Pesquisa Operacional (2013) 33(2): 217-234

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope

GLOBAL OPTIMIZATION OF CAPACITY EXPANSION AND FLOW ASSIGNMENT IN MULTICOMMODITY NETWORKS

Ricardo Poley Martins Ferreira[1*], Henrique Pacca Loureiro Luna[2], Philippe Mahey[3] and Mauricio Cardoso de Souza[4]

Received December 9, 2011 / Accepted January 22, 2013

ABSTRACT. This paper describes an exact algorithm to solve a nonlinear mixed-integer programming model due to capacity expansion and flow assignment in multicommodity networks. The model combines continuous multicommodity flow variables associated with nonlinear congestion costs and discrete decision variables associated with the arc expansion costs. After establishing precise correspondences between a mixed-integer model and a continuous but nonconvex model, an implicit enumeration approach is derived based on the convexification of the continuous objective function. Numerical experiments on medium size instances considering one level of expansion are presented. The results reported on the performance of the proposed algorithm show that the approach is efficient, as commercial solvers were not able to tackle the instances considered.

Keywords: capacity expansion, flow assignment, global optimization, implicit enumeration, multicommodity flow problems.

1 INTRODUCTION

We consider a model for the joint problem of capacity expansion and flow assignment in multicommodity flow networks which takes into account congestion effects. The resulting optimization problem relies on the combination of two conflicting criteria: the expansion and the congestion cost functions. Capacity expansion cost functions are discrete as only a finite number of capacity sizes are available. On the other hand, congestion cost functions are nonlinear convex increasing functions as they try to capture queueing effects on the network [3, 4, 18, 23]. For

*Corresponding author
[1]Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais – UFMG, Brazil. Phone: +55(31) 3409-3503. E-mail: poley@demec.ufmg.br
[2]Instituto de Computação, Universidade Federal de Alagoas – UFAL, Maceió, Brazil.
E-mail: henrique.luna@pq.cnpq.br
[3]LIMOS UMR 6158 CNRS, Université Blaise Pascal, Clermont-Ferrand, France. E-mail: philippe.mahey@isima.fr
[4]Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais – UFMG, Brazil.
E-mail: mauricio@dep.ufmg.br

instance, Ishfaq & Sox [18], in the context of an intermodal hub network, deal with shipment delays due to limited resources at logistics hubs for measuring service performance. The authors conducted a study of strategies to integrate hub operation queuing model and the hub location-allocation model on a 25-city road-rail intermodal logistics network. Our aim is to study the general multicommodity flow problem under such assumptions, with no particular application in mind.

Typically, the two decision levels of capacity expansion and flow assignment are decoupled to treat the corresponding difficulties separately, and few works, at least those considering queueing effects, have been done on the joint optimization problem. Gerla & Kleinrock [14] stated the general Capacity and Flow Assignment problem (CFA), decomposed it into simpler subproblems, and then suggested a heuristic procedure that alternates a capacity assignment phase with a flow assignment phase. Similar approaches have been proposed in successive papers by Gavish and co-authors [11, 12] and Gerla *et al.* [15]. Most of that literature concerns heuristic procedures, but the paper by Mahey *et al.* [20] is an exception for this rule, showing that a generalized Benders decomposition method can find exact solutions for the CFA problem.

The capacity expansion problem is a special case of CFA where initial capacities are already installed on each arc of the network. Given a traffic requirement matrix between origin-destination pairs, the problem consists in jointly deciding which arc capacities, if any, should be expanded and the flow assignment leading to a feasible routing that minimizes expansion and congestion costs. Thus, the problem results in finding a trade-off between investment and routing costs.

Luna & Mahey [19] modeled the capacity expansion and flow assignment as a piecewise convex multicommodity flow problem. Congestion is modeled by a convex increasing function for a given capacity and, at given breakpoints, which represents the maximum tolerable congestion for the users (thus strictly lower than the available capacity), expansion to a higher capacity is decided, decreasing the marginal congestion cost in a discontinuous way. The combinatorial nature of the problem, related to arc expansion decisions, is therefore embedded in a continuous objective function that encompasses congestion and investment costs. The resulting objective function is continuous, but it is nonconvex and nonsmooth. Mahey and Souza [22] derived local optimality conditions for the model proposed in [19]. By exploiting complete optimality conditions for local minima, Souza *et al.* [25] give the convergence analysis of the negative-cost cycle canceling method. Remark that these former works only consider the case of simple expansions from one installed capacity to a new one. The case of the general expansion problem where several capacity expansion values are available for each arc is analyzed by Ferreira & Luna [7], where a method to find solutions with performance guarantee is introduced.

The contribution of this paper is to provide an algorithm to assure the global optimality of the problem, also including computational experiments to show the efficiency of the method to cope with the case of simple expansions. After showing the correspondences between the continuous model of Luna & Mahey and a mixed-integer model, we analyze the numerical behavior of an implicit enumeration algorithm. The discrete model is used to assign capacities to a subset of arcs and hence to define what is called a partial solution. The continuous model is used to compute, given a partial solution, a lower bound on the value of the best solution that can be obtained with the assigned capacities. The resulting procedure is tested on different types of networks and shown to be quite efficient to solve the mixed-integer nonlinear model.

2 RELATIONS BETWEEN THE CONTINUOUS AND MIXED-INTEGER MODELS

This section presents the network expansion model. The basic component of the model is a digraph $\mathcal{G} = (V, E)$ with n nodes and m arcs. Any kind of traffic between a given pair of nodes is treated as a separate commodity k. Let T be a $(n \times n)$ traffic requirement matrix such that t_{ij} is the traffic between origin i and destination j. We will consider the problem of deciding which arcs should be expanded from a given installed capacity c_0 to a greater capacity c_l while minimizing the total congestion and expansion costs.

Given a commodity k, we consider the set of directed paths P_k joining the corresponding origin and destination. Let x_{kp} be the amount of flow of commodity k through the path $p \in P_k$ and a_{kp} its arc-path incidence vector defined by

$$a_{kp}^{e} = \begin{cases} 1 & \text{if arc } e \text{ is used in path } p \text{ of commodity } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The vector x is composed by the component x_e which denotes the total flow on arc e, and also by the component x_{kp} which denotes the flow of commodity k routed through path p. These two components are related by

$$x_e = \sum_{k} \sum_{p \in P_k} a_{kp}^{e} x_{kp}.$$

The set of multicommodity flow vectors, denoted by $\mathcal{M}(T)$ can be described by the arc-path formulation, *i.e.*, for each commodity k flowing between nodes i and j, the active paths must satisfy

$$\sum_{p \in P_k} x_{kp} = t_{ij}.$$

That implicit formulation (as the paths are not known in advance) is generally preferred to the node-arc formulation where $x_e = \sum_k x_e^k$ and each x^k is a flow vector on $\mathcal{G}$ satisfying flow constraints for commodity k.

