HAL open science

Comparison of Formulations for the Two-Level Uncapacitated Facility Location Problem with Single Assignment Constraints

Bernard Gendron, Paul-Virak Khuong, Frédéric Semet

▶ To cite this version:

Bernard Gendron, Paul-Virak Khuong, Frédéric Semet. Comparison of Formulations for the Two-Level Uncapacitated Facility Location Problem with Single Assignment Constraints. Computers and Operations Research, 2017, 86, pp.86 - 93. 10.1016/j.cor.2017.02.020 . hal-01663595

HAL Id: hal-01663595

https://inria.hal.science/hal-01663595

Submitted on 14 Dec 2017

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.

CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Comparison of Formulations for the Two-Level Uncapacitated Facility Location Problem with Single Assignment Constraints

Bernard Gendron
Paul-Virak Khuong
Frédéric Semet

August 2016

CIRRELT-2016-39

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Comparison of Formulations for the Two-Level Uncapacitated Facility Location Problem with Single Assignment Constraints

Bernard Gendron[1,2,*], Paul-Virak Khuong[3], Frédéric Semet[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] AppNexus, 23 West 23rd Street, New York, 4th Floor, USA, NY 10010

[4] École Centrale de Lille, bâtiment C, bureau C343, Cité Scientifique, B.P. 48, 59651 Villeneuve d'Ascq Cedex, France

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.

Keywords. Two-level uncapacitated facility location, mixed-integer programming, relaxations, formulations.

Acknowledgements. We gratefully acknowledge financial support for this project provided by Natural Sciences and Engineering Research Council of Canada (NSERC), 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.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

* Corresponding author: Bernard.Gendron@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

© Gendron, Khuong, Semet and CIRRELT, 2016

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) [18]. 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 [31] and telecommunications [9]. 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 [9].

The literature on multi-level facility location problems [16, 24, 30, 35], which generalize the TUFLP-S, focuses on the comparison of two types of models: arc-based [19, 22, 27, 28] and path-based [3, 11, 15, 29]. The comparison between these different models is performed both from theoretical and experimental perspectives [6, 7, 9, 23]. Several methods have been proposed for solving multi-level facility location problems, based on polyhedral theory [1, 2, 9, 20], Lagrangian relaxation [4, 23, 27, 28], linear programming (LP) relaxation and approximation algorithms [5, 8, 10, 33, 34], greedy algorithms and metaheuristics [3, 9, 13, 21, 25, 26, 32].

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 [14] 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 [3] 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.

Barros and Labbé [3] propose to solve this problem with a MIP formulation that uses the following sets of binary variables:

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

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

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

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

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

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

subject to

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

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

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

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

$$0 \le x_{ijk} \qquad \forall (i,j,k) \in I \times J \times K, \tag{6}$$

$$y_i \in \{0,1\}, \qquad \forall i \in I, \tag{7}$$

$$z_j \in \{0,1\}, \qquad \forall j \in J, \tag{8}$$

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

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. Last, variables x_{ijk} have not to be upper bounded due to constraints (2).

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 [1, 5, 20] 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:

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

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).

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 [14]. 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 $\overline{F}$ 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 $v(\overline{G}) = v(\overline{W}) \leq v(W) = v(G)$.*

Proof. Since $(\overline{G})$ is a relaxation of $(\overline{W})$, we have $v(\overline{G}) \leq v(\overline{W})$.
To show that $v(\overline{G}) \geq v(\overline{W})$, consider an optimal solution to $(\overline{G})$ 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 $(\overline{G})$ that satisfies constraints (10). Hence, $v(\overline{G}) \geq v(\overline{W})$.
The proof of the equation $v(W) = v(G)$ follows the same argument. ■

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:

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

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 *$v(\overline{W}) \leq v(\overline{S})$ and the inequality can be strict.*

