 0 to 14. For each class the absolute class timing intervals are given. By firing a transition t_i we move from one class to another. The dotted lines point to the classes in which the transitions were enabled. If the transition was enabled in the class immediately prior to the current class, the dotted line is omitted.

Proceeding through the calculation of all absolute class timing intervals, ultimately an interval for the final class is found. In figure 5 this final class is indicated by C_5. With our choice of parameters its absolute timing interval is (18,39). At this point the decision maker has the possibility to accept or reject this interval. In case of rejection the static timing intervals of some transitions should be reconsidered. In larger applications it might be appropriate to redesign the system by enforcing or prohibiting specific firing sequences using interlocks. The real world equivalent of this interlock might be an additional rule to the procedure to be followed before some activity is started.

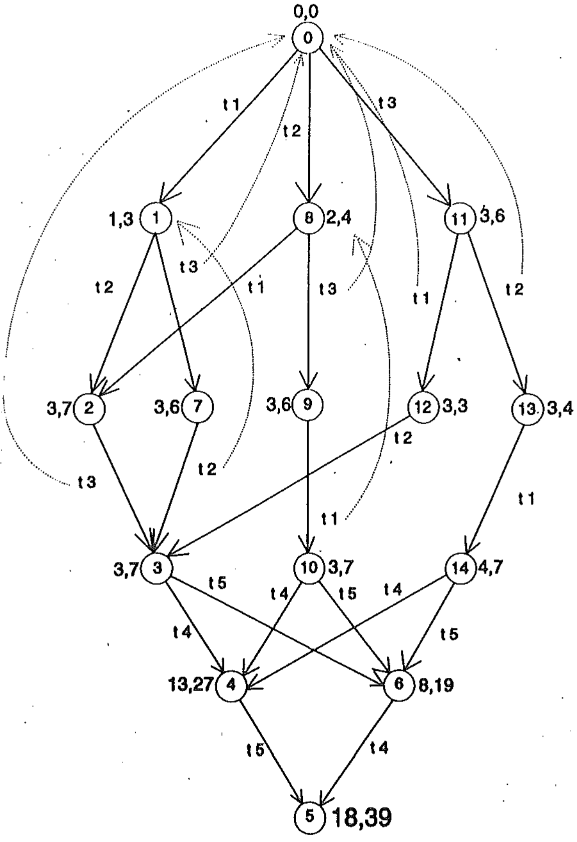


Figure 5

Having calculated all timings, we now see clearly how individual firing times determine the possible sequence of transition firings and how they influence the possible finish times of the production run.

3. Limitations on the use of Petri nets and risk

The reachability tree is the basis for all analysis problems related to the net under study. It is the enumeration of all reachable markings. It can theoretically be applied to all classes of Petri nets, whatever features have been added.

Generating the entire graph can be impractical due to its size for a complex system. Other techniques as matrix equations and reduction techniques are powerful but are limited to subclasses of Petri nets (e.g.Murata (1989, section V)).

In risk analysis we are interested only in high-risk states and how they can be reached. The backward reachability graph of the inverse Petri nets (i.e. the net in which input and output functions are reversed) can serve this goal. It gives an answer on the question whether a high-risk state be reached from the initial state. But, as this graph is as large as the original graph, this does not solve our problem of size. Leveson and Stolzy (1987) present a solution to this problem. By introducing a particular type of state, called 'critical state', their algorithm does not require the entire backward reachability graph to be generated.

Similar size problems have been encountered in modeling systems by means of Markovian modeling. In case the number of system states becomes too large, numerical problems appear to solve the balance equation for the steady-state probabilities. The extension of Petri nets to stochastic Petri nets has eased the use of Markovian models. If an exponentially distributed random firing time is associated with each transition, the reachability is isomorphic to a homogeneous Markov chain (Pagnoni (1986)). In free-choice nets, probabilities can be assigned to a set of enabled transitions. Masapati (1987) studies the effect on the opportunities of vertex reduction rules in these graphs.

Hand-constructed Markov models can now be generated by simple understandable stochastic network models. But also here, the traditional methods become intractable for large systems. Coloured and regular nets make use of a lumping idea. Sanders and Meyer (1991) also reduce the stochastic process by making use of the specific performance variable. They present an algorithm to construct this "reduced" process without requiring the generation of the complete marking space. Their method is not applicable to the supra mentioned problem of investigating high-risk states. In their case the risk is defined as a "bad" performance, measured by a performance variable.

As risk analysis is being done in a stochastic environment, future opportunities can be expected from Stochastic Petri nets (SPN). State changes in an SPN occur with some probability, rather than arbitrarily as in a standard Petri net. An SPN has more structure. The additional structure allows the extraction of additional performance information about the modeled system. The SPN model allows the calculation of the marking probabilities in steady state. This opens up an area for such performance measures as average delay and throughput. A discrete-time stochastic Petri net is discussed in Molloy (1985).