We assume now that for each arc e in the topology is assigned a positive capacity c_{0e} that is expandable to a larger capacity chosen among a given set of capacities $c_{1e} < \ldots < c_{Ne}$ at given fixed costs π_{le}, $l = 1, \ldots, N$. Let $\delta_{le} = c_{le} - c_{0e}$, $l = 1, \ldots, N$, be the increment of capacity to the l-th capacity value. The capacity expansion model will minimize the total congestion cost plus the expansion fixed costs. Let $\Phi(c_e, x_e)$ be the arc congestion function for a given capacity c_e. It is assumed that Φ is expressed in terms of monetary values and that it is convex smooth and increasing up to infinity on the interval $[0, c_e]$. A common choice is the Kleinrock's average delay function valid for M/M/1 queues which is proportional to $\frac{x_e}{c_e - x_e}$, see [3, 14].

A mixed integer model makes use of a binary variable $y_{le}, l = 1, \ldots, N, e \in E$, that assumes 1 if capacity of arc e is to be expanded from c_{0e} to c_{le} and 0 otherwise. We can now define a mixed integer nonlinear model for the capacity expansion problem (DCE):

$$\begin{aligned} \text{Minimize} \quad & \phi(x, y) = \sum_{e \in E} \left[\Phi(c_{0e} + \sum_{l=1}^{N} \delta_{le} y_{le}, x_e) + \sum_{l=1}^{N} \pi_{le} y_{le} \right] \\ \text{subject to} \quad & x \in \mathcal{M}(T) \\ & x_e \leq c_{0e} + \sum_{l=1}^{N} \delta_{le} y_{le}, \forall e \in E \\ & \sum_{l=1}^{N} y_{le} \leq 1, \forall e \in E \\ & y_{le} \in \{0, 1\}, \forall e \in E, l = 1, \ldots, N \end{aligned} \tag{2}$$

We will now study the relationship between (DCE) and a continuous model which does not make use of any boolean decision variables y (CCE):

$$\begin{aligned} \text{Minimize} \quad & f(x) = \sum_{e \in E} f_e(x_e) \\ \text{subject to} \quad & x \in \mathcal{M}(T) \end{aligned} \tag{3}$$

where we assume $\pi_{0e} = 0, \forall e \in E$, and $f_e(x_e) = \min\{\Phi(c_{le}, x_e) + \pi_{le}, l = 0, \ldots, N\}$.

Remarks.

1. Thanks to the feasibility assumption above and the fact that $\Phi(c_{Ne}, x_e) \to +\infty$ whenever $x_e \uparrow c_{Ne}$, we do not need any capacity constraint in the continuous model.

2. As shown on Figure 1, where the nonconvex resulting arc cost function of (CCE) is represented by a bold line, we denote by $\gamma_{(l-1)e}, l = 1, \ldots, N$ the breakpoint at which expansion occurs from $c_{(l-1)e}$ to c_{le}. The breakpoint can thus be interpreted as the capacity where congestion is such that the network manager is willing to pay for a new expansion. Thus, $\pi_{le} - \pi_{(l-1)e} = \Phi(c_{(l-1)e}, \gamma_{(l-1)e}) - \Phi(c_{le}, \gamma_{(l-1)e})$ is the new expansion cost converted in congestion cost units.

3. The arc cost function in (CCE) is continuous but nonconvex and nonsmooth at the breakpoints γ_{le}. It is shown in [19] how one can easily compute a lower bound on the optimal value of (CCE) by taking the convex envelope of each arc cost function.

Proposition 1. *If (x, y) is feasible for (DCE), x is feasible for (CCE); If x is feasible for (CCE), then there exists y such that (x, y) is feasible for (DCE); If one of both problems is infeasible, so is the other one.*

The proof is straightforward and it is omitted here for sake of simplicity, as the correspondence between (DCE) and (CCE) works with feasibility the same way it works with optimality, as developed below.

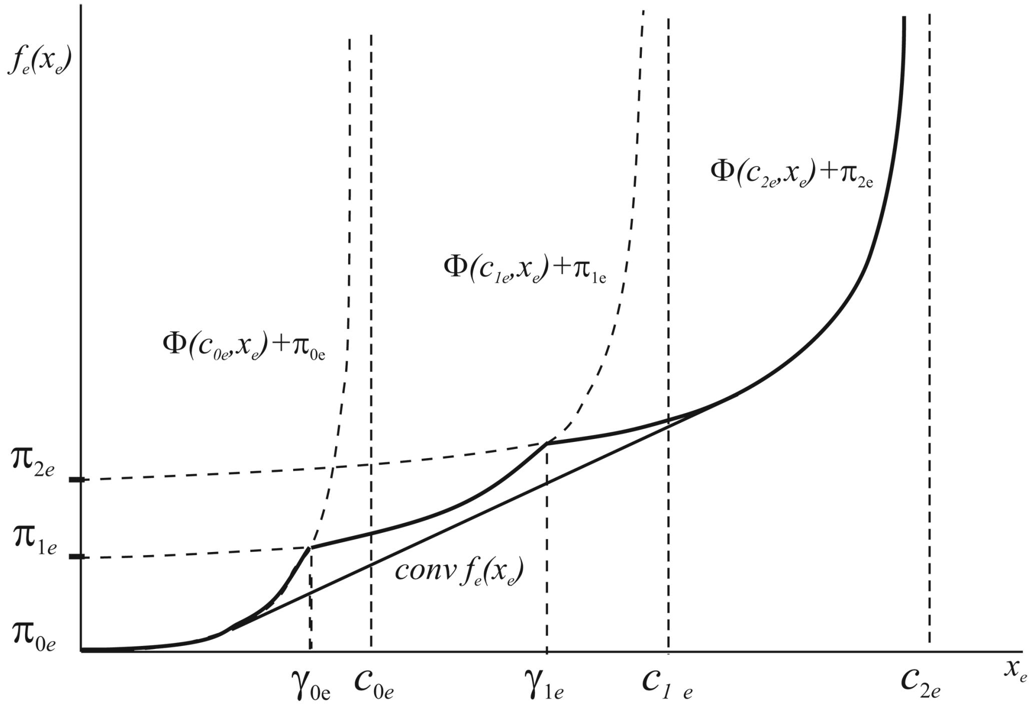

Figure 1 – The integrated function of congestion and expansion costs and its convex envelope $conv(f_e(x_e))$.

