ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2018, Vol. 57, No. 3, pp. 453–462. © Pleiades Publishing, Ltd., 2018.

SYSTEMS ANALYSIS
AND OPERATIONS RESEARCH

Linear Formulations for the Vehicle Routing Problem with Synchronization Constraints[1]

E. López-Aguilar[a,*], V. Boyer[a], M. A. Salazar-Aguilar[a], and M. Sbihi[b]

[a] *Graduate Program in Systems Engineering, UANL, México*

[b] *ENAC, Toulouse, France*

**e-mail: eduardo.lopezag@uanl.edu.mx*

Received August 12, 2015; in final form, March 20, 2017

Abstract—This paper studies a vehicle routing problem with synchronization constraints and time windows. In this problem, a subset of nodes requires more than one vehicle to satisfy its demand simultaneously. We propose three new mixed integer linear formulations for this problem and we evaluate their efficiency over a large set of instances taken from the literature. The computational results reveal that the proposed linear formulations allow solving larger instances in a shorter computational time than the ones previously proposed in the literature.

DOI: 10.1134/S106423071803005X

INTRODUCTION

The Vehicle Routing Problem (VRP) was first proposed by Dantzig and Ramser [1] as a generalization of the Traveling Salesman Problem (TSP). As the TSP, the VRP has been widely studied in the literature (see for instance Golden et al. [2], Toth and Vigo [3], and Laporte [4]). In this work, we focus on a variant of the VRP known as the Synchronized Vehicle Routing Problem with Time Windows, SVRP (also called Vehicle Routing Problem with Multiple Synchronization Constraints—VRPMS). In this problem, in addition to the classical constraints of the VRP, some customers need more than one vehicle to fulfill a task or operation, and in this case the visits of the vehicles should be synchronized in order to provide the service simultaneously. Synchronization constraints generally lead to more complex problems and involve non-linear formulations. In this context, we present a literature review of linear formulations that have been proposed and, from their analysis three new linear formulations are derived, which allow solving larger instances.

Drexl [5] presents a complete review of the different variants of the VRP with synchronization constraints and proposes the following classification:

—Task synchronization:

"Each task must be performed exactly once by one or more suitable vehicle(s)."

—Operation synchronization:

"The offset, that is, the time that may elapse between the start of execution of a specified operation by a suitable vehicle at a certain vertex and the start of execution of another specified operation by another suitable vehicle at another certain vertex, must lie within a specified finite interval of zero or positive length, both vehicles must be compatible, and the vertices may be the same one or different ones".

—Movement synchronization:

"For a vehicle to be able to move along a certain arc, a different but compatible vehicle must move along the same arc at the same time, that is, both vehicles must leave the tail of the arc at the same time, traverse the arc together and reach the tail of the arc at the same time".

—Load synchronization:

"For each vertex with specified negative, zero, or positive demand, the difference between the total amount of load unloaded at the vertex by all active vehicles visiting it and the total amount of load received at the vertex by all passive vehicles visiting it must be equal to the specified demand."

[1] The article is published in the original.

—Resource synchronization:

"The total consumption of a specified resource by all vehicles must be less than or equal to a specified limit."

In this paper, we focus on the linear formulation of operations synchronization when the offset is equal to zero, and the synchronization occurs on the same vertex. In this case, a vehicle may wait at a vertex that the other required vehicles arrive in order to start the operation. Furthermore, we consider time windows on the operation starting time. This problem can have many practical applications, such as maintenance operations, transshipment operations, teamwork scheduling, etc.

This paper is organized as follows. Section 1 presents a literature review of the SVRP. In Section 2, the linear formulations of the SVRP that can be found in the literature are discussed. Section 3 is dedicated to our proposed linear formulations. Computational results are presented in Section 4, followed by some conclusions and the future work in Section 5.

1. LITERATURE REVIEW

In this section, we mention the most relevant publications that include synchronization constraints in a VRP. Amaya et al. [6] introduce the Capacitated Arc Routing Problem (CARP) with refill points for the problem of road marking. The objective of the CARP is to synchronize, at minimum cost, the routes of the marking vehicles with the refill trucks that will replenish them. The authors propose an integer linear model for this problem and use a cutting plane approach for its solution.