Proof. When considering the LP relaxations of (W) and (S), it is also true that (3), (8) and (11) implies (5) and (10). Hence, $(\overline{W})$ is a relaxation of $(\overline{S})$ and $v(\overline{W}) \leq v(\overline{S})$.
To show that the inequality can be strict, we exhibit an instance such that any optimal solution to $(\overline{W})$ satisfies, for some $j' \in J$, $\sum_{i \in I} t_{ij'} > z'_j$. This instance, shown in Figure 1, has origin-destination pairs $i_1 \rightarrow k_1$ and $i_2 \rightarrow k_2$. All fixed and transportation costs are 0, except for the use of satellites j_1 and j_2, with a fixed cost of 1. Model $(\overline{W})$ has only one optimal solution, which uses the two paths from i_1 to k_1 and the two paths

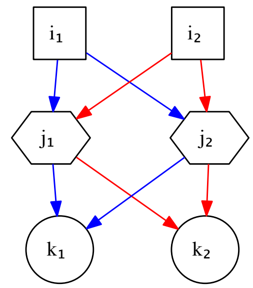

Figure 1: A TUFLP-S instance for which $v(\overline{W}) < v(\overline{S})$.

from i_2 to k_2 with corresponding values of variables x_{ijk} and t_{ij} equal to $\frac{1}{2}$. The values of variables z_{j_1} and z_{j_2} are also equal to $\frac{1}{2}$, for an optimal value $v(\overline{W}) = 1$, and we have $\sum_{i \in I} t_{ij_1} > z_{j_1}$ and $\sum_{i \in I} t_{ij_2} > z_{j_2}$. The solution with the same values of variables x_{ijk} and t_{ij} is also optimal for $(\overline{S})$, but then, we have $z_{j_1} = z_{j_2} = 1$, because of constraints (11), and $v(\overline{S}) = 2$. ■

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):

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

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 (note that (S_P^C) has been used in [14] to derive a Lagrangian relaxation method). Each of these models is obtained from the corresponding one by adding constraints (13) and by relaxing the integrality of variables y_i.

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 $v(\overline{W}) = v(\overline{W^C}) \leq v(\overline{S}) = v(\overline{S_P}) = v(\overline{S^C}) = v(\overline{S_P^C})$.

Proof. To show that, for any formulation $(\overline{F})$, we have $v(\overline{F}) = v(\overline{F}^C)$, we have to show that constraints (13) are redundant for $(\overline{F})$. The argument is based on constraints (3), (4) and the nonnegativity of h_{ij}. Indeed, for any $(i, j) \in I \times J$, $h_{ij} \geq 0$ implies there is an optimal solution to $(\overline{F})$ such that $t_{ij} = \max_{k \in K}\{x_{ijk}\} \equiv x_{ijk^*}$, by constraints (3). Thus, $t_{ij} = x_{ijk^*} \geq \sum_{j' \in J} x_{ij'k^*} \geq y_i$, by constraints (4), and constraints (13) are redundant for $(\overline{F})$.
Finally, the equality $v(\overline{S}) = v(\overline{S_P})$ is trivial and the inequality $v(\overline{W}) \leq v(\overline{S})$ was shown in Proposition 2.

■

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)? Our computational experiments, to be presented next, address these issues.

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 [12]. 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 [14]. 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 [12]. 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, 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 [14]. A set of 30 two-level

Instances	$\|I\|$	$\|J\|$	$\|K\|$	$\|A\|$	$\|P\|$
Tiny	23	80	175	134	3366
Small	47	160	351	592	28496
Medium	70	240	526	1236	92554
Full	93	320	701	2250	222308

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

instances were obtained based on the generator proposed in [17] to obtain UFLP instances with large integrality gaps and on the procedure suggested in [20] to transform them into TUFLP instances. Instances are divided into three classes, A, B and C, with $|I| = |J| = |K| = 75$ ("LargeA", "LargeB" and "LargeC"). 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

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 2: Gaps and runtimes when computing LP relaxations

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. When y_i or z_j variables are not present or are continuous in the model solved, no priority rule is imposed between the remaining binary variables. Tables 4 and 5 report the average number 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.

