New valid inequalities for the Two-Echelon Capacitated Vehicle Routing Problem

Guillaume Marques[1,2,3] Ruslan Sadykov[2,3] Francois Vanderbeck[3,2]
Jean-Christophe Deschamps[1] Rémy Dupas[1]

[1] Univ. of Bordeaux, IMS, UMR 5218
[2] Inria Bordeaux Sud-Ouest, team RealOpt
[3] Univ. of Bordeaux, IMB, UMR 5251, team Optimal

Mots-clés : *Valid inequalities, Two-Echelon Capacitated Vehicle Routing Problem, Branch-and-Cut-and-Price algorithm*

1 Introduction

In a multi-echelon distribution system, goods are delivered to customers by processing and consolidating items through intermediate depots called satellites. In two-level distribution systems, the first level involves urban trucks that ship items from city distribution centers, located on the outskirts of cities, to satellites. The second level involves city freighters that collect items from satellites and deliver them to customers. The objective of the Two-Echelon Capacitated Vehicle Routing Problem (2ECVRP) is to plan delivery routes of vehicles in such a system at a minimum cost. This problem has been treated using a matheuristic [16], a Branch-and-Cut algorithm [12], a Branch-and-Cut-and-Price one [20], and an ad-hoc exact method [4]. The latter obtains the best results so far and solves instances up to five satellites and 100 customers. The problem has also been tackled using heuristics [7, 11, 6, 22, 1]. The last three achieve the best performance.

In this paper, we suggest a new class of valid inequalities, inspired from depot capacity constraints proposed in [5] for the location-routing problem. The new family of cuts has a significant impact on the strength of the linear relaxation of the problem formulation. We embedded separation of new inequalities in a Branch-and-Cut-and-Price algorithm and solved to optimality 24 open instances of the problem with up to 200 customers and 10 satellites.

2 Formulation

Consider a fleet $\mathcal{K}$ of homogeneous urban trucks of capacity Q_1 traveling on first-level routes and a fleet $\mathcal{L}$ of homogeneous city freighters of capacity Q_2 traveling on second-level routes. We denote by d, $\mathcal{S}$, $\mathcal{C}$ the city distribution center, the sets of satellites and customers. A satellite s can hold up to L_s city freighters and charges f_s^H for each processed item. Each customer should be visited by exactly one city freighter. The problem is to construct feasible first and second-level routes such that each customer c receives d_c items and the flow balance at each satellite is assured. The objective is to minimize the sum of transportation and handling costs.

We work with a path-based formulation for the 2ECVRP. The first level corresponds to the split delivery capacitated vehicle routing problem because several urban trucks can supply a satellite. We model the first level using a complete undirected graph $G_1 = (V_1, E_1)$ where $V_1 = \{d\} \cup \mathcal{S}$. We denote the set of first-level routes as P, and let $\tilde{x}_e^p$ be the number of times path $p \in P$ uses edge $e \in E_1$.

The second level corresponds to the multi-depot vehicle routing problem as each customer is delivered by one city freighter coming from one of the satellites. We model second level using an undirected graph $G_2 = (V_2, E_2)$ where $V_2 = \mathcal{C} \cup S$ and $E_2 = \{(i,j) \mid i \in \mathcal{S} \cup \mathcal{C}, j \in \mathcal{C}, i \neq j\}$. We denote the set of feasible second-level routes starting from satellite s as R_s, and $R = \cup_{s \in \mathcal{S}} R_s$. Let $\tilde{y}_e^r$ be the number of times path $r \in R$ uses edge $e \in E_2$. Let also $\delta_i(v)$ be the set of edges incident to vertex $v \in V_i$ in G_i, and f_e^T be the cost of traversing edge e.

