

Comparison of formulations for the two-level uncapacitated facility location problem with single assignment constraints



Bernard Gendron^a, Paul-Virak Khuong^a, Frédéric Semet^{b,*}

^a CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation) Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville Montreal, Quebec H3C 3J7, Canada

^b Univ. Lille, CNRS, Centrale Lille, Inria UMR 9189 - CRISTAL Centre de Recherche en Informatique Signal et Automatique de Lille F-59000 Lille, France

ARTICLE INFO

Article history:

Received 27 May 2016

Revised 5 December 2016

Accepted 17 February 2017

Available online 30 March 2017

Keywords:

Two-level uncapacitated facility location

Mixed-integer programming

Relaxations

Formulations

ABSTRACT

We consider the two-level uncapacitated facility location problem with single assignment constraints (TUFLP-S), an extension of the uncapacitated facility location problem. We present six mixed-integer programming models for the TUFLP-S based on reformulation techniques and on the relaxation of the integrality of some of the variables associated with location decisions. We compare the models by carrying out extensive computational experiments on large, hard, artificial instances, as well as on instances derived from an industrial application in freight transportation.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In this paper, we study and compare several formulations for the two-level uncapacitated facility location problem with single assignment constraints (TUFLP-S), an extension of the uncapacitated facility location problem (UFLP) (Krarup and Pruzan, 1983). The UFLP consists in selecting a set of depots from potential locations in order to minimize an objective function that includes fixed costs associated with each depot and transportation costs from any depot to each customer. In the two-level uncapacitated facility location problem (TUFLP), the single set of locations is substituted with two tiers of locations (depots and satellites), and the path to each customer must begin at a depot and transit by a satellite. The objective function includes fixed costs associated with the depots and the satellites, fixed costs for establishing connections between depots and satellites, and transportation costs from any depot to each customer, i.e., each path of the form depot-satellite-customer has a corresponding transportation cost. The TUFLP-S imposes the additional restriction that each satellite can be connected to at most one depot. These single assignment constraints appear in a number of applications, most notably in transportation (Tragantalerngsak et al., 1997) and telecommunications (Chardaire et al., 1999). Note also that, for a large class of

TUFLP instances for which the single assignment constraints are not explicitly enforced, there is an optimal solution that satisfies these constraints, due to the structure of the objective function (Chardaire et al., 1999).

There is a wide body of literature devoted to multi-level facility location problems (Klose and Drexler, 2005; Melo et al., 2006; Sahin and Süral, 2007; Zanjirani Farahani et al., 2014), which generalize the TUFLP-S. Most papers are dedicated to two-level problems, but some address the general multi-level case, e.g., Kratica et al. (2014) and Ortiz-Astorquiza et al. (2015). There are two types of mathematical programming models: arc-based (Kratica et al., 2014; Marín, 2006; Pirkul and Jayaraman, 1996; 1998) and path-based (Barros and Labbé, 1994; Gao and Robinson, 1992; Kaufman et al., 1977; Ro and Tcha, 1984). The comparison between relaxations of these different models is performed both from theoretical and experimental perspectives (Bloemhof-Ruwaard et al., 1994; 1996; Chardaire et al., 1999; Marín and Pelegrin, 1999). For two-level capacitated/uncapacitated facility location problems, exact methods are mainly based on polyhedral approaches or on Lagrangian relaxation. Some authors (Aardal et al., 1996; Baiou and Barahona, 2014; Chardaire et al., 1999; Landete and Marín, 2009) have shown how to strengthen the models by adding valid inequalities and facets. Lagrangian relaxations of the models are also considered in several papers (Barros, 1995; Gendron et al., 2016; Marín and Pelegrin, 1999; Pirkul and Jayaraman, 1996; 1998), exploiting well-known structures in Lagrangian sub-problems, such as UFLPs (Gendron et al., 2016) or knapsack problems (Marín and Pelegrin, 1999; Pirkul and Jayaraman, 1996; 1998).

* Corresponding author.

E-mail addresses: Bernard.Gendron@cirrelet.ca (B. Gendron), pvk@pvk.ca (P.-V. Khuong), frederic.semet@centralelille.fr (F. Semet).

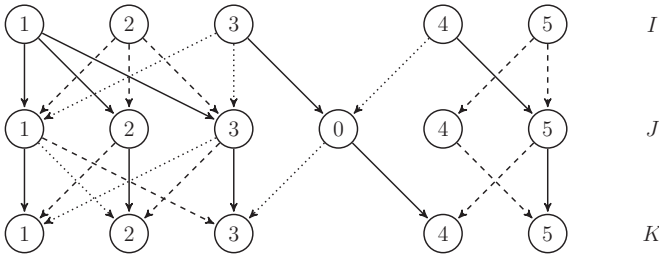


Fig. 1. Example of TUFLP instance.

When single assignment constraints are imposed, the literature becomes scarce. To the best of our knowledge, single assignment constraints were first introduced in Tragantalerngsak et al. (1997) for a two-level capacitated facility location problem. The authors describe a Lagrangian relaxation-based branch-and-bound algorithm for solving the problem. We could identify only three references addressing the TUFLP-S: Chardaire et al. (1999) proposes a strengthened formulation, as well as a heuristic method combining Lagrangian relaxation and simulated annealing; Gendron et al. (2015) presents a multi-layer variable neighbourhood search heuristic; Gendron et al. (2016) develops a Lagrangian relaxation method. These three papers focus on the development of efficient solution methods, while the present paper focuses on the modelling of the problem.

The contribution of this paper is two-fold. First, we introduce six mixed-integer programming (MIP) models for the TUFLP-S based on reformulation techniques and on the relaxation of the integrality of some of the variables associated with location decisions. One of these formulations was previously considered in Gendron et al. (2016) to derive a Lagrangian relaxation for the TUFLP-S. Second, we compare the models by solving a large number of various instances with a state-of-the-art MIP solver. The results show that, whenever fixed costs at the depots (at the satellites) are significant, it is beneficial to keep the integrality of the corresponding binary variables, but to relax the integrality of the binary variables associated with the satellites (with the depots). In our experiments, poor results are obtained by the reformulation that minimizes the number of binary variables by relaxing the integrality of the two types of location variables.

