



**HAL**  
open science

## A Tabu Search approach to solve a network design problem with user-optimal flows

Antonio Mauttone, Martine Labbé, Rosa Figueiredo

► **To cite this version:**

Antonio Mauttone, Martine Labbé, Rosa Figueiredo. A Tabu Search approach to solve a network design problem with user-optimal flows. ALIO/EURO Workshop on Applied Combinatorial Optimization, 2008, Buenos Aires, Argentina. hal-01882382

**HAL Id: hal-01882382**

**<https://hal.science/hal-01882382>**

Submitted on 26 Sep 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Tabu Search approach to solve a network design problem with user-optimal flows

Antonio Mauttone<sup>a</sup>, Martine Labbé<sup>b</sup> and Rosa M.V. Figueiredo<sup>c</sup>

<sup>a</sup> *Departamento de Investigación Operativa, Universidad de la República, J. Herrera y Reissig 565  
Piso 5, Montevideo, Uruguay, mauttone@fing.edu.uy*

<sup>b</sup> *Département d'Informatique, Université Libre de Bruxelles, Boulevard du Triomphe, B-1050,  
Bruxelles, Belgique, mlabb@ulb.ac.be*

<sup>c</sup> *Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rua São Francisco  
Xavier 524, Maracanã, Rio de Janeiro, RJ, 20550-900, Brasil, rosa@ime.uerj.br*

**Keywords:** Network design, shortest path routing, discrete bilevel programming, tabu search.

**Abstract.** In this work, we formulate a network design problem with user-optimal flows as a mixed discrete bilevel linear programming problem. The upper level objective function minimizes the sum of fixed and variable costs associated with edges. The lower level problem defines a set of independent shortest path problems on the network resulting from the decisions taken in the upper level. Using the Bellman optimality conditions for the shortest path problems in the lower level, we obtain a one-level integer formulation for the network design problem. We implemented a tabu search metaheuristic based in a framework proposed in the literature to solve discrete bilevel problems. We propose a hybrid strategy that uses information coming from the linear relaxation of the one level formulation to select the initial solutions in the tabu search. Preliminary computational results are discussed as well as some directions to continue this research.

## 1 Introduction

The fixed charge network design problem (FCNDP) consist of selecting a subset of edges from a graph, in such a way that a number of given commodities can be transported from their origins to their destinations. The problem states the minimization of the sum of fixed costs (due to selecting the edges) and variable costs (depending on the flows of commodities on those edges). Both fixed and variable costs are linear and the edges are uncapacitated. The FCNDP problem is a large class of network design problems (Magnanti and Wong, 1984).

A variant of the FCNDP states that the commodities are routed along a shortest path on the resulting network. We call this problem FCNDP-SPR. This characteristic adds new constraints to the general problem.

Namely, consider a network  $G = (V, E)$  where  $V$  is the set of vertices and  $E$  is the set of admissible edges linking these vertices. A length, a fixed and a variable cost are associated with each edge in the graph. Consider also, a set  $K$  of commodities to be transported along the edges in  $E$ . Each commodity  $k \in K$  is associated with an origin-destination pair of vertices  $(o^k, d^k)$ . The FCNDP-SPR problem is to select a set of edges in  $E$  to be opened and to find, for each commodity in  $K$ , a *feasible flow* in the resulting network such that the sum of the fixed costs plus the variable flow costs is minimized. For a commodity  $k \in K$ , a flow is said to be *feasible* if it is sent along a shortest path from  $o^k$  to  $d^k$ , with respect to the edge lengths.

The FCNDP-SPR involves two distinct decision makers acting non-cooperatively and in a sequential way. At the upper level the *leader* is in charge of choosing the set of edges to be opened and he makes this decision trying to minimize the fixed and variable costs. As an answer, at the lower level the *followers* choose a set of shortest paths in the resulting network

along which the commodities will be sent. The effect of a decision maker on the other part is indirect: the followers' decision is affected by the network designed in the upper level whereas the leader's decision is affected by the variable costs imposed by the routes set in the lower level.