Most authors use SPN with a continuous time domain. They associate with each transition a random firing delay with an exponential distribution (Ajmone Marsan and Chiola, 1987). This is because the models rely upon Markovian theory. Maybe they are well suited for system modelling at a high level of abstraction, but they cannot handle specific timing constraints like time-outs. Also, exponential timing may lead to an undesirable approximation of the system specifications.

More general models are required. Arbitrary probability distributions are used for the transitions of the Petri net. If the model represents a solvable stochastic process it is solved analytically, otherwise a simulation is conducted (Marsan et al., 1989). In some cases a Moment Generating Function (MGF) based approach can be used to derive the MGF of interesting performance measures, and thus closed-form analytical solutions are at least theoretically possible. The nets are called arbitrary stochastic Petri nets (ASPN) to distinguish them from the other proposed stochastic Petri nets (Guo et al., 1992).

4. Conclusion

A short overview of the use of Petri nets in risk is described. Petri nets can help to study risk in a modeled system, to find out how to avoid risk, how to recover from a risky situation towards a more acceptable situation and how to detect the fault tolerance of the designed system. Besides this, Petri nets can be very useful in project management, also because the inclusion of shared resources is very easy. Timing information (on a global clock) about the completion of the project or parts of it can be obtained. Because state classes can be reached following different sequences of transition firings, an OR function is defined to obtain best and worst case information. An example was given for a fictitious chemical plant. Finally, a critical view on the applicability of Petri nets for risk analysis is conceived.

Concluding, it can be stated that a lot of works has to be done in the area of graph reduction. The opportunities of modeling by Petri nets have been described worldwide, and we touched their power in analyzing risk. But we are aware of the complexity problem. As a result we are not over-optimistic on a large scale introduction of Petri nets in the safety area unless in the near future great advances are made on graph reduction or other types of analysis.

5. References

[1] Berthomieu, B. and M. Menasche (1983), An enumerative approach for analyzing time Petri nets, in R.E.A. Mason (ed.), Information Processing 83, North Holland, 41-46.

[2] Berthomieu, B. and M. Diaz (1991), Modeling and Verification of Time Dependent Systems Using Time Petri Nets, IEEE Transactions on Software Engineering SE-17, 259-273.

[3] Brams, G.W. (1983), Réseaux de Petri: Théorie et pratique, Masson, Paris.

[4] Chapman, C.B. (1992), A Risk Engineering Approach to Risk Management, in: Ansell, Jake, and Wharton, Frank (eds.), Risk: analysis, assessment and management, John Wiley & Sons, New York.

[5] Guo, D.L., F. DiCesare and M.C. Zhou (1992), A moment generating function based approach for evaluating extended stochastic Petri nets, IEEE Transaction on Automatic Control 37.

[6] Leveson, N.G. and J.L. Stolzy (1987), Safety Analysis Using Petri Nets, IEEE Transactions on Software Engineering SE-13, 386-397.

[7] Marsan, M.A. and G. Chiola (1987), On Petri nets with deterministic and exponentially distributed firing times, in: Rozenberg, G. (ed.), Advances in Petri nets 1987, Lecture Notes in Computer Science 266, 132-145.

[8] Marsan, M.A., G. Balbo, A. Bobbio, G.Chiola, G.Conte and A. Cumani (1989), The effect of execution policies on the semantics and analysis of stochastic Petri nets, IEEE Transactions of Software Engineering SE-15, 832-846.

[9] Masapati, G.H. (1987), Performance prediction using timed Petri nets, M.Sc. thesis, University of Ottawa, Canada.

[10] Menasche, M. and B. Berthomieu (1983), Time Petri nets for analyzing and verifying time dependent communication protocols, in H.Rudin and C.H. West (eds.), Protocol Specification, Testing and Verification III, Elsevier North Holland, 161-171.

[11] Merlin, P.M. and D.J. Farber (1976), Recoverability of Communication Protocols - Implications of a Theoretical Study, IEEE Transactions on Communications, COM-24, 1036-1043.

[12] Molloy, M.K. (1985), Discrete time stochastic Petri nets, IEEE Transactions of Software Engineering SE-11, 417-423.

[13] Murata, T. (1989), Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE 77, 541-580.

[14] Pagnoni, A. (1986), Stochastic nets and performance evaluation, in: Rozenberg, G. (ed.), Advances in Petri nets 1985, Lecture Notes in Computer Science 222, Springer Verlag, 460-478.

[15] Petri, C.A. (1962), Kommunikation mit Automaten, Schriften des IIM 3, Institut für Instrumentelle Mathematik, Bonn.

[16] Peterson, J.L. (1977), Petri nets, Computing Surveys 9, 223-252.