The paper is organized as follows. In Section 2, we present a general formulation for the TUFLP (Barros and Labbé, 1994) and we adapt this model to derive an initial MIP formulation for the TUFLP-S. We then propose five additional MIP formulations and theoretically compare the LP relaxations of these models. The formulations are then compared experimentally in Section 3. Last, some conclusions are drawn in Section 4.

2. Formulations for the TUFLP-S

To define the TUFLP, we introduce the following notation: I is the set of potential depot locations, J is the set of potential satellite locations, and K is the set of customers. Fixed costs and transportation costs are defined as follows: f_i , g_j and h_{ij} are the nonnegative fixed costs for, respectively, each depot $i \in I$, each satellite $j \in J$ and each pair of depot-satellite $(i, j) \in I \times J$; c_{ijk} is the transportation cost on each path $(i, j, k) \in I \times J \times K$ from a depot i to a satellite j to a customer k .

Example 1. Fig. 1 shows a small instance of the TUFLP with $|I| = |K| = 5$ and $|J| = 6$. Each depot $i \in I$ has a fixed cost $f_i = 1$ and each satellite $j \in J$ has a fixed cost $g_j = 1$, except satellite 0, which has a fixed cost $g_0 = \epsilon$, with $0 < \epsilon < 1$. We assume that the fixed cost for any pair $(i, j) \in I \times J$ is $h_{ij} = 0$. The instance is constructed in such a way that any arc $(j, k) \in J \times K$ displayed in

Fig. 1 belongs to a single path $(i, j, k) \in I \times J \times K$. Hence, we use plain, dotted and dashed lines to easily identify the paths in $I \times J \times K$ that are shown in Fig. 1. For instance, path $(1, 1, 1)$ is shown (with a plain line) in Fig. 1, but not path $(1, 1, 2)$, since the arc $(1, 2)$ between satellite 1 and customer 2 belongs to path $(3, 1, 2)$ shown (with a dotted line) in Fig. 1. Any path $(i, j, k) \in I \times J \times K$ displayed in Fig. 1 has a cost $c_{ijk} = 0$, while any other path in $I \times J \times K$ is assumed to have an arbitrarily large cost. This TUFLP instance is used throughout the text to illustrate the main results.

Barros and Labbé (1994) propose to solve the TUFLP with a MIP formulation that uses the following sets of binary variables:

$$y_i = \begin{cases} 1, & \text{if depot } i \text{ is open,} \\ 0, & \text{otherwise,} \end{cases} \quad \forall i \in I,$$

$$z_j = \begin{cases} 1, & \text{if satellite } j \text{ is open,} \\ 0, & \text{otherwise,} \end{cases} \quad \forall j \in J,$$

$$t_{ij} = \begin{cases} 1, & \text{if depot } i \text{ is connected to satellite } j, \\ 0, & \text{otherwise,} \end{cases} \quad \forall (i, j) \in I \times J,$$

$$x_{ijk} = \begin{cases} 1, & \text{if customer } k \text{ is served through pair } (i, j), \\ 0, & \text{otherwise,} \end{cases} \quad \forall (i, j, k) \in I \times J \times K.$$

The MIP model, which we denote (G) , is then written as follows:

$$\min_{y,z,t,x} \sum_{i \in I} f_i y_i + \sum_{j \in J} g_j z_j + \sum_{(i,j) \in I \times J} h_{ij} t_{ij} + \sum_{(i,j,k) \in I \times J \times K} c_{ijk} x_{ijk}, \quad (1)$$

subject to

$$\sum_{(i,j) \in I \times J} x_{ijk} = 1, \quad \forall k \in K, \quad (2)$$

$$x_{ijk} \leq t_{ij}, \quad \forall (i, j, k) \in I \times J \times K, \quad (3)$$

$$\sum_{j \in J} x_{ijk} \leq y_i, \quad \forall (i, k) \in I \times K, \quad (4)$$

$$\sum_{i \in I} x_{ijk} \leq z_j, \quad \forall (j, k) \in J \times K, \quad (5)$$

$$0 \leq x_{ijk} \leq 1, \quad \forall (i, j, k) \in I \times J \times K, \quad (6)$$

$$y_i \in \{0, 1\}, \quad \forall i \in I, \quad (7)$$

$$z_j \in \{0, 1\}, \quad \forall j \in J, \quad (8)$$

$$t_{ij} \in \{0, 1\}, \quad \forall (i, j) \in I \times J. \quad (9)$$

Constraints (2) guarantee the satisfaction of the demand for each customer. Constraints (3) to (5) ensure that fixed costs are incurred for the use of, respectively, depot-satellite pairs, depots and satellites. Since there are no capacity constraints, there always exists an optimal solution to (G) where the demand for a single customer is not split across multiple paths, and the integrality requirements on variables x_{ijk} can be relaxed.

In most papers devoted to two-level uncapacitated facility location problems, a particular case is considered by setting the costs on the arcs between depots and satellites to zero ($h_{ij} = 0$ for any $(i, j) \in I \times J$). For this special case, solution approaches (Aardal et al., 1996; Barros, 1998; Landete and Marín, 2009) are

based on the MIP formulation that eliminates variables t_{ij} and constraints (3) from (G). This does not affect the LP relaxation bounds, since no cost is incurred when variables t_{ij} are set to one.

We now address a variant of the TUFLP that forces each satellite to be connected to at most one depot. For some instances, solutions to (G) will naturally satisfy these single assignment constraints. Here, we explicitly consider these constraints to derive MIP formulations that model the TUFLP-S and apply to instances where the single assignment requirements are either satisfied implicitly or need to be enforced.