Let variable λ_p equals the number of urban trucks traveling on first-level route $p \in P$. Let variable w_s^p equals the number of items that first-level route $p \in P$ delivers to satellite $s \in S$. Let variable μ_r equals the number of city freighters traveling on second-level route $r \in R$.

$$
\begin{aligned}
(F) \quad \min \quad & \sum_{p \in P} \sum_{e \in E_1} f_e^T \tilde{x}_e^p \lambda_p + \sum_{r \in R} \sum_{e \in E_2} f_e^T \tilde{y}_e^r \mu_r + \sum_{s \in \mathcal{S}} \sum_{r \in R_s} \sum_{c \in \mathcal{C}} \sum_{e \in \delta_2(c)} f_s^H d_c \tfrac{1}{2} \tilde{y}_e^r \mu_r && && (1)\\
\text{s.t.} \quad & \sum_{r \in R} \sum_{e \in \delta_2(c)} \tilde{y}_e^r \mu_r = 2 && c \in V_C && (2)\\
& w_s^p \leq Q_1 \sum_{e \in \delta_1(s)} \tfrac{1}{2} \tilde{x}_e^p \lambda_p && p \in P, s \in \mathcal{S} && (3)\\
& \sum_{s \in \mathcal{S}} w_s^p \leq Q_1 \lambda_p && p \in P && (4)\\
& \sum_{r \in R_s} \sum_{c \in \mathcal{C}} \sum_{e \in \delta_2(c)} \tfrac{1}{2} d_c \tilde{y}_e^q \mu_r \leq \sum_{p \in P} w_s^p && s \in \mathcal{S} && (5)\\
& \sum_{p \in P} \lambda_p \leq |\mathcal{K}| && && (6)\\
& \sum_{r \in R_s} \mu_r \leq L_s && s \in \mathcal{S} && (7)\\
& \sum_{r \in R} \mu_r \leq |\mathcal{L}| && && (8)\\
& \lambda \in \mathbb{N}^{|P|} && && (9)\\
& \mu \in \mathbb{N}^{|R|} && && (10)\\
& w_s^p \geq 0 && p \in P, s \in \mathcal{S} && (11)
\end{aligned}
$$

Objective function (1) minimizes the sum of travel and handling costs. Constraint (2) ensures that only one second-level route visits a customer. Constraint (3) ensures that a first-level route p delivers items to a satellite s if and only if p visits s. Constraint (4) specifies that an urban truck cannot deliver more items than its capacity. Constraint (5) ensures that urban trucks deliver enough items to any satellite s to cover the demand of customers served from s. Constraints (6), (7) and (8) are upper bounds on the number of used vehicles.

3 Satellite Supply Inequalities

Now, let us introduce Satellite Supply Inequalities (SSI). Given a solution to the linear relaxation of (F), we wish to strengthen the formulation by ensuring that any subset $S \subset \mathcal{S}$ of satellites can supply all the demand of customers served by satellites in S. We will use the following aggregated variables. Let $x_e = \sum_{p \in P} \tilde{x}_e^p \lambda_p$ be the number of times edge $e \in E_1$ is used by urban trucks. Let $y_e^s = \sum_{r \in R_s} \tilde{y}_e^r \mu_r$ be the number times edge $e \in E_2$ is used by city freighters coming from satellite $s \in \mathcal{S}$.

3.1 Inequalities

First, let us look at the solution to the linear relaxation of (F) pictured in Figure 1. Consider urban trucks with capacity $Q_1 = 10$ items and city freighters with capacity $Q_2 = 6$ items. Let C be a subset of customers asking for $d_C = 11$ items and supplied by a subset of satellites $S = \{s\}$. One urban truck delivers at most 10 items to s. Two second-level routes start at s.

Those routes use 1.8 city freighters and must deliver $d_C = 11$ items to cover the total demand of visited customers. Clearly, s cannot supply d_C items. Thus the corresponding SSI states that either more urban trucks should visit s or a city freighter coming from $S^{\complement} = \mathcal{S} \setminus S$ should visit at least one customer in C to deliver the missing item. Since only 0.2 city freighters come from $S^{\complement}$, this fractional solution violates this SSI.

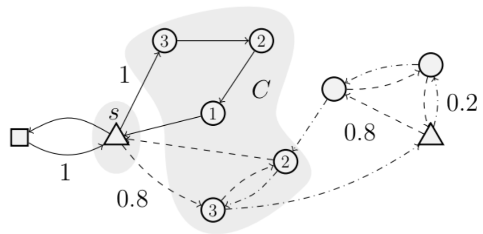

FIG. 1 – Example of a fractional solution violating an SSI. The square is the city distribution center, triangles are satellites, and circles are customers. Demand of a customer is written inside the circle.