[17] Peterson, J.L. (1981), Petri Net Theory and the Modelling of Systems, Englewood Cliffs, NJ, Prentice Hall.

[18] Ramchandani, C. (1974), Analysis of asynchronous concurrent systems by timed Petri nets, MAC Technical Reports MAC-TR-120, Massachusetts Institute of Technology, Cambridge MA.

[19] Resig, W. (1985), Petri nets - an introduction, EATCS Monographs on Theoretical Computer Scicence 4, Springer Verlag, New York.

[20] Sanders, W.H. and J.F. Meyer (1991), Reduced base model construction methods for stochastic activity networks, IEEE Journal on Selected Areas in Communications, SAC-9, 25-36.

[21] Zuberek, W.M. (1985), M-timed Petri nets, priorities, preemptions and performance evaluation, in: Rozenberg, G. (ed.), Advances in Petri Nets 1985, Lecture Notes in Computer Science 222, Springer Verlag, 478-498.

INVESTIGAÇÃO OPERACIONAL, vol. 14, n.° 1 (Junho 1994) 103-113

PROGRAMAÇÃO MATEMÁTICA NA OPTIMIZAÇÃO DO PRÉ-ESFORÇO

A. A. Serra Neves
Faculdade de Engenharia
Universidade do Porto
Porto

J. L. Silva Pinho
Universidade do Minho
Braga

Abstract

A mathematical model for the optimum design of the tendon configurations in prestressed concrete structures is presented. Geometrical constraints are considered in order to assure the feasibility of the solutions. The serviceability limit state of the concrete is guarantee by suitable constraints. The minimum of the cost of prestressing solution is used as objective function. A transformation of variables is employed to reduce the optimization to a solution of a linear programming problem. Applications of the proposed method are illustrated by examples.

Resumo

Apresenta-se o modelo matemático que permite a obtenção do traçado óptimo de múltiplos cabos de pré-esforço em estruturas de betão armado pré-esforçado. São consideradas limitações de ordem geométrica, de modo a garantir a exequibilidade da solução. São ainda contempladas as múltiplas condições impostas pela regulamentação estrutural do betão armado pré-esforçado no que respeita à verificação dos estados limites de fendilhação. Definindo como critério de optimização a minimização do custo do pré-esforço, é possível gerar um programa matemático que uma vez expresso em função de um criterioso grupo de variáveis de projecto se transforma num programa linear. Apresentam-se vários exemplos resolvidos.

Keywords

Mathematical programming, prestressed concrete, optimum structural design, linear programming.

1. Introdução

O projecto de uma estrutura resistente é efectuado respeitando a regulamentação de estruturas em vigor, na qual se impõem múltiplas condições, onde se destacam as verificações dos estados limites últimos de resistência e os estados limites de utilização que englobam os estados limites de fendilhação e deformação.

No cálculo/projecto de estruturas de betão armado correntes, seguem-se geralmente duas etapás distintas, antecedidas de uma fase de *concepção/pré-dimensionamento* de armaduras. Aquelas duas fases, *análise estrutural e dimensionamento*, são geralmente independentes, na medida em que as disposições regulamentares permitem a não consideração das armaduras no processo de análise estrutural. Sendo assim, uma vez pré-dimensionadas as secções da estrutura procede-se à quantificação das acções e análise estrutural. Conhecidos os esforços nas múltiplas secções, calculam-se as armaduras necessárias em cada secção.

Nas estruturas de betão armado pré-esforçado o quadro é bem diferente. Aqui, as armaduras de pré-esforço condicionam as acções a considerar na fase de análise estrutural, o que obriga à consideração simultânea das condições inerentes às fases de análise e dimensionamento. O problema que inclui todas essas condições constitui um problema complexo não linear. Por outro lado, a experiência nestes tipos de estruturas permite concluir que as restrições regulamentares condicionantes são as relativas à verificação dos estados limites de fendilhação. Vários trabalhos foram desenvolvidos neste domínio quer referentes a estruturas porticadas [6] [7], quer para meios contínuos [8].

Nesta conformidade, foi desenvolvido um modelo de dimensionamento de cabos de pré-esforço nos seguintes pressupostos:

a) A estrutura é conhecida no que respeita às secções de betão a utilizar [2].

b) As condições relativas à verificação dos estados limites últimos de resistência e as relativas à verificação do estado limite de deformação não são condicionantes.

c) Considera-se "óptima" a solução que implique o mínimo de custo de pré-esforço (valor do(s) pré-esforço(s) multiplicado(s) pelo desenvolvimento do(s) cabo(s) respectivo(s)).

O modelo desenvolvido, segue as disposições contempladas na Regulamentação Portuguesa em vigor REBAP[4] e RSA[5], efectuando-se algumas simplificações na consideração dos estados limites de utilização, de modo a obter relações lineares num conjunto de variáveis convenientemente escolhidas. O modelo contempla a hipótese de uso de vários cabos de pré-esforço independentes.