Formulation (G) includes variables t_{ij} to determine whether arc $(i, j) \in I \times J$ is selected or not. We define the following constraints to enforce each satellite to be connected to at most one depot:

$$\sum_{i \in I} t_{ij} \leq 1, \quad \forall j \in J. \quad (10)$$

By adding these constraints to (G) we obtain the *weak formulation* (W) for the TUFLP-S, which is then defined by the objective (1) subject to constraints (2) to (10).

Example 1 (continued). We use the example of Fig. 1 to show that an optimal solution to the TUFLP might not satisfy the single assignment constraints (10), i.e., there are instances such that $v(G) < v(W)$. Indeed, an optimal solution to (G) consolidates the demands of customers 3 and 4 at satellite 0, taking advantage of the small fixed cost $g_0 = \epsilon$. In such a solution, only depots 3 and 4 need to be opened, along with satellites 0, 1, 3 and 5, allowing to satisfy all customer demands through paths (3, 3, 1), (3, 1, 2), (4, 0, 3), (3, 0, 4) and (4, 5, 5), thus giving an optimal value $v(G) = 5 + \epsilon$. Clearly, this solution violates constraints (10), since two ingoing arcs are used at satellite 0, i.e., $t_{40} + t_{30} = 2 > 1$. When constraints (10) are added, we can derive several optimal solutions of value $v(W) = 7$, for instance by opening depots 1 and 5, along with all satellites, except 0, and by using paths (1, 1, 1), (1, 2, 2), (1, 3, 3), (5, 5, 4) and (5, 4, 5). Note that any solution that opens satellite 0 would incur a cost of $v(W) = 7 + \epsilon$ and would not be optimal. Thus, for this instance, we have $v(G) = 5 + \epsilon < 7 = v(W)$.

The literature on the TUFLP often considers the special case where the transportation costs on the paths are separable by arc, i.e., $c_{ijk} = d_k a_{ij} + b_{jk}$ for any $(i, j, k) \in I \times J \times K$. For TUFLP instances that satisfy this property, in addition to $h_{ij} = 0$ for any $(i, j) \in I \times J$, it is easy to show that (G) and (W) are equivalent (Gendron et al., 2016). We can go one step further and show that the LP relaxations of the two models give the same bound. For any model F , we denote by $v(F)$ and \bar{F} its optimal value and its LP relaxation, respectively.

Proposition 1. For any TUFLP instance such that $h_{ij} = 0$, $\forall (i, j) \in I \times J$ and $c_{ijk} = d_k a_{ij} + b_{jk}$, $\forall (i, j, k) \in I \times J \times K$, we have $v(\bar{G}) = v(\bar{W}) \leq v(W) = v(G)$.

Proof. Since (\bar{G}) is a relaxation of (\bar{W}) , we have $v(\bar{G}) \leq v(\bar{W})$.

To show that $v(\bar{G}) \geq v(\bar{W})$, consider an optimal solution to (\bar{G}) that violates constraints (10), i.e., $\sum_{i \in I} t_{ij} > 1$ for some $j' \in J$. This implies that there is at least one pair of arcs (i', j') and (i'', j') such that $t_{i'j'} > 0$ and $t_{i''j'} > 0$. We assume without loss of generality that $a_{i'j'} \leq a_{i''j'}$. Because $t_{i''j'} > 0$, there exists $L \subseteq K$, $L \neq \emptyset$ such that $x_{i''j'l} > 0$ for $l \in L$ and $x_{i''j'k} = 0$ for $k \in K \setminus L$. If we move the total flow $d_l x_{i''j'l}$ on path (i'', j', l) to path (i', j', l) for all $l \in L$, we obtain another feasible solution where we can set $t_{i''j'} = 0$. The cost of this solution is necessarily the same as that of the original optimal solution, i.e., we have constructed another optimal solution. By repeating this argument a finite number of times, we eventually end up with an optimal solution to (\bar{G}) that satisfies constraints (10). Hence, $v(\bar{G}) \geq v(\bar{W})$.

The proof of the equation $v(W) = v(G)$ follows the same argument. \square

Example 1 (continued). It is clear that the instance of Fig. 1 does not satisfy the conditions of Proposition 1, since, for instance, $c_{111} = d_1 a_{11} + b_{11} = 0$ and $c_{211} = d_1 a_{21} + b_{11}$ is arbitrarily large imply a_{21} is arbitrarily large, but this contradicts $c_{213} = d_3 a_{21} + b_{13} = 0$. For this instance, we have seen above that $v(G) = 5 + \epsilon < 7 = v(W)$. It is interesting to note, however, that $v(\bar{G}) = v(\bar{W})$ for this instance. Indeed, an optimal solution to the two models splits the flow equally on all 0-cost paths going to each customer, i.e., $x_{ijk} = 1/3$, for each $(i, j, k) \in I \times J \times K$ such that $c_{ijk} = 0$, $k = 1, 2, 3$, and $x_{ijk} = 1/2$, for each $(i, j, k) \in I \times J \times K$ such that $c_{ijk} = 0$, $k = 4, 5$. Correspondingly, we have $t_{ij} = 1/3$, when $j = 1, 2, 3$, and $t_{ij} = 1/2$, when $j = 0, 4, 5$, for each $(i, j) \in I \times J$ shown in Fig. 1. As a result, the location variables assume the following values: $y_i = 1/3$, $i = 1, 2$, $y_i = 1/2$, $i = 3, 4, 5$, $z_j = 1/3$, $j = 1, 2, 3$ and $z_j = 1/2$, $j = 0, 4, 5$, which yields an optimal value $v(\bar{G}) = v(\bar{W}) = 4 + 1/6 + \epsilon/2$. Thus, for the TUFLP solved with model (G), the integrality gap is at least 20% (if $\epsilon \rightarrow 0$) and at most 28.6% (if $\epsilon \rightarrow 1$), while for the TUFLP-S solved with model W, the integrality gap varies between 50% (if $\epsilon \rightarrow 1$) and 68% (if $\epsilon \rightarrow 0$).