The inclusion of shortest path routing constraints in a mixed integer program is not straightforward. Difficulties arise both at the modeling as well as to design efficient solution methods. For the best of our knowledge, the FCNDP-SPR has only been addressed two times in the literature (Billheimer and Gray, 1973; Kara and Verter, 2004) and has been treated as part of bigger problems in some applications (Holmberg and Yuan, 2004).

The remainder of this text is structured as follows. In Section (2) we present a discrete bilevel formulation (Colson et al., 2005) and a one-level integer formulation to the FCNDP-SPR problem. In Section (3) a tabu search procedure for the FCNDP-SPR problem is briefly described. Preliminary computational results for the one-level formulation and the tabu search are provided in Section (4). Finally, in Section (5) some concluding remarks are given.

## 2 Discrete formulations

Let  $A^E = \{(i, j), (j, i) \mid \forall [i, j] \in E\}$  be the set of arcs obtained by bidirecting the edges in  $E$ . With each edge  $e \in E$  we associate a fixed opening cost  $f_e$  and with each arc  $a \in A^E$  we associate a positive length  $c_a$  and an operating variable cost  $g_a^k$ , for each commodity  $k \in K$ . Considering an arc  $a = (i, j) \in A^E$  we define  $e(a) = [i, j]$  and  $\bar{a} = (j, i)$ . Also, considering an edge  $e = [i, j] \in E$  we define  $a(e) = (i, j)$  and  $\bar{a}(e) = (j, i)$ . Notice that there is no capacities associated neither with edges in set  $E$  nor with arcs in set  $A^E$ .

Each commodity  $k \in K$  must be sent along one unique path carrying the total demand associated with its origin-destination pair. Then, for each commodity  $k \in K$  a demand vector  $b^k \in \{-1, 0, 1\}^{|V|}$  is specified as

$$b_i^k = \begin{cases} 1 & \text{if } i = o^k, \\ -1 & \text{if } i = d^k, \\ 0 & \text{otherwise.} \end{cases}$$

For each edge  $e \in E$ , we define an upper level variable  $y_e \in [0, 1]$  such that  $y_e = 1$  if and only if edge  $e$  is part of the network design. Also, for each commodity  $k \in K$  we define a set of continuous lower level variables  $x_a^k$ ,  $a \in A^E$ , used to model the shortest path from vertex  $o^k$  to vertex  $d^k$ , with respect to the lengths  $c_a$ . The discrete bilevel formulation (BIP) to the FCNDP-SPR problem is

$$\min_y \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{a \in A^E} g_a^k x_a^k \quad (1)$$

$$y_e \in \{0, 1\}, \forall e \in E, \quad (2)$$

where  $x$  is a solution to the problem

$$\min_x \sum_{k \in K} \sum_{a \in A^E} c_a x_a^k \quad (3)$$

$$\text{subject to } \sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^-(i)} x_a^k = b_i^k, \quad \forall i \in V, \forall k \in K, \quad (4)$$

$$x_{a(e)}^k + x_{\bar{a}(e)}^k \leq y_e, \quad \forall e \in E, \forall k \in K, \quad (5)$$

$$x_a^k \geq 0, \quad \forall a \in A^E, \forall k \in K. \quad (6)$$

Let  $\delta^+(i)$  (respectively  $\delta^-(i)$ ) denotes the set of arcs having node  $i$  as its tail (respectively head). The upper level objective function (1) minimizes the sum of the fixed and variable costs. The lower level problem is described by (3)–(6) in which constraints (4) are the well-known flow conservation constraints. Constraints (5) force each variable  $y_{e(a)}$  to take value 1 whenever  $x_a^k$  is positive for some  $k \in K$ .

The FCNDP-SPR problem can be reformulated as a one-level integer formulation by replacing the lower-level linear program defined by (3), (4), (5) and (6) with its optimality conditions. That could be done by applying the fundamental theorem of duality and the complementary slackness theorem (Bazaraa et al., 2004). However, optimality conditions for the lower level problem are, in fact, shortest path optimality conditions and they can be written in a more compact and effective way considering the Bellman optimality conditions for the shortest path problem (Ahuja et al., 1993) and using a simple lifting procedure (de Giovanni, 2004).