Salazar-Aguilar et al. [7] propose an extension of the CARP, the Synchronized Arc Routing and Node Routing Problem (SANRP). This problem considers multiple capacitated marking vehicles and multiple replenishment vehicles for the paint. The authors propose a non-linear formulation and an Adaptive Large Neighborhood Search metaheuristic (ALNS, see Ropke and Pisinger [8]) for the SANRP.

In another work, Salazar-Aguilar et al. [9] introduce a synchronized arc routing problem inspired from snow plowing operations. In this problem, synchronization occurs on the arcs, i.e. streets where maintenance is required, since streets with multiple lines in the same direction should be plowed simultaneously in order to prevent the formation of snow mounds. A non-linear mixed integer formulation and an ALNS procedure are proposed. The authors present results over a set of artificial instances and on a real instance from the city of Dieppe, New Brunswick, Canada.

Ioachim et al. [10] consider an aircraft fleet routing and scheduling problem that tries, for a given set of aircrafts, to cover a set of flights with a flexible schedule in the departure time, at minimum cost. They propose a group of synchronization constraints called same-departure time, where the flights with the same identifier are assigned on different days of the week but their departure time needs to be at the same hour in every day assigned. A multicommodity formulation and a Branch and Bound algorithm with an embedded column generation are presented. Besides, we can cite the work of Bélanger et al. [11] who worked on a similar problem.

Crainic et al. [12] propose a general framework for modeling city logistic systems. Here, synchronization is required for the transshipment of goods. The presented model is based on a discretization of the planning horizon and the duplication of nodes for each pair transshipment location/period. The authors consider a hierarchical decomposition procedure to solve this problem.

Synchronization constraints can also be found in scheduling problems, and many authors used a VRP formulation to model these problems [see Drexl [5]]. In this case, synchronization occurs when teamwork is required to accomplish a particular task, and the schedule of the employees should be design accordingly.

Li et al. [13] study the Manpower Allocation Problem (MAP). In this problem, given a set of workers with different skills and a set of work locations, which require a composed team to perform a job, the goal is to minimize the traveling time between locations and the total number of workers. The authors formulate this problem as a VRP, where a worker plays the role of one vehicle and each work location is a node, resulting in an integer-programming model. They use a simulated annealing metaheuristic as a solution procedure.

Similar works can be found in Dohn et al. [14] and Kim et al. [15]. Dohn et al. [14] propose a similar formulation than Li et al. [13] and propose a Branch and Price approach to solve the problem. Kim et al. [15] propose a Particle Swarm Optimization metaheuristic (Poli et al. [16]) for solving the MAP.

Eveborn et al. [17] introduce a Home Health Care (HHC) problem, in this problem the authors deal with an application for elderly people that require medical services that involve more than one doctor and/or nurse. They formulate this problem as a minimum matching problem.

Bredström and Rönnqvist [18] and Bredström and Rönnqvist [19] propose a Mixed Integer Programming (MIP) formulation for the HHC based on the VRP with time windows, synchronization, and precedence constraints. Both articles use the same model, however, in the first one, they implement a heuristic procedure and in the second one, they propose a Branch and Price algorithm for this problem.

The most recent work for the HCC problem can be found in Labadie et al. [20]. The authors propose a different MIP formulation based on the VRP with time windows. The difference lies in the way they formulate the synchronization constraints. Furthermore, they present a heuristic procedure.

Besides of the previous cited works, references can also be done to Rosa et al. [21] and Del Pia and Filippi [22] who study variants of the SVRP, however no linear formulation is reported by these authors.

Therefore, to the best of our knowledge, these are the only linear formulations that have been proposed in the literature for the SVRP or that can be considered for the SVRP. Furthermore, no comparison of these models has been presented. In the next sections, these formulations are described and compared with the new formulations that we propose.