We first comment on the choice of the branching strategy. Branching on y_i variables reveals itself to be a good option for the solution of instances of type "I". Models (W), (S) and (S_P), compared with other MIP formulations, require less than half the number

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

Table 3: Gaps and runtimes at the the root node

of nodes. However, the picture changes when instances of type "L" are solved. Indeed, models (W), (S), (S_P) and (S_P^C) behave poorly, and three of them, (W), (S) and (S_P), fail to solve these instances to optimality systematically. The formulations leading to the best performances, both in terms of CPU times and number of nodes, are models (W^C) and (S^C), in which the binary requirements on the y_i variables are relaxed.

When branching is performed in priority on the z_j variables, interesting remarks can be made. Several formulations exhibit the same efficiency in terms of nodes and CPU times on instances of type "I". Only model (S^C) requires prohibitive computational times. This indicates 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. It is more efficient to have the option to branch either on the y_i variables or only on t_{ij} variables. The behaviour of the models is more contrasted on instances of type "L", for which the solution of the models including z_j variables, i.e., (W), (W^C), (S) and (S^C), require much shorter computational times than formulations (S_P) and (S_P^C), which cannot be solved to optimality for all instances. CPLEX solves more efficiently the instances using model (S^C) compared with the others,

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	88812	4949	64036	70626	4472	39482
		Gap (%)	0.01	0.00	0.18	0.01	0.00	0.01
		Time (s)	5860	675	7210	5857	898	3706
	LargeB	Nodes	193390	6516	137558	145892	4940	50610
		Gap (%)	0.46	0.00	0.88	0.54	0.00	0.01
		Time (s)	14474	1196	16788	5857	1198	3706
	LargeC	Nodes	150798	3938	102193	104740	3215	45483
		Gap (%)	0.96	0.00	1.21	1.08	0.00	0.01
		Time (s)	14051	854	15777	13304	1039	5815

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

(W), (W^C) and (S).

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

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	574323	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	77033	1772	39482
		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	152767	2141	50610
		Gap (%)	0.00	0.00	0.00	0.89	0.00	0.01
		Time (s)	853	1072	1123	17568	365	3706
	LargeC	Nodes	3726	3291	2547	107810	1533	45483
		Gap (%)	0.00	0.00	0.00	1.17	0.00	0.01
		Time (s)	755	741	969	15976	486	5815

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

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.

References

[1] Aardal, K., Labbé, M., Leung, J.M.Y., Queyranne, M. (1996). On the Two-Level Uncapacitated Facility Location Problem. *INFORMS Journal on Computing* 8, 289-301.

[2] Baïou, M., Barahona, F. (2014). A polyhedral study of a two level facility location model. *RAIRO-Operations Research* 48, 153-165.

[3] Barros, I., Labbé, M. (1994). A General Model for the Uncapacitated Facility and Depot Location Problem. *Location Science* 2, 173-191.

[4] Barros, I. (1995). Discrete and Fractional Programming Techniques for Location Models. Ph.D. Thesis. Erasmus University, Rotterdam.

[5] Barros, I. (1998). Discrete and Fractional Programming Techniques for Location Models. *Combinatorial optimization vol. 3* Kluwer Academic Publishers, Dordrecht, Boston, London.

[6] 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. *European Journal of Operational Research* 79, 325-339.

[7] Bloemhof-Ruwaard, J.M., Salomon, M., Van Wassenhove, L.N. (1996). The Capacitated Distribution and Waste Disposal Problem. *European Journal of Operational Research* 88, 490-503.

[8] Bumb, A. (2001). An approximation algorithm for the maximization version of the two level uncapacitated facility location problem. *Operations Research Letters* 29, 155-161.

[9] Chardaire, P., Lutton, J.-L., Sutter, A. (1999). Upper and Lower Bounds for the Two-Level Simple Plant Location Problem. *Annals of Operations Research* 86, 117-140.

[10] Gabor, A.F., van Ommeren, J.-K.C.W. (2010). A new approximation algorithm for the multilevel facility location problem. *Discrete Applied Mathematics* 158, 453-460.

[11] Gao, L.-L., Robinson, E.P. (1992). A Dual-Based Optimization Procedure for the Two-Echelon Uncapacitated Facility Location Problem. *Naval Research Logistics* 39, 191-212.