Let aggregated variable u_S denote the number of urban trucks visiting a non-empty subset $S \subseteq \mathcal{S}$ of satellites. Let $P_S = \{p \in P \mid \exists s \in S, e \in \delta_1(s) : \tilde{x}_e^p = 1\}$. Then $u_S = \sum_{p \in P_S} \lambda_p$. Let also $g^C(u)$ be the function which counts the minimum number of city freighters required to cover the demand of subset $C \subset \mathcal{C}$ of customers that $\lfloor u \rfloor$ urban trucks cannot supply :

$$g^C(u) = \max\left\{0, \left\lceil \frac{\sum_{c \in C} d_c - \lfloor u \rfloor Q_1}{Q_2} \right\rceil \right\} \tag{12}$$

Proposition 1 *Given a subset $C \subset \mathcal{C}$ of customers and a subset $S \subset \mathcal{S}$ of satellites, let $C^{\complement} = \mathcal{C} \setminus C$ and $S^{\complement} = \mathcal{S} \setminus S$. Then the following inequality is valid for the 2ECVRP :*

$$\sum_{s \in S^{\complement}} \sum_{i \in C} \sum_{j \in S^{\complement} \cup C^{\complement}} y^s_{(i,j)} \geq 2 g^C(u_S) \tag{13}$$

Note that $g^C(u)$ is a non-linear function and cannot be used directly. Instead, we use piece-wise linear function $h^C(u)$ which forms the convex-hull of the epigraph of $g^C(u)$. Let $k(C)$ be the minimum number of urban trucks required to cover the demand of C. We denote as $\tilde{u}^C$ the ordered vector of (integer) values u of extreme points of h^C : $\tilde{u}^C = (\tilde{u}^C_0 = 0, \tilde{u}^C_1, \ldots, \tilde{u}^C_{k(C)})$. Figure 2 depicts an example of functions g^C and h^C. In the left plot, the epigraph of g^C is the grey area. In the right plot, function h^C is the bold line. Extreme points of h^C are H_0, H_2, H_3, H_4, but not H_1. Therefore, $\tilde{u}^C = (0, 2, 3, 4)$.

Proposition 2 *Given a subset $C \subset \mathcal{C}$, $S \subset \mathcal{S}$, and an integer $0 < k \leq k(C)$, the following inequality is valid for the 2ECVRP*

$$\sum_{s \in S^{\complement}} \sum_{i \in C} \sum_{j \in S^{\complement} \cup C^{\complement}} y^s_{(i,j)} \geq 2 \cdot \left(g^C(\tilde{u}^C_{k-1}) - \frac{g^C(\tilde{u}^C_{k-1}) - g^C(\tilde{u}^C_k)}{\tilde{u}^C_k - \tilde{u}^C_{k-1}} \cdot (u_S - \tilde{u}^C_{k-1}) \right). \tag{14}$$

The right-hand side of each constraint (14) corresponds to a linear piece of function h^C. Proposition 2 follows from Proposition 1 and from the fact that $h^C(u) \leq g^C(u)$ for all $u \geq 0$.

3.2 Separation

We will now address the question of how to separate the family of SSI efficiently. Given a solution to the the linear relaxation of (F), let $(\bar{y}, \bar{u})$ be the values of variables y and u. Then the problem (15) finds the most violated SSI.

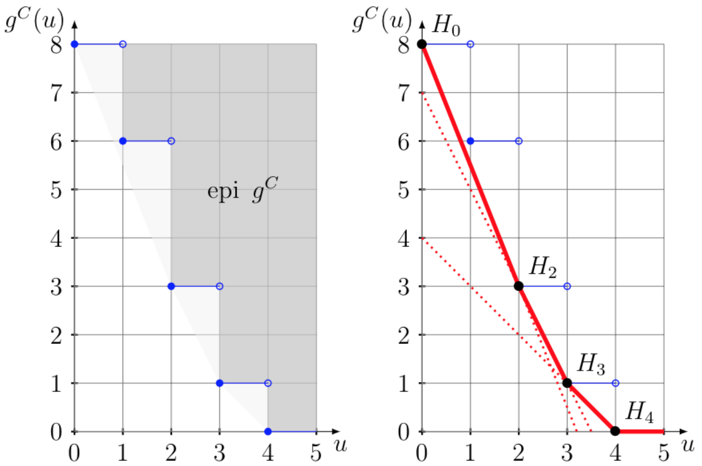

FIG. 2 – Example of functions $g^C(u)$ and $h^C(u)$ for $Q_1 = 10$, $Q_2 = 4$, and $\sum_{c \in C} d_c = 32$.