2. LINEAR FORMULATIONS FOR THE SVRP

Let $G = (N, A)$ be a directed graph, where $N = \{1, 2, \ldots, n\} \cup \{o, d\}$ is the set of nodes and $A = \{(i, j) \mid i \in N \backslash \{d\},\ j \in N \backslash \{o\},\ i \neq j\}$ is the set of arcs. The nodes $\{1, 2, \ldots, n\}$ represent the customers that require a service and the nodes o and d are the initial and the final depot, respectively, which may be at the same location. Each node $i \in N$ has an associated servicing time s_i ($s_o = s_d = 0$), a demand of n_i vehicles that must provide the service simultaneously and a time window $[a_i, b_i]$ for the starting time of the service. Each node has an associated traveling time T_{ij}. The set of available vehicles is denoted by V. We define then the following routing variables:

$$x_{ij}^{v} = \begin{cases} 1, & \text{if the arc } (i, j) \in A \text{ is used by vehicle } v \in V, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, the classic routing problem (without synchronization) can be formulated as follows:

$$\min \sum_{(i,j)\in A} \sum_{v\in V} T_{ij} x_{ij}^{v}, \tag{2.1}$$

$$\sum_{j\in N\backslash\{o\}} x_{oj}^{v} = 1, \quad \forall v \in V, \tag{2.2}$$

$$\sum_{j\in N\backslash\{d\}} x_{jd}^{v} = 1, \quad \forall v \in V, \tag{2.3}$$

$$\sum_{i\in N\backslash\{d\}} x_{ij}^{v} - \sum_{k\in N\backslash\{o\}} x_{jk}^{v} = 1, \quad \forall v \in V, \tag{2.4}$$

$$x_{ij}^{v} \in \{0,1\}, \quad \forall v \in V, \quad \forall i, j \in N\backslash\{o, d\}. \tag{2.5}$$

The objective (2.1) is to minimize the total traveling time of the vehicles. Constraints (2.2) and (2.3) are the common VRP constraints to guarantee the departure from the depot and the arrival to the depot, respectively. Constraints (2.4) impose flow conservation. Note that, the formulation of the demand in service for each node depends on the choice of the scheduling variables that will be described in the next sections.

In the following models, we will detail the linear formulation of the scheduling constraints induced by the synchronization. These scheduling constraints also allow the elimination of subtours since the time consistency should be satisfied. There are mainly two approaches for the scheduling variables that have been used in the literature.

2.1. SVRP1

The first approach consists in using the scheduling variable t_i^{v}, which represents the starting time of the service at node $i \in N$ by vehicle $v \in V$. These scheduling variables have been used by Ioachim et al. [10], Bélanger et al. [11], Bredström and Rönnqvist [18], Bredström and Rönnqvist [19], and Labadie et al. [20].

Let $S \subset N \times N$ be the set of pairwise synchronized visits. Note that, in our case the synchronized visit occurs on the same node, hence N should contain n_i duplicates of node i. The demand and scheduling constraints are then formulated as follows:

$$\sum_{v \in V} x_{ij}^v = 1, \quad \forall j \in N \backslash \{o, d\}, \tag{2.6}$$

$$t_i^v + (s_i + T_{ij}) x_{ij}^v \le t_j^v + b(1 - x_{ij}^v), \quad \forall (i, j) \in A, \quad \forall v \in V, \tag{2.7}$$

$$a_i \sum_{(i,j) \in A} x_{ij}^v \le t_i^v \le b_i \sum_{(i,j) \in A} x_{ij}^v, \quad \forall i \in N \backslash \{o, d\}, \quad \forall v \in V, \tag{2.8}$$

$$\sum_{v \in V} t_i^v = \sum_{v \in V} t_j^v, \quad \forall (i, j) \in S, \tag{2.9}$$

$$t_i^v \ge 0, \quad \forall i \in N, \quad \forall v \in V. \tag{2.10}$$

Constraints (2.6) state that each node should be serviced once (remember that in this case N contains duplicates of the nodes that require more than one vehicle). Constraints (2.7) ensure time consistency: if node j is visited after node i by vehicle v, then $t_i^v + s_i + T_{ij} \le t_j^v$. Constraints (2.8) force t_i^v to be equal to 0 if vehicle v does not visit node i, otherwise, the time window should be respected. Constraints (2.9) guarantee the synchronization of visits.

The main drawback of this formulation, which we will refer as *SVRP1*, is the duplication of nodes when the synchronization occurs on the same node (which is the context of this study). This leads to an increment in the number of variables and constraints, which generate a more complex problem to solve.