A consideração dos estados limites últimos de resistência bem como do estado limite de deformação pode ser atendida através de verificações posteriores. Não é explicitamente considerada a utilização do pré-esforço parcial que no entanto não alteraria no essencial o modelo apresentado [8].

2. Condições a considerar

2.1. Considerações gerais

Na formulação de cada problema, são definidas várias secções, onde são verificadas diversas condições, que constituem as restrições do problema.

As restrições envolvidas num problema de optimização do pré-esforço, podem agrupar-se em dois grupos distintos: restrições de tensões e restrições de ordem geométrica. As primeiras decorrem da verificação dos estados limites de fendilhação impostos na regulamentação aplicável [4], [5] e o segundo tipo de restrições, decorre de imposições geométricas, colocadas ao traçado dos cabos.

2.2. Restrições de tensões

As restrições de tensões consideradas nos problemas que envolvem vários cabos de pré-esforço são idênticas às apresentadas no modelo proposto para vigas contínuas [1]. As tensões, são assim limitadas, aos valores regulamentares para cada uma das combinações de acções, associadas aos estados limites a verificar. Assim tem-se:

$$\sigma_{ext} \leq f \qquad (1)$$

em que σ_{ext}, representa a tensão numa das fibras extremas da secção em estudo, para a combinação de acções considerada e f representa o valor da tensão limite que varia com o estado limite de fendilhação a verificar, definido no Quadro-I.

Quadro - I
Limites de tensões

Ambiente	Combinações acções	Estado limite	Limite de tensão f_{min}	f_{max}
Pouco agressivo	Frequentes	Largura de fendas	-	fctk
	Quase permanentes	Descompressão	-	0
	Raras	Compressão máxima	fcd	-
Moderadamente agressivo	Frequentes	Largura de fendas	-	fctk
	Quase permanentes	Descompressão	-	0
	Raras	Compressão máxima	fcd	-
Muito agressivo	Frequentes	Descompressão	-	0
	Raras	Largura de fendas	-	fctk
		Compressão máxima	fcd	-

Com:

fctk - Valor característico de rotura por tracção simples, definido no Artº 16 do REBAP[4].

fcd - Valor de cálculo da tensão de rotura do betão à compressão definido no Artº 19 do REBAP[4].

2.3. Restrições de ordem geométrica

2.3.1. Restrições relativas às ordenadas dos cabos de pré-esforço

Do estabelecimento do recobrimento mínimo de betão, para os diversos cabos de pré-esforço, surgem as limitações às ordenadas associadas a cada um dos cabos, que são da forma (figura 1):

$$y_{ij} \leq h - c_1 \qquad (2)$$

$$y_{ij} \geq c_2 \qquad (3)$$

em que y_{ij}, é a ordenada associada ao cabo **j** na secção **i**, c_1 e c_2 são os recobrimentos mínimos de betão superior e inferior e **h** é a altura da secção.

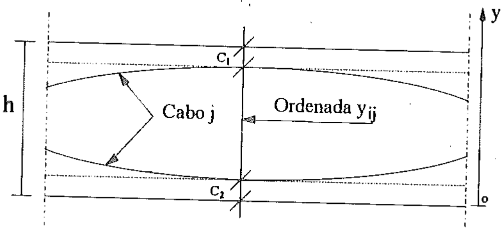


Fig. 1 - Posições limite do cabo de pré-esforço

2.3.2. Outras restrições geométricas

No presente trabalho são considerados unicamente cabos parabólicos em conformidade com a prática corrente no projecto de soluções pré-esforçadas em Engenharia Civil. Assim, cada cabo j é caracterizado pelas ordenadas $y_{i\text{-}2\ j}$, $y_{i\text{-}1\ j}$ e $y_{i\ j}$ em três pontos distintos.

Dois tipos de restrições de ordem geométrica são geralmente considerados no traçado de cabos de pré-esforço. O primeiro tipo refere-se às restrições resultantes da imposição de igualdade de tangentes, entre trechos de cabos parabólicos adjacentes (figura 2).

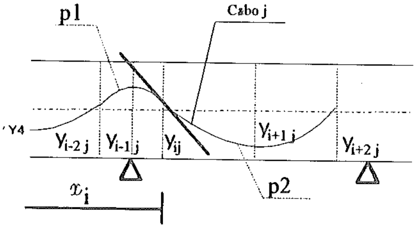


Fig. 2 - Igualdade de tangentes

$$p_1'(y_{i\text{-}2\ j}, y_{i\text{-}1\ j} \text{ e } y_{i\ j})_{x\ =x_i} = p_2'(y_{i\ j}, y_{i+1\ j} \text{ e } y_{i+2\ j})_{x\ =x_i} \qquad (4)$$

em que p1 e p2 são as equações de dois trechos de parábola adjacentes, pertencentes ao cabo **j**, em função das ordenadas *y* em três pontos.