The following lemma is a direct consequence of the cost structure of (DCE).

Lemma 1. *Let* (x^*, y^*) *be an optimal solution of (DCE); then, we have the correspondences:*

$$\gamma_{(l-1)e} < x_e^* < \gamma_{le} \Longrightarrow y_{le}^* = 1,$$

$$and \;\; y_{qe} = 0, q = 1, \ldots, N, q \neq l$$

Moreover, if there exists an arc e with $x_e^* = \gamma_{(l-1)e}$*, then either* $y_{(l-1)e}^* = 1$ *and* $y_{le}^* = 0$ *or* $y_{(l-1)e}^* = 0$ *and* $y_{le}^* = 1$*, so the optimal solution is not unique.*

The two cases where x_e^* is not a breakpoint are straightforward. If $x_e^* = \gamma_{(l-1)e}$, we have:

$$\Phi(c_{(l-1)e}, \gamma_{(l-1)e}) + \pi_{(l-1)e} = \Phi(c_{le}, \gamma_{(l-1)e}) + \pi_{le}$$

which shows that the value of the arc cost function does not change whenever $y_{(l-1)e}^* = 1$ and $y_{le}^* = 0$ or, conversely, $y_{(l-1)e}^* = 0$ and $y_{le}^* = 1$. The correspondence between optimal solutions of (DCE) and (CCE) follows immediately.

Proposition 2. i) *If* (x^*, y^*) *is an optimal solution of (DCE), then* x^* *is optimal for (CCE) and the cost values are equal.*

ii) *If x^* is an optimal solution of (CCE), then (x^*, y^*) is optimal for (DCE) with:*

$$y^*_{le} \begin{cases} = 1 & \text{if} \quad \gamma_{(l-1)e} < x^*_e < \gamma_{le} \\ \in \{0, 1\} & \text{if} \quad x^*_e = \gamma_{(l-1)e} \\ \in \{0, 1\} & \text{if} \quad x^*_e = \gamma_{le} \\ = 0 & \text{if} \quad x^*_e \notin [\gamma_{(l-1)e}, \gamma_{le}] \end{cases} \tag{4}$$

$$y^*_{(l-1)e} + y^*_{le} = 1, \text{ if } x^*_e = \gamma_{(l-1)e} \tag{5}$$

where $l = 1, \ldots, N$, and the cost values are equal.

Finally, we would like to point out that the tight relationship between the optimal solutions of both models does not mean that they are equivalent. In general, the continuous model is not able to take into account additional constraints on the topology which, unlike, can be generally done by the y-variables. Nevertheless, we will mention a few common situations where it is possible to convert such constraints from (DCE) to (CCE):

a. Many models of network design require the same capacity on arcs (i, j) and (j, i) between two adjacent nodes i and j in $\mathcal{G}$. The orientation of arcs (i, j) and (j, i) being mainly to model the flow that pass from i to j and from j to i in a common physical link. As these two flows actually share a common link, it is required in such models symmetry between capacities of arcs (i, j) and (j, i). This is modelled in (DCE) by the constraint $y_{ij} = y_{ji}$ for some arc $e = (i, j)$. To obtain the same effect, we must add the following constraints for each l in (CCE):
$$(x_{ij} - \gamma_{lij})(x_{ji} - \gamma_{lji}) \geq 0$$

b. *Cutset constraints*: Let A be a subset of nodes of V and C_A the corresponding cutset. If subset A contains somehow crucial nodes for the network, the arcs in the cutset, *i.e.*, those with one extremity in A and the other in $V \setminus A$, may be considered bottleneck arcs as they are the only to carry flow between nodes in A and nodes in $V \setminus A$. Thus, it may be of interest forcing the subset A to be connected to the other nodes by at least one expanded arc. This is modelled in (DCE) by the constraint $\sum_{e \in C_A} y_{le} \geq 1$, which is equivalent in (CCE) to:
$$\max_{e \in C_A} \frac{x_e}{\gamma_{le}} \geq 1$$

Observe that both constraints derived in a. and b. define polyhedral nonconvex regions of $\mathbb{R}^m$. Such a situation could not be treated in the following approach, that requires convexity to assure global optimality.

3 GLOBAL OPTIMIZATION STRATEGY

In this section, we follow the exposition provided by Geoffrion [13] to describe the application of an implicit enumeration algorithm to solve the capacity expansion and flow assignment problem. Geoffrion [13] presents, including his own developments, an unified approach of the works of Balas [1] and Glover [16] on the implicit enumeration scheme. See also Balas [2]. We consider that each arc e is expandable from an installed capacity c_{0e} to a capacity c_{1e}, although the procedure can be generalized to deal with more than one possibility of expansion.

The implicit enumeration algorithm combines information from both discrete and continuous models. A partial solution S defines the capacities of a subset $\bar{E}$ of arcs. Here, the discrete model is used to assign capacities to the arcs of $\bar{E}$. According to the notational convention introduced in [13], for each discrete variable associated to an arc in $\bar{E}$, the symbol e (resp. $-e$) denotes $y_e = 1$ (resp. $y_e = 0$). The discrete variables associated to arcs not in $\bar{E}$ are called free. As an example, suppose a small network with five arcs and a partial solution $S^t = \{1, 3, -5\}$. In this example, $y_1 = 1$, $y_3 = 1$, $y_5 = 0$, and y_2 and y_4 are free. A completion of a partial solution S is defined as a solution that is determined by S together with a binary specification of the values of the free variables. It is said to be a feasible completion if the assignment of values to the binary variables leads to a feasible solution. The four possible completions for S^t in the above example are $\{1, 3, -5, 2, 4\}$ where the free variables assume $y_2 = 1$ and $y_4 = 1$; $\{1, 3, -5, -2, 4\}$ where $y_2 = 0$ and $y_4 = 1$; $\{1, 3, -5, 2, -4\}$ where $y_2 = 1$ and $y_4 = 0$; and $\{1, 3, -5, -2, -4\}$ where $y_2 = 0$ and $y_4 = 0$.