$$\max_{S \subset \mathcal{S}, C \subset \mathcal{C}} 2h^C(\bar{u}) - \sum_{s \in S^{\complement}} \sum_{i \in C} \sum_{j \in S^{\complement} \cup C^{\complement}} \bar{y}^s_{(i,j)} \tag{15}$$

Henceforth, we denote as $\rho(S, C)$ the objective function of (15) for given $C \subset \mathcal{C}$ and $S \subset \mathcal{S}$. Note that the first and second terms of (15) are non-linear functions of C and S. Thus, enumeration of C and S is required to solve (15) exactly. The second term of (15) is the value of the cut $\mathscr{C}(C, S^{\complement} \cup C^{\complement})$ of the second-level graph induced by vertices $\mathcal{C} \cup S^{\complement}$ with value function $e \to \sum_{s \in S^{\complement}} \bar{y}^s_e$, $e \in E_2$.

We propose a heuristic to separate SSI. Although the heuristic does not necessarily find the most violated inequality, it offers a good trade-off between computational effort and the violation value of the inequalities found. First, we pick certain subsets of satellites. Then, for each of these subsets, we look for the minimum cut in the second-level valued subgraph induced by $\mathcal{C} \cup S^{\complement}$. Then, we try to find other violated inequalities by augmenting the current subset C of customers.

Proposition 3 *Consider the two following Satellite Supply Inequalities :*

$$\sum_{s \in S_1^{\complement}} \sum_{i \in C_1} \sum_{j \in S_1^{\complement} \cup C_1^{\complement}} y^s_{(i,j)} \geq b_1 - a_1 u_{S_1} \tag{16}$$

$$\sum_{s \in S_2^{\complement}} \sum_{i \in C_2} \sum_{j \in S_2^{\complement} \cup C_2^{\complement}} y^s_{(i,j)} \geq b_2 - a_2 u_{S_2} \tag{17}$$

If $a_1 \leq a_2$, $b_1 \geq b_2$, $S_2 \subseteq S_1$, $C_1 = C_2$ *and* $\bar{u}_{S_1} = \bar{u}_{S_2}$, *violation of inequality* (16) *is not smaller than that of* (17) *for* $(\bar{y}, \bar{u})$.

Using Proposition 3, we can avoid separating inequalities for all subsets of satellites. We use procedure SatsSubsetsFiltering (not presented here due to the lack of space) that returns the subsets of satellites to separate. The procedure removes a subset S of satellites from the list of subsets to separate if there exists a subset S' such that $S \subsetneq S'$ and $\bar{u}_S = \bar{u}_{S'}$.

The separation algorithm retrieves subsets of satellites to separate. Then, for each subset S to separate, the next steps are performed. First, we split customers into three sets : (1) the set U_I of customers not visited by any route starting from satellites in S, (2) the set C_I of customers only visited by routes coming from satellites in S, and (3) the set $D = \mathcal{C} \setminus (C_I \cup U_I)$. Then, we build the separation valued graph G_S. First, we form the subgraph of G_2 induced by vertices $U_I \cup D \cup S^{\complement}$. Second, we set the value of each edge $e \in G_2$ to $\sum_{s \in S^{\complement}} \bar{y}^s_e$. Finally, we contract edges for which both extremities are in $U_I \cup S^{\complement}$ so that one vertex in G_S represents vertices in $U_I \cup S^{\complement}$ in G_2. Once G_S built, we compute the minimum cut $\mathscr{C}(C_D, U_D)$ of G_S and verify violation of all inequalities (14) for the subset S of satellites and subset C_D of customers. The value of $h^C(\bar{u}_S)$ increases when the total demand d_C of customers in C increases. Therefore, we may find more violated inequalities by augmenting set C_D. We thus greedily choose a customer in U_D that increases the total demand not supplied by S while not increasing much the value of the cut. We verify the violation of inequalities again. We repeat in the same manner until set U_D is empty. Finally, we keep the $n = 150$ most violated inequalities. Algorithm 1 summarizes the separation algorithm for SSI.