O segundo tipo de restrições de ordem geométrica refere-se às limitações à curvatura dos cabos. A expressão geral de tais restrições de curvatura para o trecho **k** do cabo **j**, é a seguinte:

$$(1/R_{kj}) \leq L^s_{kj}) \qquad (5)$$

$$(1/R_{kj}) \leq L^i_{kj}) \qquad (6)$$

com,

R_{kj} - Raio de curvatura no trecho **k** do cabo **j**

L^s_{kj} - Limite superior do inverso do raio de curvatura no trecho **k** do cabo **j**

L^i_{kj} - Limite inferior do inverso do raio de curvatura no trecho **k** do cabo **j**

3. Optimização da Solução

Tal como para o caso da optimização do pré-esforço em vigas contínuas (1), o critério de optimização adoptado, nos tipos de problemas agora abordados, é o de minimizar o custo do pré-esforço aplicado, através da seguinte expressão:

$$\text{Minimizar } \mathrm{Z} = \sum_{j=1}^{M} C_j \times l_j \times \mathrm{P_j} \qquad (7)$$

onde Z representa o custo da solução, C_j é o custo unitário do pré-esforço no cabo **j**, l_j é o comprimento do cabo **j**, P_j o valor do pré-esforço aplicado e **M** é o número total de cabos de pré-esforço.

4. Organização do Programa Linear

4.1. Variáveis de projecto

Como já foi referido anteriormente, as secções a estudar em cada problema, terão que ser definidas previamente, devendo conter todas aquelas em que se produzem os esforços máximos. Na figura 3 representam-se incógnitas a determinar para o caso de um pórtico pré-esforçado, em que se admite um traçado do cabo parabólico na viga e rectilíneo em cada um dos pilares.

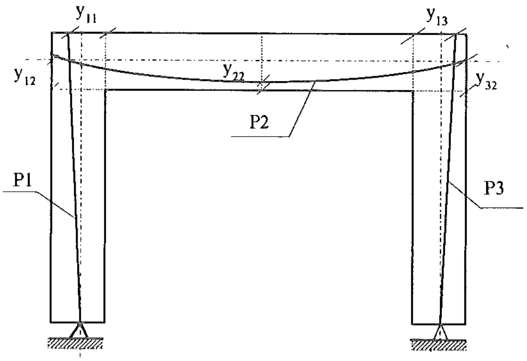


Fig. 3 - Incógnitas de um problema de optimização

Para que o programa matemático definido pela associação da condição (7) às restrições (1) a (6) seja linear escolheram-se para variáveis de projecto para expressar cada uma das condições envolvidas no referido programa, as seguintes:

$$\mathrm{X_j} = \mathrm{P_j} \qquad (8)$$

$$\mathrm{X_i} = \mathrm{y_{kj}} \times \mathrm{P_j} \qquad (9)$$

$$\mathrm{j} = 1,\ldots, \mathrm{M} \quad \mathrm{i} = \mathrm{M}+1,\ldots, \mathrm{N}$$

com **M** igual ao número de cabos de pré-esforço, **N** igual ao número total de variáveis de projecto e em que $\mathrm{y_{kj}}$ representa a ordenada **k**, associada ao cabo **j**.

4.2. Tensões nas secções de controlo

A expressão da tensão para uma determinada secção **i** quando actua na estrutura a combinação de acções de ordem s, apresenta a seguinte forma:

$$\sigma^i = \sum_{j=1}^{L} \frac{1}{A_i} \times X_j + \frac{1}{W_i} \times \sum_{j=M+1}^{N} m_{ij} \times X_j + \frac{M_{si}}{W_i} + \frac{N_{si}}{W_i} \qquad (10)$$

em que:

L - Número de cabos na secção.

A - Área da secção transversal.

W - Módulo de flexão da secção.

Msi - Momento na secção em estudo devido à combinação de acções exteriores s.

Nsi - Esforço axial na secção em estudo devido à combinação de acções exteriores s.

Todas as restrições de tensões do programa linear são formuladas de acordo com a expressão (10). A quantificação dos coeficientes m_{ij}, resulta da aplicação de um método análogo ao usado no modelo proposto para vigas contínuas [1] e cujos passos fundamentais são os seguintes:

a) Decompõe-se a acção equivalente ao pré-esforço de cada um dos cabos em acções elementares (expressas em função das incógnitas geométricas) dos tipos indicados no Quadro II.

b) Aplicam-se acções unitárias em conformidade com as acções equivalentes elementares e determinam-se os momentos provocados nas secções em estudo, por cada uma delas.

c) Quantificam-se os momentos devidos à aplicação do pré-esforço, em cada uma das secções em estudo, em função das variáveis de projecto, adicionando-se o efeito de cada uma das acções elementares.