A key feature of implicit enumeration is the ability to generate information that can be used to exclude all the completions of a partial solution S from further consideration. Here, the continuous model is used either to provide a lower bound on the value of the best feasible completion of S, *i.e.*, a feasible completion that minimizes the objective function among all feasible completions of S, or to show that S has no feasible completion. To do this, we solve a convex multicommodity flow problem P_S:

$$
\begin{aligned}
\text{Minimize} \quad & z_S = \sum_{e \in \bar{E}} [\Phi(c_{0e} + \delta_e \bar{y}_e, x_e) + \pi_e \bar{y}_e] \\
& + \sum_{e \in E \setminus \bar{E}} conv(\min\{\Phi(c_{0e}, x_e), \Phi(c_{1e}, x_e) + \pi_e\}) \\
\text{subject to} \quad & x \in \mathcal{M}(T) \\
& x_e \leq c_{0e} + \delta_e \bar{y}_e, \forall e \in \bar{E}
\end{aligned}
\tag{6}
$$

where $\bar{y}_e$, $e \in \bar{E}$, is fixed at partial solution S and $conv(f_e(x_e))$ is the convex envelope of function $f_e(x_e)$, *c.f.*, Figure 1. Any efficient algorithm designed for convex multicommodity flow problems (see [24] for instance) can be employed to solve P_S. The value z_S^* of the optimum solution of P_S is a lower bound on the value of the best completion of S.

An upper bound is always possible to be derived if P_S has an optimum solution. Let $\bar{x}$ be an optimum solution of P_S, then

$$
\bar{z}_S = \sum_{e \in \bar{E}} \left[\Phi(c_{0e} + \delta_e \bar{y}_e, \bar{x}_e) + \pi_e \bar{y}_e\right] + \sum_{e \in E \setminus \bar{E}} \min \left\{\Phi(c_{0e}, \bar{x}_e), \Phi(c_{1e}, \bar{x}_e) + \pi_e\right\}
$$

is an upper bound. Remark that for feasible problems, an upper bound is derived even if $S = \emptyset$. We store the best upper bound $\bar{z}$ found during the search as the incumbent. If a partial solution

S leads to an upper bound that improves upon the incumbent, then it replaces the latter as the new incumbent. If the lower bound z^*_S is greater than or equal to $\bar{z}$ or P_S is infeasible, then S is fathomed. And in this case all completions of S have been implicited enumerated as they can be excluded from further consideration.

A partial solution is said to be nonredundant if it cannot generate a completion equal to one generated with a previous solution that was fathomed. Geoffrion [13] gives a procedure for generating nonredundant partial solutions that terminates fathoming all feasible solutions. By starting with $\bar{E} = \emptyset$, *i.e.*, S^0 has no capacity previously assigned to any arc, P_{S^0} gives a lower bound $z^*_{S^0}$ to the optimum value and a first upper bound $\bar{z}_{S^0}$ taken as the incumbent $\bar{z}$. Then, the procedure augment the partial solution by assigning a capacity to an arc at a time. Now suppose a partial solution S^t is fathomed. Nonredundancy is achieved by having at least one element of subsequent partial solutions complementary to S^t. The last element that was added to S^{t-1} to generate S^t is then underlined and changed to its complement. In the above example, the sequence would be $S^1 = \{1\}$, $S^2 = \{1, 3\}$, $S^3 = \{1, 3, -5\}$. If S^3 could be fathomed, then the next partial solution in the sequence would be $S^4 = \{1, 3, \underline{5}\}$, which means change capacity assigned to arc 5 from $c_{0,5}$ in S^3 ($y_5 = 0$) to $c_{1,5}$ in S^4 ($y_5 = 1$). If S^4 could also be fathomed, then S^5 would be $S^5 = \{1, \underline{-3}\}$. Otherwise, S^5 would be generated by assigning a value to a free variable and by adding it to S^4, for instance $S^5 = \{1, 3, \underline{5}, -2\}$. After fathom a solution S^t, the procedure locates the rightmost element of S^t that is not underlined. If there is none, then end with the stored incumbent as being the optimal solution. Otherwise, replace this element by its complement underlined $e \rightarrow \underline{-e}$ and delete all elements that are on the right. In the example, if $S^5 = \{1, 3, \underline{5}, -2\}$ and also $S^6 = \{1, 3, \underline{5}, \underline{2}\}$ could be both fathomed, the next solution generated would be $S^7 = \{1, \underline{-3}\}$.

We now discuss some strategies to choose an arc $e \in \bar{E} \setminus E$ with which augment a partial solution S^t that cannot be fathomed. Note we chose an arc among those such that $conv(f_e(\bar{x}_e)) < f_e(\bar{x}_e)$. Such strategies rely on the flow distribution $\bar{x}$ of an optimum solution of P_{S^t}. Various strategies have been tested:

- assign c_{1e} to the arc e with the highest value $f_e^{'+}(\bar{x}_e)$ of the right partial derivative of arc cost function f_e with respect to $\bar{x}_e$.
- assign c_{0e} to the arc e with the lowest value $f_e^{'-}(\bar{x}_e)$ of the left partial derivative of arc cost function f_e with respect to $\bar{x}_e$.
- the arc inducing the highest value of z^*_S when P_S is defined for $\bar{E} \cup \{e\}$ assigning c_{0e} to e if $\bar{x}_e \leq \gamma_e$ and c_{1e} otherwise;
- the arc with the highest flow value;

The last rule was found to be the most efficient, and was adopted.

We employ algorithms proposed in [7], which have guaranteed performance, to find a good initial upper bound $\bar{z}_0$. Those algorithms find a feasible solution and then gradually reduce the objective value of the obtained solution until no better solution can be found. They are based in two phases: a common first phase where a multicommodity flow problem taking the convex envelope of arc cost functions is solved and a lower bound is found; and a second phase where the obtained routing is used as starting point for switching methods between capacity assignment and the application of a local search algorithm until no more improvement occurs.

The pseudo-code of the implicit enumeration algorithm is as follows:

Implicit enumeration procedure

Step 1 – Initialize with $S^0 = \emptyset$. Set $t = 0$ and $\bar{z} = \bar{z}_0$.