Algorithm 1 Satellites Supply Inequalities separation

```
SubsetsToSeparate ← SatsSubsetsFiltering(𝒮, ū)
for all S ∈ SubsetsToSeparate do
    Build the separation graph G_S
    Compute a minimum cut 𝒞(C_D, U_D) of G_S
    repeat                                          ▷ Expansion of covered customers
        if ρ(S, C_I ∪ C_D) > 0 then (14) is violated
        c ← argmax_{c∈U_D} { Σ_{s∈S^∁} Σ_{e∈δ_2(c)} ȳ^s_e d_c / Σ_{s∈S^∁} Σ_{i∈U_D\{c}} ȳ^s_{(c,i)} }   ▷ Pick a customer c
        C_D ← C_D ∪ {c}                             ▷ Move customer c
        U_D ← U_D \ {c}
    until U_D = ∅
end for
Keep the n most violated inequalities
```

4 Branch-and-Cut-and-Price Algorithm

We start by noticing that the number of variables λ and μ in formulation (F) is exponential. However, while solving the linear relaxation of (F), we dynamically add variables μ using delayed column generation. Considering that the number of satellites in the literature instances does not exceed 10, a standard approach is to enumerate first-level routes by computing the minimum cost path visiting each subset of satellites. The pricing problem for second-level routes is decomposed into subproblems, one per satellite. Each of them is the resources constrained shortest path problem which is solved using the bucket graph based labeling algorithm [18] including the bucket arc elimination procedure. In pricing subproblems, the only resource is the city freighter capacity. We define the capacity consumption of edge (i, j) as $\frac{1}{2}(d_i + d_j)$, $i, j \in V_2$, ($d_i = 0$ for all $i \in \mathcal{S}$) to allow the labeling algorithm to exploit the forward-backward path symmetry as explained in section 3.6 of [18]. We define the accumulated capacity consumption interval as $[0, Q_2]$ for each vertex $v \in V_2$. We use the ng-path relaxation [3] to speed up the labeling algorithm at the cost of a slightly worse column generation bound.

We solve (F) using the Branch-and-Cut-and-Price algorithm presented in details in section 5 of [18]. We solve the linear relaxation of (F) as described in the previous paragraph. We use automatic dual price smoothing technique [17] to stabilize column generation. To strengthen the linear relaxation of (F), besides SSI, we separate limited memory Rank-1 Cuts (R1C) [15] based on set-partitioning constraints (2), and Rounded Capacity Cuts (RCC) [14] that define lower bounds on the number of city freighters that must visit a subset of customers. A tailing-off condition stops the separation of cuts in a node. We also use the enumeration of elementary routes that have reduced cost smaller than the current gap [2]. Our experiments showed that it is essential to branch first on aggregated variables u. Once all of them are integer, we branch on variables λ, y, on the number of city freighters starting from a satellite, and on the assignment of customers to satellites. We use multi-phase strong branching similar to [15] to choose the most promising branching candidate. We solve the restricted master (with the time limit of 20 seconds) on every node of the search tree to find feasible solutions.

5 Experiments and Results

The model and the separation algorithm for SSI were implemented in Julia 0.6 using JuMP [9] and LightGraphs packages. We used the BaPCod library [21] which implements the Branch-and-Cut-and-Price framework including the labeling algorithm and the separation of Rank-1 Cuts. The CVRPSEP package [13] separated Rounded Capacity Cuts. We used IBM CPLEX Optimizer version 12.8.0 as the LP solver in column generation and as the solver for the enumerated MIPs. Experiments were run on a 2 Dodeca-core Haswell Intel Xeon E5-2680 v3 server at 2.5 GHz. On each server, we solved 24 instances that share 128Go of RAM. Each instance is solved on a single thread. Table 1 shows the sets of instances from the literature that we used. Set 4B limits the size of city freighters parks at satellites making constraint (7) required. Set 5 duplicates each instance : the first instance with a city freighter with low capacity, and the second one with a city freighter with high capacity. Set 6B has non-zero handle costs.