A one-level formulation (IP) for the FCNDP-SPR problem is as follows.

$$\min_{y,x} \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{a \in A^E} g_a^k x_a^k$$

subject to (2), (4), (5),

$$x_a^k \in \{0, 1\}, \quad \forall a \in A^E, \forall k \in K, \quad (7)$$

$$\pi_i^k - \pi_j^k \leq M^k - y_{e(a)}(M^k - c_a) - 2c_a x_a^k, \quad \forall a = (i, j) \in A^E, \forall k \in K, \quad (8)$$

$$\pi_{d^k}^k = 0, \quad \forall k \in K. \quad (9)$$

Variables  $\pi_i^k$ ,  $k \in K$ ,  $i \in V$ , are the shortest distances from vertice  $i$  to vertex  $d^k$ . Then, we can set variable  $\pi_{d^k}^k$  equal to zero. Assuming the integrality of variables  $y$  and  $x$  and assuming also that inequalities (5) are satisfied, constraints (8) are equivalent to Bellman optimality conditions for the set of  $|K|$  origin-destination pairs. The constant value  $M^k$ ,  $k \in K$ , has to be sufficiently large to guarantee that variables  $\pi_i^k$  and  $\pi_j^k$  will satisfy the shortest path conditions whenever  $y_e = 0$ ,  $e = [i, j] \in E$ .

### 3 A simple tabu search procedure

Tabu Search (TS) is a metaheuristic originally proposed by Glover (Glover, 1989) to allow local search methods to overcome local optima. Its basic principle is to accept non-improving moves in order to allow the local search to escape from local optima; cycling back to already visited solutions is prevented by using memories called tabu lists, that record the recent history of the search (Gendreau, 2003). TS has been successful in solving many hard combinatorial optimization problems, obtaining approximate solutions that are close to optimality (Glover and Laguna, 1997).

In this work we apply TS to obtain approximate solutions to the FCNDP-SPR problem. Our main interest at this stage of the work is to compare its performance against the bounds obtained using the (IP) formulation in a branch-and-bound framework, for problem instances of different sizes. We propose a TS based on the algorithm of Wen and Huang (Wen and Huang, 1996), that is originally proposed to solve mixed-integer linear bilevel problems with discrete variables in the upper level. The algorithm of Wen and Huang represents a solution by recording explicitly the value of each variable of the higher level of the mathematical formulation. Our algorithm, that we call as Simple Tabu Search (STS), uses the same solution representation; thus we are able to use information coming from the mathematical formulation to improve the performance of STS.

STS represents a solution  $S$  as a binary vector  $(s_1, \dots, s_{|E|})$ , where  $s_e = 1$  means that edge  $e$  belongs to  $S$  ( $s_e = 0$  if  $e$  is not included in  $S$ ). We use the neighborhood structure proposed by Wen and Huang, where the set  $N(S)$  of neighbor solutions of  $S$  is obtained by replacing  $s_e$  by  $1 - s_e$  for each  $e \in 1..|E|$ . This simple neighborhood structure changes the status of an edge in the solution, including the edge if it is not included or removing it, if it is included on the solution; according to this, the size of  $N(S)$  is  $|E|$  for any solution  $S$ . Notice that the search only changes values of the  $y$  variables of the formulation (those belonging to the higher level); the values of  $x$  variables must be calculated according to the lower level problem, which in our case is equivalent to calculate  $|K|$  independent shortest paths.

Algorithm 1 shows a pseudo-code of STS. At line 3, a new solution is constructed by following a given procedure; it must check that the solution allows to connect the origin and destination vertices of each commodity (constraint (4)). The neighborhood of  $S_i$  is explored at line 5. For each neighbor obtained by changing the status of a single edge, the feasibility of the resulting network  $G'$  is checked with respect to constraint (4). Then, shortest paths in  $G'$  are calculated for every commodity  $k$ . Since formulation (BIP) implicitly represents the optimistic version of the bilevel problem (Colson et al., 2005), we must check the existence of more than one shortest path for each commodity. If  $q_k$  is the number of shortest paths from  $o^k$  to  $d^k$  in  $G'$ , we evaluate the objective function of the higher level for  $q_1 \times \dots \times q_{|K|}$  different solutions of the  $x$  variables and we select the less costly according to objective function (1). At line 6, the tabu list is updated, by stating that the edge involved in the move performed at line 5, can not change its value during the following  $t$  local search iterations (where the value of  $t$  is set according to a given criterion). The tabu status of a move can be dropped (aspiration) if the move leads to a solution with less cost than a given value (aspiration level).