[12] Gendron, B., Semet, F. (2009). Formulations and Relaxations for a Multi-Echelon Capacitated Location-Distribution Problem. *Computers & Operations Research* 36, 1335-1355.

[13] 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.

[14] 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 *Transportation Science*.

[15] Kaufman, L., Van Eede, M.V., Hansen, P. (1977). A Plant and Warehouse Location Problem. *Operational Research Quarterly* 28, 547-554.

[16] Klose, A., Drexl, A. (2005). Facility Location Models for Distribution System Design. *European Journal of Operational Research* 162, 4-29.

[17] Kochetov, Y., Ivanenko, D. (2005). Computationally difficult instances for the uncapacitated facility location problem. *Metaheuristics: Progress as real problem solvers*, 351-367.

[18] Krarup, J., Pruzan, P.M. (1983). The Simple Plant Location Problem: Survey and Synthesis, *European Journal of Operational Research* 12, 36-81.

[19] Kratica, J., Dugošija, D., Savić, A. (2014). A new mixed integer linear programming model for the multi level uncapacitated facility location problem, *Applied Mathematical Modelling* 38, 2118-2129.

[20] Landete, M., Marín, A. (2009). New facets for the two-stage uncapacitated facility location polytope. *Computational Optimization and Applications* 44, 487-519.

[21] Maric, M. (2010). An efficient genetic algorithm for solving the multi-level uncapacitated facility location problem. *Computing and Informatics* 29, 183201.

[22] Marín, A. (2006). Lower Bounds for the Two-Stage Uncapacitated Facility Location Problem. *European Journal of Operational Research*, *European Journal of Operational Research* 179, 1126-1142.

[23] Marín, A., Pelegrin, B. (1999). Applying Lagrangian Relaxation to the Resolution of Two-Stage Location Problems. *Annals of Operations Research* 86, 179-198.

[24] 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. *Computers & Operations Research* 33, 181-208.

[25] Mitropoulos, P., Giannikos, I., Mitropoulos, I. (2009). Exact and Heuristic Approaches for the Locational Planning of an Integrated Solid Waste Management System. *Operational Research* 9, 329-347.

[26] Narula, S.C., Ogbu, U.I. (1979) An hierarchal location–allocation problem. *Omega* 7, 137-143.

[27] Pirkul, H., Jayaraman, V. (1996). Production, Transportation and Distribution Planning in a Multi-Commodity Tri-Echelon System. *Transportation Science* 30, 291-302.

[28] Pirkul, H., Jayaraman, V. (1998). A Multi-Commodity Multi-Plant Capacitated Facility Location Problem: Formulation and Efficient Heuristic Solution. *Computers & Operations Research* 25, 869-878.

[29] Ro, H.-B., Tcha, D.-W. (1984). A Branch and Bound Algorithm for the Two-Level Uncapacitated Facility Location Problem with Some Side Constraints. *European Journal of Operational Research* 18, 349-358.

[30] Sahin, G., Süral, H. (2007). A Review of Hierarchical Facility Location Models. *Computers & Operations Research* 34, 2310-2331.

[31] Tragantalerngsak, S., Holt, J., Rönnqvist, M. (1997). Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research* 102, 611-625.

[32] Vilcapoma Ignacio, A.A., Martins Ferreira Filho, V.J., Dieguez Galvao, R. (2008). Lower and Upper Bounds for a Two-Level Hierarchical Location Problem in Computer Networks. *Computers & Operations Research* 35, 1982-1998.

[33] Wu, C., Du, D., Xu, D. (2015). Primal–dual approximation algorithm for the two-level facility location problem via a dual quasi-greedy approach. *Theoretical Computer Science* 562, 213-226.

[34] Zhang, J. (2006). Approximating the two-level facility location problem via a quasi-greedy approach. *Mathematical Programming* 108, 159-176.

[35] Zanjirani Farahani, R., M. Hekmatfar, B. Fahiminia, N. Kazemzadeh. (2014). Hierarchical facility location problem: Models, classifications, techniques, and applications. *Computers & Industrial Engineering* 68, 104-117.