Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 175 (2006) 1577–1587
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Issues in the implementation of the DSD algorithm for the traffic assignment problem

Pablo A. Lotito

METALAU Project, INRIA, Rocquencourt, BP 105, Le Chesnay Cedex F-78153, France
INRETS, Arcueil, France

Available online 11 April 2005

Abstract

In this paper we consider the practical implementation of the disaggregated simplicial decomposition (DSD) algorithm for the traffic assignment problem. It is a column generation method that at each step has to solve a huge number of quadratic knapsack problems (QKP). We propose a Newton-like method to solve the QKP when the quadratic functional is convex but not necessarily strictly. Our O(n) algorithm does not improve the complexity of the current methods but extends them to a more general case and is better suited for reoptimization and so a good option for the DSD algorithm. It also allows the solution of many QKP's simultaneously in a vectorial or parallel way.

Keywords: Large scale optimization; Non linear programming; Quadratic programming; Traffic; Network flows

1. Introduction

Given a transportation network, the traffic assignment problem consists in determining a flow that satisfies a given demand between certain pairs of origins and destinations and a certain equilibrium condition. The demand of transport can be fixed or variable with respect to travel times, this is the elastic demand case.

When the condition is that *no traveller can improve his travel time by unilaterally changing routes*, which is the Wardrop Principle (in Section 2 we give a mathematical formulation), the solution is said to be a user equilibrium (UE). This definition assumes that all the users of the network behave identically and have complete information about route costs. To relax that, the travel times can be defined as a deterministic term plus a random variable. Now the Wardrop principle has its stochastic version wich is called stochastic

E-mail address: pablo.lotito@inrets.fr

doi:10.1016/j.ejor.2005.02.029

user equilibrium (SUE) (see [15,8]). The word Static may be added to stress the fact that network conditions are constant, in contrast with dynamic user equilibriums (DUE) where demand and travel time parameters can vary with time (for a definition see [1]).

Under certain assumptions the UE problem becomes a convex optimization problem. These assumptions require that the network be strongly connected and the travel time on each arc be a positive function of the flow over that arc (separable case).

More general cases have been proposed, they consider that travel time functions depend on the flow over many arcs or there are different classes of flows that affect differently the travel time. The difficulty for the solution of this problems is that in general they are no more equivalent to an optimization problem but to a variational inequality. Under certain hypothesis this variational inequality can be solved with an iterative algorithm that at each step solve a standard UE.

The UE problem can appear also as a subproblem when considering bilevel problems where the lower level problem is a traffic assignment problem. Even if the UE may seem very simple to be realistic our interest in its numerical solution stems from the fact that to solve other problems, as the ones mentioned above, many instances of it must be efficiently solved. Efficiency will depend on the specific problem because sometimes we will need a fast solution less approximated and sometimes a very precise solution no matter the time.

Among the different algorithms to obtain the UE (see [13] for a review) the so called DSD, developed by Larsson and Patriksson and presented in [7], appears among the more efficient ones. This algorithm has as a subproblem, a family of quadratic knapsack problems, i.e., the minimization of a quadratic objective function subject to an only one linear constraint (the knapsack constraint). Clearly, the lowest the computation time of these subproblems the better the performance of the global problem. To solve them, the authors suggested to use the algorithm presented by Brucker in [2].

More precisely, we will call QKP the following problem:

$$\text{(QKP)} \qquad \begin{aligned} \min \quad & \sum_{i\in I} a_i x_i + \frac{1}{2} b_i x_i^2, \\ \text{s.t.} \quad & \sum_{i\in I} x_i = 1, \quad x_i \geqslant 0, \ \forall i \in I, \end{aligned} \tag{1}$$

where I is a finite set of indices.

If we define the functions $f_i(x)$ as $a_i x_i + \frac{1}{2} b_i x_i^2$ when $x \geqslant 0$ and $+\infty$ otherwise and replace the knapsack constraint by $\sum_{i\in I} x_i = y$, the QKP is equivalent to compute the inf-convolution of the functions f_i evaluated at $y = 1$. Using the Fenchel transform and some properties of the dual function (see [4] or [14]) we obtain an analytic formula for the QKP solution, $x_i = f_i^F(p)$. The value of p can be obtained solving the equation defined by the knapsack constraint, that is looking for a zero of the function $\sum_{i\in I} f_i^F(p) - 1$. A derivation of this formula in a more general case without the Fenchel transform is presented in the Section 3.1.

So far, the solution of the QKP is equivalent to search a zero of a function that in our case is piece-wise linear, convex and monotone. The method of Brucker performs a binary search on the index set of critical values of this function. This O(n) algorithm extends and improves the results obtained in [6] where an O($n \log n$) algorithm is presented to solve a less general problem. In [12] the authors present a *randomized* version with worst case second order complexity but with expected linear time and a small time constant. In [10] the authors develop four algorithms for the solution of a more general problem. Instead of a quadratic objective function they consider strictly convex, continuously differentiable functions. Their algorithms are designed to be suitable for parallel implementation. One of these algorithms can be considered similar to Brucker's, another can be seen as a Newton method to find a zero of the function mentioned above.

Based in those previous works we propose an algorithm that uses a Newton search directive and that can be used also in the case of non-strict convexity (all the fore-mentioned algorithms consider strictly convex