---

#### Algorithm 1 STS

---

```

1:  $S_{best} \leftarrow \emptyset$ ;
2: for  $i \in 1..startingSolutions$  do
3:    $S_i \leftarrow$  generate new solution;
4:   while not stopping criterion do
5:      $S_i \leftarrow$  select best neighbor of  $S_i$  not in the tabu list;
6:     Update tabu list;
7:     if  $S_i$  improves  $S_{best}$  then
8:        $S_{best} \leftarrow S_i$ ;
9:     end if
10:  end while
11: end for

```

---

## 4 Preliminary computational results

STS was tested using a set of random instances specially generated for this problem. The instances are characterized by three parameters, namely number of vertices  $n$ , number of commodities  $k$  and density of edges  $\alpha \in [0, 1]$ . The program that generates the instances takes these parameters, and randomly generates  $m$  (a fourth parameter) different instances with the characteristic indicated by  $(n, k, \alpha)$ . Parameter  $\alpha$  is used as a probability to decide whether or not to include in the instance, each possible edge of the complete undirected graph of  $n$  vertices. Fixed and variable costs, and edge lengths are set as random uniform numbers in the integer range  $[1..100]$ .

Instances were generated for values of  $n$  in the set  $\{10, 20, 30\}$ ; for each value of  $n$  we considered values of  $k$  in  $\{n/2, n, 3n/2\}$  and  $\alpha$  in  $\{0.3, 0.5, 0.8\}$ . For each one of these 27

combinations of parameters, we generated  $m = 5$  different instances.

We solved the formulation (IP) using the Branch and Bound algorithm of the Xpress Optimizer 1.7, turning off its pre-solve routine and its automatic cuts generation. We put a time limit of 1 hour in a 1 GHz processor computer.

The solver found the optimal solution for the small and medium-sized instances. For the smallest instances (10 vertices), it took less than 1 second. For some of the biggest instances (30 vertices, density 0.8, 30 and 45 commodities), the solver does not found the optimal solution within the imposed time-out.

For the TS we configured the algorithm as follows: the number of *startingSolutions* is equal to  $|E|$ ,  $t$  is randomly selected at each move from the integer interval  $1..|E|$ , the stopping criterion uses a time limit and a condition that is true when no improvement is found after  $|E|/2$  moves. The aspiration level is set as the objective value of the current solution of the search. The algorithm was coded in C++, and run in a 1.8 Ghz computer. The time limit was 30 minutes. We implemented three variants of the routine of generation of initial solutions:

1. Uniform random generation (STS1): for each edge  $e$ , a random uniform value in the real interval  $[0, 1]$  is sampled, and it is used to decide whether or not to include the edge on the solution.
2. Biased random generation (STS2): similar to the previous one, but using probabilities which are proportional to the value of the variable  $y_e$ , in the optimal solution of the linear relaxation of (IP) formulation.
3. Profit-guided generation (STS3): the edges are sorted in decreasing order according to the values of their respective variables in the optimal solution of the linear relaxation of (IP) formulation. At each new solution generated, a new edge is added to it taken from the sorted list described above.