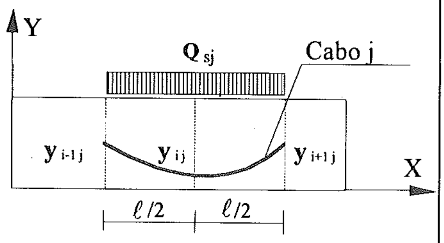
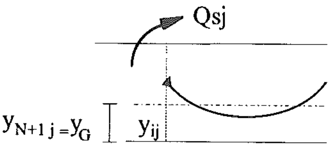
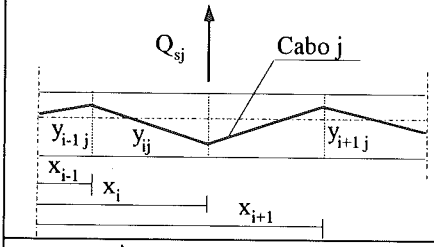
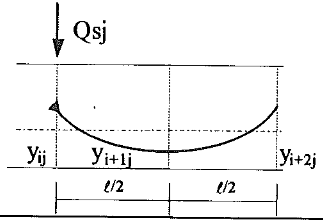
4.3. Restrições geométricas

Todas as restrições geométricas, podem ser expressas linearmente em função das variáveis de projecto escolhidas [1], não sendo introduzida qualquer dificuldade pelo facto de se considerarem vários cabos de pré-esforço. Os coeficientes relativos a ordenadas definidoras do cabo que não pertencem ao trecho considerado são tomados com valor nulo.

Quadro II - Acções elementares equivalentes ao pré-esforço

$$Q_{sj} = P_j \times \sum_{i=1}^{T} q_{si} \times y_{ij}$$

T - Nº de incógnitas geométricas associadas ao cabo j

ACÇÕES	COEFICIENTES		
 Y Q_{sj} Cabo j $y_{i-1\,j}$ y_{ij} $y_{i+1\,j}$ X ℓ/2 ℓ/2	$q_{s\,i-1} = \pm\frac{1}{\left(\frac{\ell}{2}\right)^2}$	$q_{s\,i} = \mp\frac{2}{\left(\frac{\ell}{2}\right)^2}$	$q_{s\,i+1} = \pm\frac{1}{\left(\frac{\ell}{2}\right)^2}$
	$q_{si} = 1$	$q_{s\,N+1} = -1$	
 Q_{sj} Cabo j $y_{i-1\,j}$ y_{ij} $y_{i+1\,j}$ x_{i-1} x_i x_{i+1}	$q_{s\,i-1} = \frac{1}{x_i - x_{i-1}}$	$q_{si} = \frac{1}{x_i - x_{i+1}} + \frac{1}{x_{i-1} -}$	$q_{s\,i+1} = \frac{1}{x_{i+1} - x_i}$
	$q_{s\,i} = \frac{3}{2\left(\frac{\ell}{2}\right)}$	$q_{s\,i+1} = -\frac{2}{\left(\frac{\ell}{2}\right)}$	$q_{s\,i+2} = \frac{1}{2\left(\frac{\ell}{2}\right)}$

4.4. Programa linear

As soluções óptimas dos problemas de estruturas pré-esforçadas, envolvendo diferentes cabos de pré-esforço são obtidas do seguinte programa linear [3]:

$$\textbf{Mínimo}\ Z = \sum_{j=1}^{M} C_j \times l_j \times X_j$$

Sujeito a:

$$\sum_{j=1}^{L} \frac{1}{A_i} \times X_j + \frac{1}{W} \times \sum_{j=M+1}^{N} m_{ij} \times X_j \le f - \frac{M_{si}}{W} - \frac{N_{si}}{A_i} \qquad \text{a)}$$

$$-(h - c_1) \times X_j + X_i \le 0 \qquad \text{b)}$$

$$- c_2 \times X_j + X_i \ge 0 \qquad \text{b)}$$

$$\frac{1}{\ell} \times X_{i-2} + \frac{-4}{\ell_1} \times X_{i-1} + \left(\frac{3}{\ell_1} + \frac{3}{\ell_2}\right) \times X_i + \frac{-4}{\ell_2} \times X_{i+1} + \frac{1}{\ell_2} + X_{i+2} = 0 \qquad \text{c)}$$

$$-L_{kj}^{s} \times X_j + \frac{X_i^k - 2X_{i+1}^k + X_{i+2}^k}{\alpha_k} \le 0 \qquad \text{d)}$$

$$-L_{kj}^{i} \times X_j + \frac{X_i^k - 2X_{i+1}^k + X_{i+2}^k}{\alpha_k} \le 0 \qquad \text{d)}$$