To improve model (W), we propose a reformulation based on a simple property of feasible solutions. For a given satellite $j \in J$, at most one variable t_{ij} can be equal to 1 due to constraints (10). Moreover, z_j is equal to 1 if and only if $t_{i'j}$ is equal to 1 for some i' , since fixed costs g_j are nonnegative. In other words, either $z_j = t_{ij} = 0$, for any $i \in I$, or there exists a single $i' \in I$ such that $t_{i'j} = z_j = 1$ and $t_{ij} = 0$, for any $i \in I$, $i \neq i'$. On the basis of these observations, we can add to (W) the following valid inequalities:

$$\sum_{i \in I} t_{ij} = z_j, \quad \forall j \in J. \quad (11)$$

We can then remove constraints (10), which become redundant, as well as constraints (5), which are implied by (3) and (11). We thus obtain the *strong formulation* (S), defined by the objective (1) subject to constraints (2) to (4), (6) to (9) and (11).

Proposition 2. $v(\bar{W}) \leq v(\bar{S})$ and the inequality can be strict.

Proof. When considering the LP relaxations of (W) and (S), it is easy to see that (3) and (11) imply (5) and (10), given that $z_j \leq 1$, $j \in J$. Hence, (\bar{W}) is a relaxation of (\bar{S}) and $v(\bar{W}) \leq v(\bar{S})$. Example 1 (see below) provides an instance for which the inequality can be strict. \square

Example 1 (continued). The optimal solution to \bar{W} given above does not satisfy constraints (11), since, for any $j \in J$, except 4, we have $\sum_{i \in I} t_{ij} > z_j$. An optimal solution to model \bar{S} does not use either depot 1 or depot 2 (either $y_1 = 0$ or $y_2 = 0$) and splits the flow equally on all remaining 0-cost paths going to each customer. For instance, if $y_1 = 0$, then an optimal solution satisfies $x_{ijk} = 1/2$, for each $(i, j, k) \in I \times J \times K$ such that $c_{ijk} = 0$, $i \neq 0$, with the remaining variables assuming the following values: $y_i = 1/2$, $i = 2, 3, 4, 5$, $z_j = 1/2$, $j = 2, 4$, $z_j = 1$, $j = 0, 1, 3, 5$ and $t_{ij} = 1/2$ for each $(i, j) \in (I \setminus \{1\}) \times J$ shown in Fig. 1. The optimal value is then $v(\bar{S}) = 6 + \epsilon$, corresponding to an integrality gap that varies between 0% (if $\epsilon \rightarrow 1$) and 16.7% (if $\epsilon \rightarrow 0$), which is significantly better than the LP relaxation bound obtained by model \bar{W} , i.e., $v(\bar{W}) = 4 + 1/6 + \epsilon/2 < 6 + \epsilon = v(\bar{S})$.

With the objective of reducing the number of binary variables, a reformulation of (S) can be obtained by projecting out variables z_j using equalities (11) and by reintroducing constraints (10). This yields a simpler formulation, denoted (S_p) :

$$\min_{y,t,x} \sum_{i \in I} f_i y_i + \sum_{(i,j) \in I \times J} (g_j + h_{ij}) t_{ij} + \sum_{(i,j,k) \in I \times J \times K} c_{ijk} x_{ijk} \quad (12)$$

subject to (2) to (4), (6) to (10). As can be seen from the objective (12), constraints (11) imply that we can attribute the fixed costs g_j to every variable t_{ij} .

In any of the three TUFLP-S formulations introduced so far, (W), (S) and (S_p), the integrality requirements on variables y_i can be relaxed after introducing the following valid inequalities:

$$t_{ij} \leq y_i, \quad \forall (i, j) \in I \times J. \tag{13}$$

Only non-negativity constraints on variables y_i have then to be imposed. Adding constraints (13) yields a corresponding model with more constraints, but fewer binary variables than the initial model. Thus, we can derive models (W^C), (S^C) and (S_p^C) from formulations (W), (S) and (S_p), respectively. Each of these models is obtained from the corresponding one by adding constraints (13) and by relaxing the integrality of variables y_i . Note that one of these models, (S_p^C), has been used in Gendron et al. (2016) to derive a Lagrangian relaxation method.

To summarize, we have introduced six formulations for the TUFLP-S for which the LP relaxations can be compared with the following result:

Proposition 3. $\nu(\bar{W}) = \nu(\bar{W}^C) \leq \nu(\bar{S}) = \nu(\bar{S}_p) = \nu(\bar{S}^C) = \nu(\bar{S}_p^C)$.

Proof. To show that, for any formulation (\bar{F}), we have $\nu(\bar{F}) = \nu(\bar{F}^C)$, we have to show that constraints (13) are redundant for (\bar{F}). The argument is based on constraints (3), (4) and the non-negativity of h_{ij} . Indeed, for any $(i, j) \in I \times J$, $h_{ij} \geq 0$ implies there is an optimal solution to (\bar{F}) such that $t_{ij} = \max_{k \in K} \{x_{ijk}\} \equiv x_{ijk}^*$, by constraints (3). Thus, $t_{ij} = x_{ijk}^* \leq \sum_{j' \in J} x_{ij'k^*} \leq y_i$, by constraints (4), and constraints (13) are redundant for (\bar{F}).

Finally, the equality $\nu(\bar{S}) = \nu(\bar{S}_p)$ is trivial and the inequality $\nu(\bar{W}) \leq \nu(\bar{S})$ was shown in Proposition 2. \square

The four formulations (S), (S_p), (S^C) and (S_p^C) all provide the same LP relaxation bound. They differ in the binary variables that remain: (S) preserves all the decision variables from (W), (S_p) eliminates variables z_j , while (S^C) and (S_p^C) both relax the integrality of variables y_i .