Table 1 shows the results produced by both exact (Branch and Bound) and approximated (STS) approaches, for the 5 instances of characteristic (30,30,0.8). In that table,  $z^{best}$  and  $z^*$  are the objective value of the best solution found by the exact and approximated approach respectively,  $t$  is the CPU time in seconds and  $gap = (z^{best} - z^*)/z^{best} * 100$  is the percentage gap. We observe that gap values of STS1 are very high ( $\dagger$ ), while gaps of STS2 and STS3 are much lower ( $\ddagger$ ). This suggest that the information provided by the optimal solution of the linear relaxation of formulation (IP) greatly improves the performance of the STS algorithm; on the other hand the “blind” generation of solutions of STS1 performs poorly when compared with the exact approach. Moreover we observe that the exact approach does not found the optimal solution for 4 out of the 5 instances ( $-$ ), while STS2 and STS3 found better solutions for 3 of these instances ( $\checkmark$ ).

instance	Exact		STS1			STS2			STS3		
	$z^{opt}$	$t$	$z^*$	gap	$t$	$z^*$	gap	$t$	$z^*$	gap	$t$
1	4830	61,39	52119	979,07 $\dagger$	1802	5026	4,06 $\ddagger$	230	4927	2,01 $\ddagger$	1110
2	7276	-	50788	598,02 $\dagger$	1810	7426	2,06 $\ddagger$	1262	7322	0,63 $\ddagger$	93
3	8749	-	58623	570,05 $\dagger$	1826	8321	$\checkmark$	98	8142	$\checkmark$	565
4	9741	-	64783	565,05 $\dagger$	1822	8864	$\checkmark$	1059	8828	$\checkmark$	1287
5	7560	-	83644	1006,40 $\dagger$	1870	7573	0,17 $\ddagger$	1212	7502	$\checkmark$	794

Table 1: Results for instances (30,30,0.8)

## 5 Conclusions and future work

A variant of the fixed charge network design problem is studied, where the commodities are routed along the shortest path of the network. We model it as a linear mixed-integer bilevel problem with discrete variables in the upper level, and we solve it using a tabu search algorithm. We compare the results of the tabu search against exact solutions obtained by branch and bound using a one-level integer linear formulation derived from the original bilevel formulation. Tabu search obtains good results, specially for the biggest instances of the problem; information coming from the linear relaxation of the one-level integer formulation greatly contributes to the performance of the tabu search.

Some aspects of the implementation of the tabu search could be changed in order to improve its performance. The computation of shortest paths during the exploration of the neighborhood is a critical component of the overall algorithm. Avoiding recomputation of all these paths (Buriol et al., 2008) could improve the performance of the algorithm.

**Acknowledgments:** This work was supported by grant from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-CAPES and the Project ALFA II-0457-FA-FCD-FI-FC.

## References

- Ahuja, R. K., Magnanti, T. L. and Orlin, J. B.: 1993, *Network flows*, Prentice-Hall, New Jersey.
- Bazaraa, M. S., Jarvis, J. and Sherali, H. D.: 2004, *Linear Programming and Network Flows*, 3rd edn, Wiley-Interscience.
- Billheimer, J. W. and Gray, P.: 1973, Network design with fixed and variable cost elements, *Transportation Science* **7**, 49–74.
- Buriol, L., Resende, M. and Thorup, M.: 2008, Speeding up dynamic shortest-path algorithms, *INFORMS Journal on Computing* **20**(2), 191–204.
- Colson, B., Marcotte, P. and Savard, G.: 2005, Bilevel programming: A survey, *4OR: A Quarterly Journal of Operations Research* **3**(2), 87–107.
- de Giovanni, L.: 2004, *The internet protocol network design problem with reliability and routing constraints*, PhD thesis, Politecnico di Torino.
- Gendreau, M.: 2003, An introduction to tabu search, in F. Glover and G. Kochenberger (eds), *Handbook of metaheuristics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 37–54.
- Glover, F.: 1989, Tabu search part I, *ORSA Journal on Computing* **1**, 190–206.
- Glover, F. and Laguna, M.: 1997, *Tabu search*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Holmberg, K. and Yuan, D.: 2004, Optimization of internet protocol network design and routing, *Networks* **43**(1), 39–53.
- Kara, B. Y. and Verter, V.: 2004, Designing a road network for hazardous materials transportation, *Transportation Science* **38**(2), 188–196.
- Magnanti, T. L. and Wong, R. T.: 1984, Network design and transportation planning: Models and algorithms, *Transportation Science* **18**(1), 1–55.
- Wen, U. P. and Huang, A. D.: 1996, A simple tabu search method to solve the mixed-integer linear bilevel programming problem, *European Journal of Operational Research* **88**, 563–571.