Set	nb of instances	nb of sat.	nb of cust.	Notes	Authors
3	18	2	21, 32 or 50		[10]
4A	54	2, 3 or 5	50	$L_s < \|\mathcal{L}\|$	[8]
4B	54	2, 3 or 5	50		[8]
5	18	5 or 10	100 or 200	low and high Q_2	[11]
6A	27	4, 5 or 6	50, 75 or 100		[4]
6B	27	4, 5 or 6	50, 75 or 100	$f_s \neq 0$	[4]

TAB. 1 – Sets of instances used for experiments

In the first experiment, we estimated the effect of SSI on the primal-dual gap of the root node. Table 2 shows the results. The second and the third columns in the table give the results for column-and-cut generation algorithm at the root node using only R1C and RCC. Here “Gap” is the average gap between the value of the linear relaxation of (F) and the best solution from the literature, and “Time” is the average running time in seconds. The fourth and fifth columns show the same statistics for the variant in which also SSI are separated. We see that the separation of SSI decreases the root gap significantly. The separation of SSI has the largest effect on instances of sets 6A and 6B for which the root gap decreased by a factor of 9. The increase of the root solution time is reasonable. Moreover, for each subset of satellites, we compared the most violated SSI found by our heuristic and the most violated SSI found by the exact MIP separation with the time limit of one minute. On average, the maximum violation found by the MIP is $\approx 25\%$ larger than the one found by our heuristic. Therefore, there is still a room for improvement of the separation heuristic.

	R1C + RCC		R1C + RCC + SSI	
Set	Gap	Time	Gap	Time
3	2.47%	16.4	1.65%	40.0
4A	5.33%	43.6	2.86%	116.4
4B	4.14%	48.6	1.90%	122.7
5	5.12%	206.6	1.11%	924.9
6A	6.85%	39.3	0.70%	166.8
6B	3.07%	44.6	0.35%	153.3

TAB. 2 – Effect of SSI on the root node

In the second experiment, we tested the two variants of the Branch-and-Cut-and-Price algorithm (with and without SSI) on all instances of Table 1 and compared them to the state-of-the-art algorithm by Baldacci et al. [4]. As in the latter, no initial primal solution was used by our algorithm, and we rely on the restricted master heuristic to find feasible solutions. For every instance, we set the time limit to three hours. Table 3 presents comparison between algorithms. In this table : "Opt" is the number of instances solved to optimality within the time limit, "Time" is the average solution time, and "Nodes" is the average number of nodes in the branching tree. Note that the solution time of [4] was divided by 1.65 to take into account the difference in the computers used. Also, note that the algorithm of [4] is based on an intelligent enumeration of *combinations* of first-level routes and thus unable to tackle instances with 10 satellites (12 from 18 instances in set 5 are with 10 satellites). Our algorithm is free of this drawback and was tested on all instances of set 5.

	Baldacci et al. [4]		BCP (R1C + RCC)			BCP (R1C + RCC + SSI)		
Set	Opt	Time	Opt	Time	Nodes	Opt	Time	Nodes
3	18/18	98.9	18/18	39.3	4.1	18/18	35.9	3.7
4A	50/54	982.4	54/54	802.4	9.8	54/54	723.1	7.1
4B	52/54	778.3	54/54	375.0	10.6	54/54	329.2	8.2
5	3/6	10,563.0	5/18	9,122.9	93.2	9/18	7,493.1	60.3
6A	22/27	5,997.0	26/27	1,113.1	32.9	26/27	859.5	15.8
6B	19/27	3,362.0	26/27	1,400.3	27.1	27/27	466.8	9.7

TAB. 3 – Comparison of the BCP to the state-of-the-art

Table 3 shows that BCP without SSI already outperforms the state-of-the-art, as it solved 19 open instances to optimality. The average solution time is also significantly lower. Separation of SSI allows us to solve five more open instances. The effect of these valid inequalities is especially noticeable for largest and hardest instances (sets 5 and 6B). For the first time, several instances with as much as 200 customers and 10 satellites are solved. Thus we double the size of instances which can be solved to optimality.