2.2. SVRP2

Li et al. [13] and Dohn et al. [14] propose the use of the scheduling variable t_i for the starting time of the service at node i, since the value of these variables is the same for all vehicles that provide service to this node. In this case, it is not necessary to duplicate in N the nodes where the synchronization occurs. These authors use the following formulation for the demand and scheduling constraints:

$$\sum_{v \in V} \sum_{i \in N} x_{ij}^v = n_j, \quad \forall j \in N \backslash \{o, d\}, \tag{2.11}$$

$$t_i + s_i + T_{ij} \le t_j + M(1 - x_{ij}^v), \quad \forall (i, j) \in A, \quad \forall v \in V, \tag{2.12}$$

$$a_i \le t_i \le b_i, \quad \forall i \in N \backslash \{o, d\}, \tag{2.13}$$

$$t_i \ge 0, \quad \forall i \in N, \quad \forall v \in V. \tag{2.14}$$

Constraints (2.11) impose that each node j is served by n_j vehicles, $j \in N \backslash \{o, d\}$. Constraints (2.12) ensure the time consistency, where M is a big number. Constraints (2.13) are related to the time windows. The model defined by expressions (2.1)–(2.5) and (2.11)–(2.14) will be referred as *SVRP2*.

One can observe that this model leads to fewer variables and to fewer constraints than *SVRP1*. However, the efficiency of this model is linked to the chosen value for M. An overestimation of M generally implies a bad linear relaxation solution and high computational effort.

Note that Crainic et al. [12] propose to discretize the planning horizon in order to formulate this problem. However, this implies duplication of the nodes (one for each period) and generally results in an impractical formulation. Furthermore, this leads to problematic related to the approximation, since the finer the periods are, the better is the solution precision, but the more complex the formulation is. For these reasons, we did not consider this approach in this comparison.

3. NEW FORMULATIONS FOR THE SVRP

SVRP1 and *SVRP2* contain both some advantages and some drawbacks. In this section, we present three new formulations for the *SVRP*. The main idea of these formulations is to use the scheduling variable of *SVRP2*, where no duplication of nodes is required, and to model the scheduling constraints as in *SVRP1*, where no big M is necessary.

3.1. SVRP3 and SVRP4

The first proposed formulation is based on *SVRP2*. We replace the constraints (2.12) by the following:

$$t_i + (s_i + T_{ij})x_{ij}^v \le t_j + (b_i - a_j)(1 - x_{ij}^v), \quad \forall (i,j) \in A, \quad \forall v \in V. \tag{3.1}$$

Constraints (3.1) ensure that when vehicle v visits node j after node i, $t_i + s_i + T_{ij} \le t_j$, otherwise $t_i - b_i \le t_j - a_j$, which is always true since $t_i - a_j \ge 0$ and $t_j - b_j \le 0$. Note that the resulting model contains less constraints and variables than *SVRP1* and does not require to calibrate the *M* value as in *SVRP2*. We will refer to this model as *SVRP3*.

For the next formulation, we introduce the following binary variables:

$$z_{ij} = \begin{cases} 1, & \text{if arc}(i,j) \in A \text{ is traversed by any vehicle,} \\ 0, & \text{otherwise.} \end{cases}$$

Hence, constraints (3.1) in the *SVRP3* can be replaced by:

$$x_{ij}^v \le z_{ij}, \quad \forall (i,j) \in A, \quad \forall v \in V, \tag{3.2}$$

$$t_i + (s_i + T_{ij})z_{ij} \le t_j + (b_i - a_j)(1 - z_{ij}), \quad \forall (i,j) \in A. \tag{3.3}$$

The resulting formulation will be referred as *SVRP4*. It leads to more variables and constraints than *SVRP3*, however the experiments carried out show that adding these variables tends to reduce the computational time for solving the problem. Furthermore, since the time consistency is formulated with the variables z_{ij}, $(i,j) \in A$, it is no more necessary to have a distinct variable x for each vehicle when the fleet is homogeneous. From this observation, we derive the next formulation.

3.2. SVRP5

The third proposed formulation is based on the following flow variables:

x_{ij} : Represents the number of vehicles traversing $\text{arc}(i,j) \in A$.