The performance of a state-of-the-art MIP software tool when solving these models may thus vary: is it better to mimic the initial formulation (W) with (S), to minimize the number of binary variables with (S_p^C), or to strike a compromise between these two extremes with (S_p) or (S^C)? In addition, it is not clear how a state-of-the-art MIP software tool would handle branching with each of these equivalent formulations. In particular, is it better to prioritize branching on variables y_i or on variables z_j , or to impose no branching priority on the different types of binary variables? Our computational experiments, to be presented next, address these issues, but we first look at the effect of different branching priorities on the instance of Fig. 1.

Example 1 (continued). As shown above, an optimal solution to \bar{S} is given by $y_1 = 0$, $y_i = 1/2$, $i = 2, 3, 4, 5$, $z_j = 1/2$, $j = 2, 4$, $z_j = 1$, $j = 0, 1, 3, 5$, $t_{ij} = 1/2$ for each $(i, j) \in (I \setminus \{1\}) \times J$ shown in Fig. 1 and $x_{ijk} = 1/2$, for each $(i, j, k) \in I \times J \times K$ such that $c_{ijk} = 0$, $i \neq 0$. If we give priority to branching on variables z_j , we would add constraints $z_j = 0$ and $z_j = 1$ for $j = 2$ or $j = 4$. Whatever the variable chosen, z_2 or z_4 , the two branches quickly yield an optimal solution. Indeed, the branch where $z_j = 0$ gives an LP relaxation integer solution of value $7 + \epsilon$, while the branch where $z_j = 1$ provides an LP relaxation integer solution of value 7, and this true whether $j = 2$ or $j = 4$. Thus, when variables z_j are given priority, only two nodes are needed to prove optimality, irrespective of the choice of the branching variable. By contrast, if priority is given to variables y_i , three cases might happen:

1) y_5 is selected, in which case the branch $y_5 = 0$ gives an LP relaxation integer solution of value $7 + \epsilon$ and the branch $y_5 = 1$

Table 1
Characteristics of the four industrial TUFLP-S instance sets.

Instances	I	J	K	A	P
Tiny	23	80	175	134	3366
Small	47	160	351	592	28,496
Medium	70	240	526	1236	92,554
Full	93	320	701	2250	222,308

provides an LP relaxation integer solution of value 7, which implies that only two nodes are needed to prove optimality;

2) y_3 or y_4 is selected, in which case the branch $y_i = 0$ produces an LP relaxation integer solution of value 7, but the branch $y_i = 1$ gives an LP relaxation fractional solution of value $6 + 1/2 + \epsilon$ (the optimal solution to \bar{S} with the branching variable equal to 1, instead of 1/2), which implies that, if $\epsilon < 1/2$, more than two nodes are needed before proving optimality;

3) y_2 is selected, in which case the branch $y_2 = 1$ provides an LP relaxation integer solution of value 7, but the branch $y_2 = 0$ gives an LP relaxation fractional solution of value $6 + \epsilon$ (the optimal solution to \bar{S} where $y_2 = 0$ and all other variables y_i assume value 1/2), which implies that, irrespective of the value of ϵ , more than two nodes are needed before proving optimality.

3. Computational results

To compare the computational efficiency of the six formulations, we conducted computational experiments using CPLEX 12.6.1 on a 2.5 GHz Intel Xeon E5-2609 with 128 GB RAM. The testbed includes 430 instances of two types. Instances of the first type are derived from industrial data. They result from subproblems obtained when solving a Lagrangian decomposition for a more complex distribution network design problem (Gendron and Semet, 2009). Their integrality gaps at the root are small, but they are of large size. Instances of the second type are artificial large-size TUFLP-S instances with single assignment generated in Gendron et al. (2016). They present large integrality gaps at the root node. These difficult instances allow us to study the branching strategies for the different formulations.

More specifically, we report two types of computational results. First, we give gaps and CPU times when solving the LP relaxation and when invoking CPLEX at the root node only. Then, we consider different branching priorities by branching first on either variables y_i or z_j to determine which branching scheme is more relevant for each type of instance and for each model. We report the total number of nodes, the optimality gap and the CPU time obtained after proving optimality or attaining the CPU limit of 10 hours.

3.1. Testbed

The testbed includes two types of instances. Instances of the first type (“I”) are divided into four sets of 100 instances derived from a location-distribution problem faced by a retail company (Gendron and Semet, 2009). Each set corresponds to an industrial instance from which 100 TUFLP-S instances inherit their network structure. We list their characteristics in Table 1, where columns 2 to 4 indicate the number of depots, satellites and customers, respectively, and columns 5 and 6 show the number of depot-satellite arcs in set $A \subseteq I \times J$, and the number of paths in set $P \subseteq I \times J \times K$, respectively. Instances in set “Tiny” are roughly one fourth the size of instances in set “Full”, those in “Small” half the size, and those in “Medium” three quarters. These TUFLP-S instances are defined on realistic graphs, but present a cost structure that makes them particularly difficult: depots incur the same large fixed cost, while satellites and depot-satellite arcs incur none. Moreover,

Table 2
Gaps and runtimes when computing LP relaxations.

Type	Instances		(W)	(W ^C)	(S)	(S _P)	(S ^C)	(S _P ^C)
I	Tiny	Gap (%)	0.01	0.01	0.01	0.01	0.01	0.01
		Time (s)	0	0	0	0	0	0
	Small	Gap (%)	0.43	0.43	0.43	0.43	0.43	0.43
		Time (s)	0	1	0	0	1	0
	Medium	Gap (%)	0.15	0.15	0.15	0.15	0.15	0.15
		Time (s)	4	4	4	4	6	4
	Full	Gap (%)	0.22	0.22	0.22	0.22	0.22	0.22
		Time (s)	28	24	23	25	37	21
L	LargeA	Gap (%)	20.42	20.42	19.14	19.14	19.14	19.14
		Time (s)	1	1	1	1	1	1
	LargeB	Gap (%)	27.43	27.43	24.29	24.29	24.29	24.29
		Time (s)	1	1	2	1	2	1
	LargeC	Gap (%)	24.52	24.52	21.70	21.70	21.70	21.70
		Time (s)	1	1	3	2	2	2

Table 3
Gaps and runtimes at the root node.