Step 2 – Solve P_{S^t} to compute the lower bound $z^*_{S^t}$ to the best completion of S^t. If P_{S^t} is feasible, then obtain the respective upper bound $\bar{z}_{S^t}$ and check to update $\bar{z}$.

Step 3 – If it is possible to fathom S^t, *i.e.*, P_{S^t} is infeasible or $z^*_{S^t} \geq \bar{z}$, then go to Step 5. Else go to Step 4.

Step 4 – Augment S^t by adding a free variable y_e with e or $-e$ to obtain S^{t+1}. Set $t = t + 1$ and return to Step 2.

Step 5 – Locate the rightmost element of S^t not underlined. If none exists, then stop. Else change such element by its complement underlined $e \rightarrow \underline{-e}$ and delete all elements to the right. Set $t = t + 1$ and return to Step 2.

4 NUMERICAL TESTS

Numerical tests were performed to analyze the numerical behavior of the proposed algorithm and the influence of the parameters on its performance. Three different network topologies were used for the computational tests: the C-NET introduced in [24]; the RING introduced in [11]; and the NTS100 generated using a special program driver [6, 7]. For the sake of illustration, Figure 2 and 3 present the topologies of networks C-NET and NTS100, respectively. Table 1 gives the main characteristics of the three networks used in our numerical experiments. Given a root node r, the hop-depth of a node $i \in V \setminus \{r\}$ is the number of arcs in the path between r and i that has minimum length. The hop-based diameter of the graph is the largest hop-depth among the nodes of the network. C-NET and RING are full duplex networks, *i.e.*, flow can be sent between two nodes in both directions and simultaneously.

Two sets of tests were made with these topologies:

- The first set concerns the C-NET network [24]. The aim is to verify the influence of the traffic throughput increase and the capacity expansion factor (c_{1e}/c_{0e}) on the number of iterations performed by the implicit enumeration algorithm. Results are compared with the ones obtained solving the mixed integer nonlinear formulation of the problem (DCE) using two commercial solvers: BARON and LINDOGlobal [5, 17, 27] and modeling the problem with GAMS [10]. The problems were solved with help of the NEOS Server [5, 17];

– The second test set was performed on the other two topologies RING and NTS100. The aim is to assess the effectiveness of the proposed implicit enumeration algorithm to solve the capacity expansion problem with different scenarios of larger networks and heterogeneous traffic requirement demands.

The computation of the lower bound was performed by a specialized Flow Deviation algorithm for the convex multicommodity flow problem. That algorithm has been shown to be efficient since the early work of Fratta *et al.* [9], but it is generally known to become very slow when the algorithm approaches the optimal solution (see [3] for instance). To accelerate its convergence, a parallel tangent procedure (PARTAN, see [8]) was introduced in the direction finding step. The algorithm was coded in C.

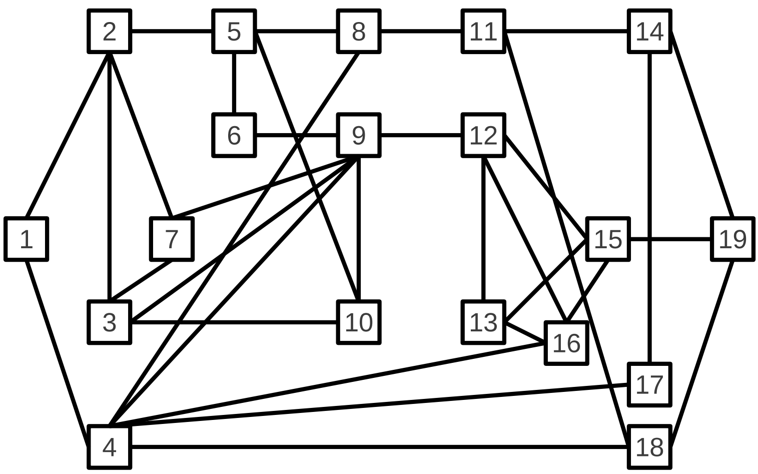

Figure 2 – The C-NET network with 19 nodes and 34 arcs.

Table 1 – Characteristics of the test networks.

Network ID	nodes	arcs	OD-pairs	aver. node degree	*hop-based* diameter
	n	m	K	$2m/n$	
C-NET	19	34	38	3.36	4
RING	32	60	496	3.75	6
NTS100	100	187	2000	3.74	11

The following notations were used:

- ϕ^* is the global optimal value;
- $\ddot{\phi}$ optimal value of the convexified problem;
- ϕ_{pg} is the solution obtained by the performance guaranteed algorithms [7];

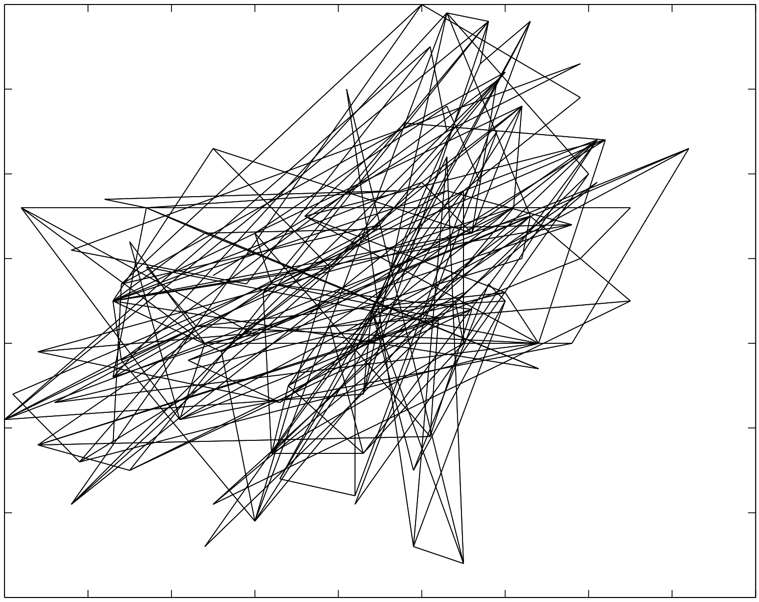

Figure 3 – NTS100 network topology.