6 Conclusion

In this paper, we proposed a new class of valid inequalities for the two-echelon capacitated vehicle routing problem and a separation heuristic. We showed that Satellite Supply Inequalities embedded in a recent Branch-and-Cut-and-Price algorithm allow us to solve five more open instances of the problem. Moreover, the total number of solved open instances is 24. Our algorithm is the first one being able to solve instances with 10 satellites.

Further enhancements of our algorithm can be undertaken. First, we believe that it is possible to improve the separation heuristic for SSI, both the subroutine which selects subsets of satellites to separate and the one which finds violated inequalities when this subset is already fixed. Second, the performance of the restricted master heuristic is not satisfactory for instances that have city freighters with large capacity. A variant of the diving heuristic [19] or an ad-hoc heuristic based on a fractional solution of formulation (F) may improve the results.

Another direction for further research concerns extension of our algorithm to the two-echelon location-routing problem and also to the two-echelon vehicle routing problem with time windows. Extension to the case when enumeration of first-level routes is not possible is also needed.

Références

[1] Y. Amarouche, R.N. Guibadj, and A. Moukrim. A Neighborhood Search and Set Cover Hybrid Heuristic for the Two-Echelon Vehicle Routing Problem. In *ATMOS 2018*, volume 65, pages 11 :1–15, Dagstuhl, Germany, 2018.

[2] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115 :351–385, 2008.

[3] R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5) :1269–1283, 2011.

[4] R. Baldacci, A. Mingozzi, R. Roberti, and R. Wolfler Calvo. An Exact Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem. *Operations Research*, 61(2) :298–314, 2013.

[5] J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler Calvo. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers and Operations Research*, 38(6) :931–941, 2011.

[6] U. Breunig, V. Schmid, R.F. Hartl, and T. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research*, 76 :208–225, 2016.

[7] T.G. Crainic, S. Mancini, G. Perboli, and R. Tadei. Grasp with path relinking for the two-echelon vehicle routing problem. pages 113–125, 2013.

[8] T.G. Crainic, G. Perboli, S. Mancini, and R. Tadei. Two-echelon vehicle routing problem : A satellite location analysis. *Procedia - Social and Behavioral Sci.*, 2(3) :5944–5955, 2010.

[9] I. Dunning, J. Huchette, and M. Lubin. JuMP : A modeling language for mathematical optimization. *SIAM Review*, 59(2) :295–320, 2017.

[10] J. Gonzalez-Feliu, G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem. 2008.

[11] V.C. Hemmelmayr, J.F. Cordeau, and T.G. Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12) :3215–3228, 2012.

[12] M. Jepsen, S. Spoorendonk, and S. Ropke. A Branch-and-Cut Algorithm for the Symmetric Two-Echelon Capacitated Vehicle Routing Problem. *Transportation Science*, 47(1) :23–37, feb 2013.

[13] J. Lysgaard. *CVRPSEP : A package of separation routines for the capacitated vehicle routing problem*. Aarhus School of Business, Department of Management Science and Logistics, 2003.

[14] J. Lysgaard, A.N. Letchford, and R.W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2) :423–445, 2004.

[15] D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1) :61–100, 2017.

[16] G. Perboli, R. Tadei, and D. Vigo. The Two-Echelon Capacitated Vehicle Routing Problem : Models and Math-Based Heuristics. *Transportation Science*, 45(3) :364–380, 2011.

[17] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2) :339–360, 2018.

[18] R. Sadykov, E. Uchoa, and A. Pessoa. A bucket graph based labeling algorithm with application to vehicle routing. Cadernos do LOGIS 7, Universidade Federal Fluminense, October 2017.

[19] R. Sadykov, F. Vanderbeck, A. Pessoa, I. Tahiri, and E. Uchoa. Primal heuristics for branch-and-price : the assets of diving methods. *INFORMS Journal on Computing*, (Forthcoming), 2018.

[20] F.A. Santos, G.R. Mateus, and A.S. da Cunha. A Branch-and-Cut-and-Price Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem. *Transportation Science*, 49(2) :355–368, may 2015.

[21] F. Vanderbeck, R. Sadykov, and I. Tahiri. Bapcod–a generic branch-and-price code. 2018.

[22] K. Wang, Y. Shao, and W. Zhou. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D : Transport and Environment*, 57 :262 – 276, 2017.