Type	Instances		(W)	(W ^C)	(S)	(S _P)	(S ^C)	(S _P ^C)
I	Tiny	Gap (%)	0.00	0.00	0.00	0.00	0.00	0.01
		Time (s)	0	0	0	0	0	0
	Small	Gap (%)	0.42	0.41	0.41	0.42	0.40	0.41
		Time (s)	2	2	2	2	3	2
	Medium	Gap (%)	0.15	0.15	0.15	0.15	0.15	0.15
		Time (s)	13	10	13	10	15	8
	Full	Gap (%)	0.22	0.22	0.22	0.22	0.22	0.22
		Time (s)	51	43	51	43	60	36
L	LargeA	Gap (%)	20.42	20.33	18.80	18.95	18.78	18.97
		Time (s)	2	2	2	2	2	2
	LargeB	Gap (%)	27.43	27.36	24.13	24.20	24.13	24.18
		Time (s)	2	3	3	2	3	2
	LargeC	Gap (%)	24.52	24.47	21.34	21.54	21.32	21.55
		Time (s)	3	3	4	3	4	3

transportation costs vary greatly depending on the locations on the path. Consequently, the single assignment constraints are not redundant and the general model (G) is not valid for these instances.

The second type (“L”) of instances were introduced in Gendron et al. (2016). A set of 30 two-level instances were obtained based on the generator proposed in Kochetov and Ivanenko (2005) to obtain UFLP instances with large integrality gaps. Such UFLP instances, including 150 facilities and 150 customers, are divided into three classes, A, B and C. For instances of class A, all depots are connected to an equal number of customers, while for instances of class B, all customers can be served by an equal number of depots. Class C instances are such that both customers and depots are connected to the same number of the other type of nodes. We used the procedure suggested in Landete and Marín (2009) to transform these UFLP instances into TUFLP instances. For each UFLP instance, the procedure consists in dividing the original arcs into two sets: the arcs associated to 75 depot-customer pairs become satellite-customer arcs in the TUFLP instance, while the arcs associated with the remaining 75 depot-customer pairs become depot-satellite arcs in the TUFLP instance. We slightly modified the procedure proposed in Landete and Marín (2009) to guarantee that the resulting TUFLP instances are feasible, given the sparsity of the original UFLP instances. Thus, we obtained three sets of instances (“LargeA”, “LargeB” and “LargeC”) with $|I| = |J| = |K| = 75$. We defined costs in such a way that the single assignment constraints have to be imposed explicitly. Thus, these Large Gap instances cannot be solved with formulation (G).

3.2. Bounds at the root node

Tables 2 and 3 report the average gaps (with respect to the optimal value) and CPU times in seconds when computing the

LP relaxations and the root node, respectively, of the six models presented in Section 2.

For instances of type “I”, the LP gaps are identical, irrespective of the model considered. This is not surprising, given that these instances do not have fixed costs associated with satellite location variables and with depot-satellite assignment variables. For instances of type “L”, the LP gaps are equal for models (W) and (W^C), and are dominated by those obtained with models (S), (S_P), (S^C) and (S_P^C), which are identical. This is in accordance with Proposition 3. As expected, LP gaps are small (less than 1%) for instances of type “I” while they are large (around 20%) for instances of type “L”. We observe no difference from one model to the other in terms of computational times. All CPU times are negligible (less than 10 seconds) with the exception of those recorded for the full-sized industrial instances.

When we consider the gap at the root node, the LP gaps are improved marginally even if CPLEX is invoked with its preprocessing and cutting methods. The LP gap is closed for most tiny industrial instances. For the remaining instances, the gap is reduced by less than 0.5%. Considering model (W), no significant impact of such methods is observed at the root node. Last, it is noteworthy that the CPU times increase significantly, even if they remain small.

3.3. Branch-and-bound performance

For a fair comparison of the different formulations, we provide as initial upper bounds to CPLEX the optimal values and deactivate the primal heuristics. We study the performance of CPLEX according to two branching strategies. The choice of the branching variables is done in priority either among the y_i variables or among the z_j variables. Tables 4 and 5 report for priority branching on y_i variables and z_j variables, respectively, the average number

Table 4
Branch-and-bound performance: priority branching on y_i variables.

Type	Instances		(W)	(W ^C)	(S)	(S _P)	(S ^C)	(S _P ^C)
I	Tiny	Nodes	1	1	1	1	1	1
		Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
		Time (s)	0	0	0	0	0	0
	Small	Nodes	35	63	33	34	70	62
		Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
		Time (s)	3	4	3	3	5	4
	Medium	Nodes	16	32	16	15	32	32
		Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
		Time (s)	19	22	18	16	30	20
	Full	Nodes	43	114	41	39	143	117
		Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
		Time (s)	134	269	131	116	399	233
L	LargeA	Nodes	88,812	4949	64,036	70,626	4472	39,482
		Gap (%)	0.01	0.00	0.18	0.01	0.00	0.01
		Time (s)	5860	675	7210	5857	898	3706
	LargeB	Nodes	193,390	6516	137,558	145,892	4940	50,610
		Gap (%)	0.46	0.00	0.88	0.54	0.00	0.01
		Time (s)	14,474	1196	16,788	5857	1198	3706
	LargeC	Nodes	150,798	3938	102,193	104,740	3215	45,483
		Gap (%)	0.96	0.00	1.21	1.08	0.00	0.01
		Time (s)	14,051	854	15,777	13,304	1039	5815

Table 5
Branch-and-bound performance: priority branching on z_j variables.