- ϕ_{Baron} is the solution obtained with BARON [27];
- ϕ_{LG} is the solution obtained with LINDOGlobal [5, 17];
- N_{ggo} is the number of partial solutions enumerated to guarantee the global optimality;
- N_o is the number of partial solutions enumerated until the optimal solution is found be not necessarily proven to be optimal;
- N_{Baron} is the number of visited nodes made by the Baron solver;
- N_{LG} is the number of visited nodes made by the LINDOGlobal solver;
- $\alpha = \frac{\phi^*}{\ddot{\phi}}$;
- $\alpha_{pg} = \frac{\phi_{pg}}{\ddot{\phi}}$.

4.1 First test set

In these experiments, the congestion cost is derived from the Kleinrock's average delay function for M/M/1 queueing networks and defined by $\rho \frac{x_e}{c_e - x_e}$. We fix $\rho = 1$, see [11]. All arcs have the same initial capacity. And the expansion cost π_{1e}, for each arc $e \in E$, is given, for each value of the parameter $\gamma = \frac{\gamma_{0e}}{c_{0e}}$ equal to 0.7 and 0.9, by

$$\pi_{1e} = \rho \gamma c_{0e} \frac{(c_{1e} - c_{0e})}{(c_{0e} - \gamma c_{0e})(c_{1e} - \gamma c_{0e})}$$

Tables 2 to 5 present the results for the C-NET network varying traffic requirement demands, parameter γ and the available capacity for expansion.

The results show that the algorithm is rather sensitive to the capacity expansion factor. When $c_{1e} = 4c_{0e}$ the number of iterations increase significantly, as observed for N_{ggo} and N_o in Tables 4 and 5 in contrast with N_{ggo} and N_o in Tables 2 and 3. Increasing γ affects the efficiency of the algorithm. Both effects can be illustrated by the values of α and α_{pg}.

Table 2 – Network C-NET, problem D38510.70.

$\ddot{\phi}$	ϕ^*	ϕ_{pg}	α	α_{pg}	N_{ggo}	N_o	Demand
13.64	13.75	13.86	1.01	1.01	26	24	0.50
37.68	39.21	39.43	1.04	1.05	206	178	1.00
68.78	70.59	71.54	1.02	1.04	152	66	1.50
181.94	184.19	184.99	1.01	1.02	146	110	2.00

Number of commodities 38, $c_0 = 5$, $c_1 = 10$, $\gamma = 0.70$.

Table 3 – Network C-NET, problem D38510.90.

$\ddot{\phi}$	ϕ^*	ϕ_{pg}	α	α_{pg}	N_{ggo}	N_o	Demand
13.64	13.75	13.86	1.01	1.02	0	0	0.50
56.45	56.91	59.67	1.01	1.06	18	16	1.00
135.06	140.21	144.10	1.04	1.07	92	27	1.50
291.38	301.38	304.86	1.03	1.05	208	112	2.00

Number of commodities 38, $c_0 = 5$, $c_1 = 10$, $\gamma = 0.90$.

Table 4 – Network C-NET, problem D38520.70.

$\ddot{\phi}$	ϕ^*	ϕ_{pg}	α	α_{pg}	N_{ggo}	N_o	Demand
13.64	13.75	13.86	1.01	1.02	168	161	0.50
24.77	31.54	34.99	1.27	1.41	2893	272	1.00
37.38	45.01	49.35	1.21	1.32	6194	968	1.50
50.82	59.60	63.02	1.17	1.24	8026	2048	2.00

Number of commodities 38, $c_0 = 5$, $c_1 = 20$, $\gamma = 0.70$.

Table 5 – Network C-NET, problem D38520.90.

$\ddot{\phi}$	ϕ^*	ϕ_{pg}	α	α_{pg}	N_{ggo}	N_o	Demand
13.64	13.75	13.86	1.01	1.02	4	2	0.50
42.05	49.31	54.41	1.19	1.29	286	268	1.00
75.52	95.14	112.54	1.26	1.49	2672	2248	1.50
110.08	133.57	151.42	1.22	1.37	2944	317	2.00

Number of commodities 38, $c_0 = 5$, $c_1 = 20$, $\gamma = 0.90$.

As the efficiency of the algorithm depends on the quality of the lower bound to exclude completions of a partial solution from further consideration, the weakest is the convexification bound, the biggest is the number of solutions enumerated. Observe that the global optimal solution was reached in many instances, some of them well before $N_{ggo} \gg N_o$.

Table 6 shows the results obtained solving the mixed integer nonlinear formulation of the problem (DCE) using two commercial solvers: BARON and LINDOGlobal [5, 17] and the results of the proposed algorithm. The Baron and the LINDOGlobal solvers were not able to guarantee the optimality of the results within the adopted time limits and or the number of iterations limits. They were not able to obtain the guaranteed global optimum of the C-NET problem. The LINDOGlobal solver uses linear approximations of the original problem which explains the large number of iterations within the limits of runtime.

Table 6 – Network C-NET, continuous *versus* discrete formulations results.

c_0	c_1	γ	$\ddot{\phi}$	ϕ^*	ϕ_{Baron}	ϕ_{LG}	N_{ggo}	N_{Baron}	N_{LG}	Demand
5	10	0.70	13.64	13.75	13.75	13.75	26	1000	16744566	0.50
5	10	0.70	37.68	39.21	39.61	40.45	206	1000	8920413	1.00
5	10	0.70	68.78	70.59	70.60	71.50	152	1000	12103300	1.50
5	10	0.70	181.94	184.19	186.54	186.77	146	1000	8069258	2.00
5	10	0.90	13.64	13.75	13.75	13.75	0	1000	16890727	0.50
5	10	0.90	56.45	56.91	56.91	57.80	18	1000	8103096	1.00
5	10	0.90	135.06	140.21	140.21	140.21	92	1000	12034430	1.50
5	10	0.90	291.38	301.38	301.82	312.33	208	1000	8303808	2.00
5	20	0.70	13.64	13.75	13.75	13.75	168	10000	14855559	0.50
5	20	0.70	24.77	31.54	31.54	34.35	2893	10000	5799308	1.00
5	20	0.70	37.38	45.01	45.01	47.07	6194	10000	6564114	1.50
5	20	0.70	50.82	59.60	59.60	62.03	8026	10000	8239060	2.00
5	20	0.90	13.64	13.75	13.75	13.75	4	10000	16982685	0.50
5	20	0.90	42.05	49.31	49.31	57.20	286	(7200s)	6236884	1.00
5	20	0.90	75.52	95.14	101.01	127.01	2672	(7200s)	5670491	1.50
5	20	0.90	110.08	133.57	133.57	151.41	2944	(7200s)	7664995	2.00

Number of commodities 38, $c_0 = 5$, $c_1 = 20$, $\gamma = 0.90$.
For the problems with $c_1 = 10$, $N_{Baron} \leq 1000$, for the problems with $c_1 = 20$, $N_{Baron} \leq 10000$.
Maximum time available to solve the problems using BARON was 7200 seconds, and using LINDOGlobal was 4000 seconds.
All problems were solved in less than 3600 seconds using the proposed algorithm.