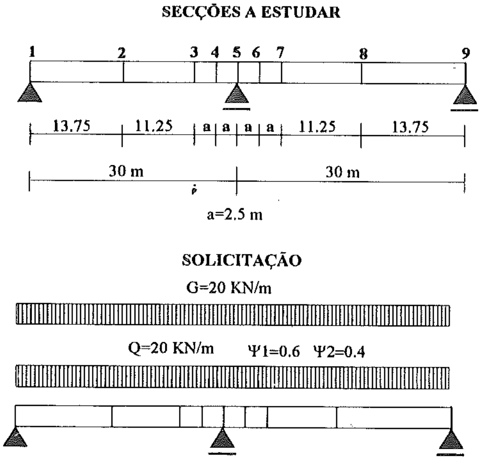
onde:

- **a)** As restrições de tensões, são impostas em todas as fibras extremas, das secções a estudar e para todas as combinações de acções.
- **b)** Os limites de ordenadas são impostos em todas as secções a verificar e para todos os cabos.
- **c)** A igualdade de tangentes é assegurada entre todos os trechos de parábolas adjacentes e/ou entre trechos rectos e trechos parabólicos, pertencentes aos diversos cabos de pré-esforço.
- **d)** São consideradas restrições impostas em todos os trechos, em que se pretenda limitar a curvatura no traçado dos cabos de pré-esforço.

5. Exemplos

5.1 Viga de dois tramos - 2 cabos de pré-esforço

O exemplo que se apresenta a seguir, refere-se a uma viga de dois tramos, em que se utilizam dois cabos de pré-esforço de trajectórias parabólicas. As características da secção transversal da viga assim como a solicitação exterior são apresentadas na figura 4.



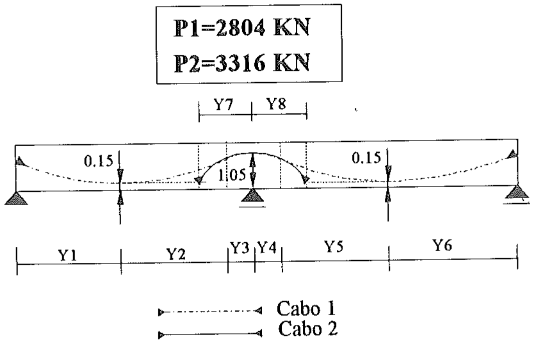
CABO	SECÇÕES
Cabo 1	1 2 4 5 6 8 9
Cabo 2	3 5 7

A=0.80 m2
I=0.106667 m4
Vs=0.40 m
Vi=0.80 m
Po=1.2 *Poo
Ambiente moderadamente agressivo
Recobrimento = 0.15 m

Fig. 4 - Viga contínua de dois tramos - Dois cabos parabólicos

No quadro que se segue, resumem-se as características principais do programa linear associado ao problema apresentado e cuja resolução conduziu aos resultados apresentados na figura 5.

Restrições de tensões	72
Restrições de limites de ordenadas	20
Restrições de igualdade de tangentes	2
	======
Número total de restrições	94
Número total de variáveis	12



PARÁBOLA	EQUAÇÃO	COMPRIMENTO
Y1	0.00417 x^2	14.00 m
Y2	0.00417 x^2	13.50 m
Y3	-0.02251 x^2	2.50 m
Y4	-0.02251 x^2	2.50 m
Y5	0.00417 x^2	13.50 m
Y6	0.00417 x^2	14.00 m
Y7	-0.03600 x^2	5.00 m
Y8	-0.03600 x^2	5.00 m

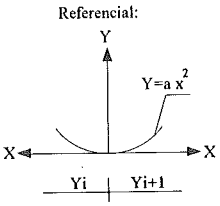


Fig. 5 - Viga de dois tramos - Solução

5.2 Pórtico pré-esforçado

O exemplo representado na figura 6, refere-se a um pórtico de um edifício, composto por um pilar e uma viga em consola. Procura-se determinar, para a solicitação caracterizada na figura, o valor do pré-esforço a aplicar admitindo-se uma trajectória parabólica na consola e rectilínea no pilar.