Type	Instances		(W)	(W ^C)	(S)	(S _P)	(S ^C)	(S _P ^C)
I	Tiny	Nodes	1	1	1	1	2	1
		Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
		Time (s)	0	0	0	0	0	0
	Small	Nodes	43	63	49	70	574,323	62
		Gap (%)	0.00	0.00	0.00	0.00	0.03	0.00
		Time (s)	4	4	4	5	9738	4
	Medium	Nodes	35	32	179	32	3220	32
		Gap (%)	0.00	0.00	0.00	0.00	0.01	0.00
		Time (s)	25	25	66	30	438	20
	Full	Nodes	64	114	977	66	5273	117
		Gap (%)	0.00	0.00	0.01	0.00	0.01	0.00
		Time (s)	182	278	1356	168	5273	233
L	LargeA	Nodes	3704	2708	2728	77,033	1772	39,482
		Gap (%)	0.00	0.00	0.00	0.33	0.00	0.01
		Time (s)	484	477	625	8116	365	3706
	LargeB	Nodes	5297	5592	3893	152,767	2141	50,610
		Gap (%)	0.00	0.00	0.00	0.89	0.00	0.01
		Time (s)	853	1072	1123	17,568	365	3706
	LargeC	Nodes	3726	3291	2547	107,810	1533	45,483
		Gap (%)	0.00	0.00	0.00	1.17	0.00	0.01
		Time (s)	755	741	969	15,976	486	5815

of nodes in the branch-and-bound tree, the gap (with respect to the optimal values) and the CPU time in seconds. A CPU time limit of 36,000 seconds was set. When y_i variables are continuous in the model solved, i.e., (W^C), (S^C) and (S_P^C), the results of branching in priority on y_i variables (Table 4) mean that no priority rule is imposed among all binary variables. Similarly, when z_j variables are not present in the model solved, i.e., (S_P) and (S_P^C), the results of Table 5 show no priority rule among the remaining binary variables.

We first comment on the results for the instances of type “I”. Branching on y_i variables reveals itself to be a good option on such instances. In Table 4, models (W), (S) and (S_P) require less than half the number of nodes, compared with the three other MIP formulations, for which y_i variables are continuous and no priority is imposed among all binary variables. When branching is performed in priority on z_j variables, as shown in Table 5, model (S^C) requires prohibitive computational times, while the best results in terms of nodes and CPU times are obtained with models (W) and (S_P). These results indicate that branching in priority on z_j variables is a poor strategy for these instances. This is easily explained by the

fact that, for these instances, there are no costs associated with these variables. Overall, it is more efficient to branch in priority on y_i variables for the instances of type “I”, with the best results obtained with model (S_P), for which z_j variables are not present.

The picture changes completely when instances of type “L” are solved. Indeed, when branching in priority on y_i variables, models (W), (S), (S_P) and (S_P^C) behave poorly, and the three models for which y_i variables are binary, (W), (S) and (S_P), fail to solve these instances to optimality systematically. In Table 4, the best results are obtained with the two models for which y_i variables are continuous and that include z_j variables, i.e., (W^C) and (S^C). The results reported in Table 5 confirm this behaviour: (S_P) and (S_P^C) requires prohibitive computational times and cannot be solved to optimality for all instances, while the formulation leading to the best performances, both in terms of CPU times and number of nodes, is (S^C). This is explained by the fact that, when branching on z_j variables, t_{ij} variables can be quickly fixed, due to constraints (11). For the same reason, model (S) also performs well, being second after (S^C), but the latter is superior simply because it does not include y_i variables. Overall, it is more efficient to branch

in priority on z_j variables for the instances of type “L”, with the best results obtained with model (S^C), for which the binary requirements on y_i variables are relaxed. Note that, on the instance of Fig. 1, we observed in a similar way the superiority of the branching priority on z_j variables over the branching priority on y_i variables (see the end of Section 2).

It is interesting to note that, for all instances, model (S_p^C) performs relatively poorly, compared with the others. This is explained by the fact that CPLEX has to branch only on t_{ij} variables, since y_i variables are continuous and z_j variables are not present in formulation (S_p^C). In contrast, branching in priority on y_i variables (for instances of type “I”) or z_j variables (for instances of type “L”) is more efficient, since it can quickly fix t_{ij} variables. In spite of being dominated by the other models, mostly because of the branching strategy adopted by CPLEX, model (S_p^C) is interesting and was used as a basis for developing the Lagrangian relaxation method presented in Gendron et al. (2016).

4. Conclusions

We have compared, both theoretically and experimentally, six MIP formulations for the TUFLP-S. The models differ first in the way they define the single assignment constraints: weak models use the most obvious definition that involves only the depot-satellite assignment variables t_{ij} , while strong models introduce a tight connection between variables t_{ij} and the satellite location variables z_j . Using this connection, it is possible to project out the z_j variables, thus obtaining equivalent strong formulations that contain less binary variables. Furthermore, by adding redundant linking constraints between variables t_{ij} and the depot location variables y_i , we can relax the integrality of variables y_i . By using these two techniques, i.e., projecting out variables z_j (only for the strong models) and relaxing the integrality of variables y_i after adding linking constraints (both for the strong and weak models), we have obtained two equivalent weak models, (W) and (W^C), and four equivalent strong models, (S), (S_p), (S^C) and (S_p^C).

On the industrial instances, which have no fixed costs on satellite location variables and depot-satellite assignment variables, our computational results show that it is beneficial to branch first on the y_i variables. The model that shows the best performance on these instances is (S_p), since it reduces the number of binary variables compared to (S), but keeps the integrality of the most significant y_i variables. The model that shows the worst performance on these instances is (S^C): it relaxes the integrality of the y_i variables, while keeping the (meaningless for these instances) z_j variables in the formulation. On the artificial instances, which have significant fixed costs on satellite location variables, our computational results emphasize the benefit of branching first on the z_j variables. The best formulation for these instances is (S^C), since it reduces the number of binary variables compared to (S), while also keeping the most significant z_j variables in the model. The worst model for these instances is (S^p), which projects out the z_j variables, while enforcing the integrality of the less significant y_i variables. For both types of instances, industrial and artificial, the model that minimizes the number of binary variables, (S_p^C), performs poorly. Model (S) is a good compromise, as it includes all types of binary variables. Branching first on the y_i variables is the best approach for industrial instances, but for artificial instances, it is significantly better to branch on the z_j variables.