4.2 Second set of tests

The second set of tests concern the RING and the NTS100 topologies with heterogeneous traffic requirement demands. The congestion costs, as in the precedent set of experiments, is given by $\rho \frac{x_e}{c_e - x_e}$. The following scheme was adopted for all these experiments:

- An initial feasible solution is computed using algorithms proposed in [7].
- The initial demand between each OD-pair was fixed to 10 for the RING network and to 3 for the NTS100 network. The installed capacities are given in Table 7.

- In a first set of instances, the demand is uniformly multiplied by a throughput factor of 20%, 40%, 60%, 80% and 100% until capacity expansion becomes economically interesting.

– In the second set, the demand increase is no more uniform: half of the demands receive the same throughput factor, but 25% receive 50% more, and 25% receive 25% less. The results are exposed in Tables 8 to 11.

– An initial distribution of the increase of demand between the OD-pairs was defined randomly and fixed for each instance.

– The parameter ρ used to compute congestion costs is set to 500, 1000, 5000, and 10000.

Table 7 – Available capacities and costs.

Capacity c	Installation cost S_c	Distance cost D_c
2	1750	40
10	2800	50
34	4800	55
155	10000	80
300	14000	90
622	21000	120
922	35000	210

$$\pi^c = S^c + D^c t_{ij}.$$

A small number of iterations was needed to solve the proposed instances. Considering the complexity of the studied problem (CNET $= 2^{34} =$ 17,179,869,184 solutions, RING $= 2^{60}$ solutions, NTS100 $= 2^{187}$ solutions), one can evaluate the effectiveness of the proposed method. The second set of test problems was easier to be solved. This can be explained by noting that for these instances, the quality of the lower bound obtained by solving the problem with the convex envelope of the integrated function of congestion and expansion costs, *c.f.*, Figure 1, is very good.

At each iteration of the implicit enumeration algorithm a convex multicommodity flow problem is solved. A convex multicommodity flow problem can be difficult, especially if feasibility problems appear when an arc flow is close to the arc capacity. The decision to apply the Partan was primarily due to the gain in speed in solving problems is adequate for a precision of about 1% or better. Each multicommodity flow subproblem spent from a second fraction to several minutes.

All test problems have many local optima with objective function value close or equal. The implicit enumeration depth search approach has modest memory requirements. It needs to store only the path from the root node to the leaf node in the search tree. As the problem has many solutions, the depth search has a good chance of finding an optimal solution after exploring a small portion of the entire search space. The proposed algorithm can be parallelized and we believe that a good scalability can be get.

Table 8 – RING network and uniform demand increase.

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	ϕ_{pg} [\$] $\times 10^6$	α	α_{pg}	Aver. increase of demand %
1000	3.69	3.71	3.71	1.00	1.00	10
1000	3.84	3.91	3.92	1.01	1.02	20
1000	4.05	4.11	4.15	1.01	1.02	40
1000	4.35	4.39	4.49	1.01	1.01	60
1000	4.70	4.76	4.76	1.01	1.01	80
1000	5.13	5.13	5.13	1.00	1.00	100
5000	4.22	4.26	4.26	1.01	1.01	10
5000	4.47	4.50	4.50	1.00	1.00	20
5000	4.79	4.84	4.84	1.01	1.01	40
5000	5.17	5.23	5.23	1.01	1.01	60
5000	5.62	5.68	5.69	1.01	1.01	80
5000	6.31	6.35	6.37	1.01	1.01	100
10000	4.61	4.62	4.62	1.00	1.00	10
10000	4.95	5.00	5.00	1.01	1.01	20
10000	5.36	5.42	5.42	1.01	1.01	40
10000	5.84	5.87	5.87	1.00	1.00	60
10000	6.38	6.42	6.42	1.00	1.00	80
10000	7.03	7.08	7.10	1.01	1.01	100

Table 9 – RING network and heterogeneous demand increase.

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	ϕ_{pg} [\$] $\times 10^6$	α	α_{pg}	N_o	N_{ggo}	Aver. increase of demand %
500	3.70	3.82	3.85	1.01	1.02	32	138	25
500	4.06	4.14	4.27	1.025	1.05	84	98	50
500	4.88	4.93	4.93	1.01	1.00	2	4	100
1000	3.89	3.97	3.97	1.01	1.02	4	265	25
1000	4.21	4.33	4.33	1.01	1.02	1	2	50
1000	5.08	5.12	5.13	1.00	1.01	1	2	100
5000	4.01	4.02	4.03	1.01	1.00	5	555	25
5000	4.97	5.02	5.02	1.01	1.01	1	2	50
5000	6.30	6.37	6.37	1.01	1.01	1	2	100
10000	4.87	4.91	4.92	1.01	1.01	6	365	25
10000	5.59	5.64	5.64	1.01	1.01	3	1	50
10000	7.06	7.07	7.07	1.00	1.01	5	1	100

The proposed algorithm can be further specialized if more information about the problems are considered. For example, in real scenarios budget and operational constraints may restrict the set of candidate arcs for expansion. The proposed procedure can also be used for capacity reduction problems.

Table 10 – RING network and heterogeneous demand increase.

ρ	N_o	N_{ggo}	Aver. increase of demand %
500	32	138	25
500	84	98	50
500	2	4	100
1000	4	265	25
1000	1	2	50
1000	1	2	100
5000	5	555	25
5000	1	2	50
5000	1	2	100
10000	6	365	25
10000	3	1	50
10000	5	1	100

Table 11 – NTS100 network and heterogeneous demand increase.