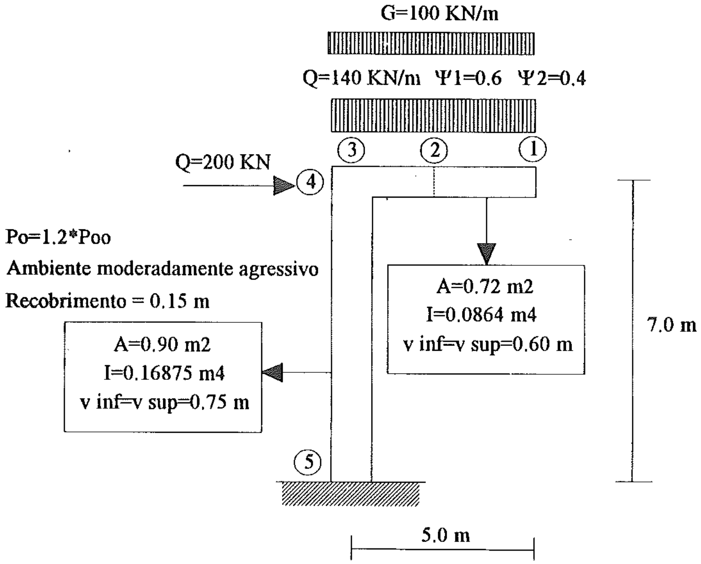


Fig. 6 - Pórtico pré-esforçado - 2 cabos

O programa linear gerado envolve o número de restrições e incógnitas apresentados no quadro que se segue.

Restrições de tensões	40
Restrições de limites de ordenadas	8
Restrições de igualdade de tangentes	0
	======
Número total de restrições	48
Número total de variáveis	6

A solução encontrada encontrada encontra-se representada na figura 7.

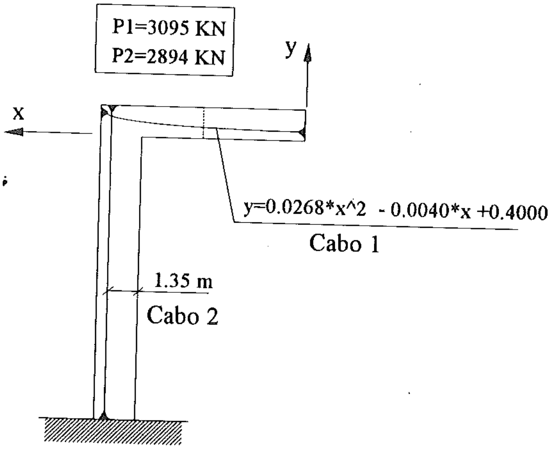


Fig. 7 - Pórtico pré-esforçado. Solução óptima

6. Referências

[1] Pinho, J.L.S. e Serra Neves, A.A (1982) - Optimização de Soluções Pré-esforçadas em Vigas Contínuas, 4º Encontro Nacional sobre Estruturas Pré-esforçadas, LNEC, Lisboa.

[2] Leonhardt, F. (1977) - Hormigon Pretensado Projecto y Construccion, Instituto Eduardo Torroja de la Construccion y del cemento, Madrid.

[3] Ramalhete, M; Guerreiro, J e Magalhães, A. (1984) - Programação Linear, Vol. 1, McGraw-Hill, Lisboa.

[4] Regulamento de Estruturas de Betão Armado e Pré-Esforçado (1983), Decreto-Lei n?349-C/83, Imprensa Nacional - Casa da Moeda, Lisboa.

[5] Regulamento de Segurança e Acções para Estruturas de Edifícios e Pontes (1983), Decreto-Lei nº235/83, Impresa Nacional - Casa da Moeda, Lisboa.

[6] Uri Kirsch (1972) - Optimum Design of Prestressed Beams, Computers & Structures, 2, 573-583.

[7] U. Kirsch and M.F Rubinstein (1970) - Optimum Prestressing by Linear Programming, UCLA Paper ENG-0670.

[8] Pinho, J.L.S.(1993) - Optimização de Soluções Pré-esforçadas em Estruturas Reticuladas e Placas Irregulares, Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto.

INSTRUÇÕES AOS AUTORES

Os autores que desejem submeter um artigo à Investigação Operacional devem enviar três cópias desse trabalho para:

Prof. Joaquim J. Júdice
Departamento de Matemática
Universidade de Coimbra
3000 Coimbra, Portugal

Os artigos devem ser escritos em Português ou Inglês. A primeira página deve conter a seguinte informação:

- Título do artigo
- Autor(es) e instituição(ões) a que pertence(em)
- Abstract (em inglês)
- Resumo
- Keywords (em inglês)
- Título abreviado

As figuras devem aparecer em separado de modo a poderem ser reduzidas e fotocopiadas. As referências devem ser numeradas consecutivamente e aparecer por ordem alfabética de acordo com os seguintes formatos:

Artigos: autor(es), título, título e número da revista (livro com indicação dos editores), ano, páginas.

Livros: autor(es), título, editorial, local de edição, ano.

Fotografia, Montagem
Impressão e Acabamentos
Tip.Nocamil
COIMBRA

ÍNDICE

apdio

Associação Portuguesa para o Desenvolvimento
da Investigação Operacional

CÉSUR - Instituto Superior Técnico - Avenida Rovisco Pais
1000 Lisboa - Telef. 80 74 55