These results point in the direction of developing more general branching priorities that are adapted to the relative importance of the fixed costs, both for the TUFLP-S, but also for more general multi-level facility location problems. Indeed, it would be interesting to generalize our findings to multi-level facility location problems. In particular, we note that, when fixed costs on intermediate facility locations (here, satellites) are significant, the

problems appear difficult to solve. Developing efficient decomposition methods to handle such difficult problems raises several theoretical and computational challenges.

Acknowledgments

We gratefully acknowledge financial support for this project provided by NSERC (Canada), by the International Campus on Safety and Intermodality in Transportation, the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the French Ministry of Higher Education and Research and the National Center for Scientific Research. We thank the associated editor and two anonymous referees whose comments have helped us to improve our paper.

References

- Aardal, K., Labbé, M., Leung, J.M.Y., Queyranne, M., 1996. On the two-level uncapacitated facility location problem. *INFORMS J. Comput.* 8, 289–301.
- Baïou, M., Barahona, F., 2014. A polyhedral study of a two level facility location model. *RAIRO Oper. Res.* 48, 153–165.
- Barros, I., 1995. *Discrete and Fractional Programming Techniques for Location Models*. Erasmus University, Rotterdam Ph.d. thesis.
- Barros, I., 1998. *Discrete and fractional programming techniques for location models*. *Combinatorial Optimization*, vol. 3. Kluwer Academic Publishers, Dordrecht, Boston, London.
- Barros, I., Labbé, M., 1994. A general model for the uncapacitated facility and depot location problem. In: *Location Science*, 2, pp. 173–191.
- Bloemhof-Ruwaard, J.M., Salomon, M., Van Wassenhove, L.N., 1994. On the coordination of product and by-product flows in two-level distribution networks: Model formulations and solution procedures. *Eur. J. Oper. Res.* 79, 325–339.
- Bloemhof-Ruwaard, J.M., Salomon, M., Van Wassenhove, L.N., 1996. The capacitated distribution and waste disposal problem. *Eur. J. Oper. Res.* 88, 490–503.
- Chardaire, P., Lutton, J.-L., Sutter, A., 1999. Upper and lower bounds for the two-level simple plant location problem. *Ann. Oper. Res.* 86, 117–140.
- Gao, L.-L., Robinson, E.P., 1992. A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. *Nav. Res. Logist.* 39, 191–212.
- Gendron, B., Khuong, P.V., Semet, F., 2015. Multilayer variable neighborhood search for two-level uncapacitated facility location problems with single assignment. *Networks* 66, 214–234.
- Gendron, B., Khuong, P.V., Semet, F., 2016. A lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints. To appear in *Transport. Sci.*
- Gendron, B., Semet, F., 2009. Formulations and relaxations for a multi-echelon capacitated location-distribution problem. *Comput. Oper. Res.* 36, 1335–1355.
- Kaufman, L., Van Eede, M.V., Hansen, P., 1977. A plant and warehouse location problem. *Oper. Res. Q.* 28, 547–554.
- Klose, A., Drexl, A., 2005. Facility location models for distribution system design. *Eur. J. Oper. Res.* 162, 4–29.
- Kochetov, Y., Ivanenko, D., 2005. Computationally difficult instances for the uncapacitated facility location problem. In: *Metaheuristics: Progress as Real Problem Solvers*, pp. 351–367.
- Krarp, J., Pruzan, P.M., 1983. The simple plant location problem: survey and synthesis. *Eur. J. Oper. Res.* 12, 36–81.
- Kratka, J., Dugošija, D., Savić, A., 2014. A new mixed integer linear programming model for the multi level uncapacitated facility location problem. *Appl. Math. Modell.* 38, 2118–2129.
- Landete, M., Marín, A., 2009. New facets for the two-stage uncapacitated facility location polytope. *Comput. Optim. Appl.* 44, 487–519.
- Marín, A., 2006. Lower bounds for the two-stage uncapacitated facility location problem. *Eur. J. Oper. Res.* 179, 1126–1142.
- Marín, A., Pelegrin, B., 1999. Applying lagrangian relaxation to the resolution of two-stage location problems. *Ann. Oper. Res.* 86, 179–198.
- Melo, M.T., Nickel, S., Saldanha da Gama, F., 2006. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Comput. Oper. Res.* 33, 181–208.
- Ortiz-Astorquiza, C., Contreras, I., Laporte, G., 2015. Multi-level facility location as the maximization of a submodular set function. *Eur. J. Oper. Res.* 247, 1013–1016.
- Pirkul, H., Jayaraman, V., 1996. Production, transportation and distribution planning in a multi-commodity tri-echelon system. *Transport. Sci.* 30, 291–302.
- Pirkul, H., Jayaraman, V., 1998. A multi-commodity multi-plant capacitated facility location problem: formulation and efficient heuristic solution. *Comput. Oper. Res.* 25, 869–878.
- Ro, H.-B., Tcha, D.-W., 1984. A branch and bound algorithm for the two-level uncapacitated facility location problem with some side constraints. *Eur. J. Oper. Res.* 18, 349–358.
- Sahin, G., Süral, H., 2007. A review of hierarchical facility location models. *Comput. Oper. Res.* 34, 2310–2331.

Tragantalerngsak, S., Holt, J., Rönnqvist, M., 1997. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *Eur. J. Oper. Res.* 102, 611–625.

Zanjirani Farahani, R., Hekmatfar, M., Fahiminia, B., Kazemzadeh, N., 2014. Hierarchical facility location problem: models, classifications, techniques, and applications. *Comput. Ind. Eng.* 68, 104–117.