ρ	$\ddot{\phi}$ [\$] $\times 10^6$	ϕ^* [\$] $\times 10^6$	ϕ_{pg} [\$] $\times 10^6$	α	α_{pg}	N_o	N_{ggo}	Aver. increase of demand %
500	3.70	3.82	3.85	1.01	1.02	32	138	25
500	4.06	4.14	4.27	1.025	1.05	84	98	50
500	4.88	4.93	4.93	1.01	1.00	2	4	100
1000	3.89	3.97	3.97	1.01	1.02	4	265	25
1000	4.21	4.33	4.33	1.01	1.02	1	2	50
1000	5.08	5.12	5.13	1.00	1.01	1	2	100
5000	4.01	4.02	4.03	1.01	1.00	5	555	25
5000	4.97	5.02	5.02	1.01	1.01	1	2	50
5000	6.30	6.37	6.37	1.01	1.01	1	2	100
10000	4.87	4.91	4.92	1.01	1.01	6	365	25
10000	5.59	5.64	5.64	1.01	1.01	3	1	50
10000	7.06	7.07	7.07	1.00	1.01	5	1	100

5 CONCLUSION

We have applied successfully an adapted implicit enumeration scheme to a mixed-integer nonlinear model for the capacity expansion and flow assignment in multicommodity networks. The key fact to explain the good performance of the method is the use of a reliable bounding procedure at each node of the enumeration tree, which is based on the convex hull of the equivalent continuous objective function.

The literature has scarce approaches to find exact solution of the discrete capacity expansion and flow assignment problem. Basically two lines of approaches have been adopted: Benders decomposition [20] and methods based on difference of convex functions – DC programming [21]. Recently a conic quadratic formulation was proposed [26]. The results show that the implicit enumeration algorithm proposed here can be considered as an alternative available to solve to global optimality such large scale problems.

Future research is to be done on the extension of the proposed approach to deal with multiple choices of available capacities for expansion on each arc. It remains to show that lower bounds obtained with the convex envelope of the integrated function of expansion and congestion costs remain sharp in the presence of multiple expansions.

ACKNOWLEDGMENTS

This work has been supported by CNPQ (305164/2010-4) and (473763/2011-7).

REFERENCES

[1] BALAS E. 1965. An additive algorithm for solving linear programs with zero-one variables. *Operations Research*, **13**: 517–546.

[2] BALAS E. 1966. Discrete programming by the filter method. *Operations Research*,**13**: 915–955.

[3] BERTSEKAS DP & GALLAGER RG. 1987. Data Networks, Prentice-Hall.

[4] DE CAMARGO RS, MIRANDA JR G, FERREIRA RPM & LUNA HP. 2009. Multiple allocation hub-and-spoke network design under hub congestion. *Computers and Operations Research*, **36**: 3097–3106.

[5] CZYZYK J, MESNIER M & MOR J. 1998. The NEOS Server. *IEEE Journal on Computational Science and Engineering*, vol. 5.

[6] DOAR MB & NEXION A. 1996. A better model for generating test networks. *Proceedings of Globecom '96*, pp. 86–93.

[7] FERREIRA RPM & LUNA HPL. 2003. Discrete capacity and flow assignment algorithms with performance guarantee. *Computer Communications*, **26**: 1056–1069.

[8] FLORIAN M, GULAT J & SPIESS H. 1987. An efficient implementation of the PARTAN variant of the linear approximation method for the network equilibrium problem. *Networks*, **17**: 319–339.

[9] FRATTA L, GERLA M & KLEINROCK L. 1973. The flow deviation method: an approach to store-and-forward communication network design. *Networks*, **3**: 97–133.

[10] www.gams.com, consulted on september 19, 2012.

[11] GAVISH B & NEUMANN I. 1989. System for routing and capacity assignment in computer communication networks. *IEEE Transactions on Communications*, **37**: 360–366.

[12] GAVISH B & ALTINKEMER K. 1990. Backbone network design tools with economic tradeoffs. *ORSA Journal on Computing*, **2**: 236–252.

[13] GEOFFRION AM. 1967. Integer programming by implicit enumeration and Balas' method. *SIAM Review*, **9**: 178–190.

[14] GERLA M & KLEINROCK L. 1977. On the topological design of distributed computer networks. *IEEE Transactions on Communications*, **25**: 48–60.

[15] GERLA M, MONTEIRO JAS & PAZOS R. 1989. Topology design and bandwith allocation in ATM nets. *IEEE Journal on Selected Areas in Communications*, **7**: 1253–1261.

[16] GLOVER F. 1965. A multiphase-dual algorithm for the zero-one integer programming problem. *Operations Research*, **13**: 879–919.

[17] GROPP W & MORÉ J. 1997. Optimization Environments and the NEOS Server. *Approximation Theory and Optimization*, M.D. Buhmann and A. Iserles, eds., pp. 167–182, Cambridge University Press.

[18] ISHFAQ R & SOX C. 2012. Design of intermodal logistics networks with hub delays. *European Journal of Operational Research*, **220**: 629–641.

[19] LUNA HPL & MAHEY P. 2000. Bounds for global optimization of capacity expansion and flow assignment problems. *Operations Research Letters*, **26**: 211–216.

[20] MAHEY P, BENCHAKROUN A & BOYER F. 2001. Capacity and flow assignment of data networks by generalized Benders decomposition. *J. of Global Optimization*, **20**: 173–193.

[21] MAHEY P, PHONG TQ & LUNA HPL. 2001. Separable Convexification and DC Techniques for Capacity and Flow Assignment Problems. *RAIRO Operations Research*, **35**: 269–281.

[22] MAHEY P & SOUZA MC. 2007. Local optimality conditions for multicommodity flow problems with separable piecewise convex costs. *Operations Research Letters*, **35**: 221–226.

[23] MORABITO R & DE SOUZA MC. 2010. Roteamento de multi-fluxos em redes de filas genericas. *Pesquisa Operacional*, **30**: 583–600.

[24] OUOROU A, MAHEY P & VIAL JP. 2000. A survey of algorithms for convex multicommodity flow problems. *Management Science*, **46**: 126–147.

[25] SOUZA MC, MAHEY P & GENDRON P. 2008. Cycle-based algorithms for multicommodity network flow problems with separable piecewise convex costs. *Networks*, **51**: 133–141.

[26] SINAN G. 2011. A conic quadratic formulation for a class of convex congestion functions in network flow problems. *European Journal of Operational Research*, **211**: 252–262.

[27] TAWARMALANI M & SAHINIDIS NV. 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, **103**: 225–249.