Hence, we consider the following flow formulation:

$$\min \sum \sum_{(i,j)\in A} T_{ij} x_{ij}, \tag{3.4}$$

$$\sum_{(o,j)\in A} x_{ij} = |V|, \tag{3.5}$$

$$\sum_{(j,d)\in A} x_{jd} = |V|, \tag{3.6}$$

$$\sum_{i\in N\backslash\{d\}} x_{ij} = n_j, \quad \forall j \in N\backslash\{o,d\}, \tag{3.7}$$

$$\sum_{i\in N\backslash\{d\}} x_{ij} - \sum_{k\in N\backslash\{o\}} x_{jk} = 0, \quad \forall j \in N\backslash\{o,d\}, \tag{3.8}$$

$$z_{ij} \le x_{ij} \le \min\{n_i, n_j\} z_{ij}, \quad \forall (i,j) \in A, \tag{3.9}$$

$$t_i + (s_i + T_{ij})z_{ij} \le t_j + (b_i - a_j)(1 - z_{ij}), \quad \forall (i,j) \in A, \tag{3.10}$$

$$a_i \le t_i \le b_i, \quad \forall i \in N, \tag{3.11}$$

$$x_{ij} \ge 0, \quad \forall (i,j) \in A, \tag{3.12}$$

$$z_{ij} \in \{0,1\}, \quad \forall (i,j) \in A, \tag{3.13}$$

$$t_i \ge 0, \quad \forall i \in N. \tag{3.14}$$

The constraints (3.5) and (3.6) guarantee that all the vehicles leave the depot and return to the depot, respectively. $|V|$ represents the cardinality of V, i.e. the number of available vehicles. Constraints (3.7) are related to the demand of vehicles that must visit each node. Flow conservation is ensured by constraints (3.8). Constraints (3.9) limit the number of vehicles traversing each arc. Constraints (3.10)

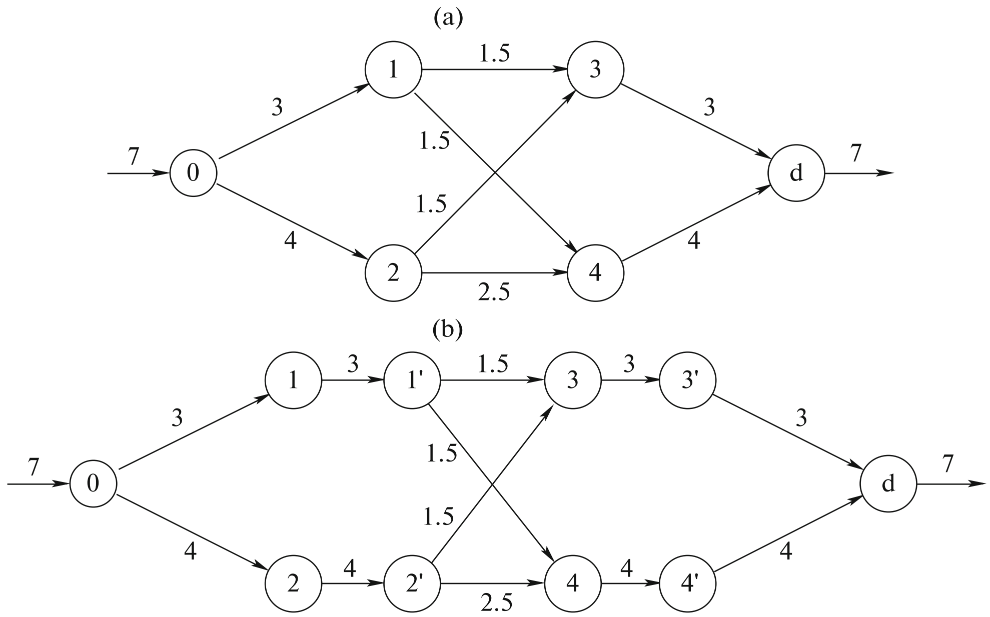

Fig. 1. Graph $\hat{G}$ (a) and the modified graph G^* (b).

and (3.11) impose time consistency and guarantee the satisfaction of time windows, respectively. Finally, constraints (3.12), (3.13), and (3.14) indicate the nature of the variables.

Note that, in *SVRP5*, we do not impose the variables x_{ij}, $(i, j) \in A$, to be integers. The reason for this is detailed in what follows. Let z_{ij}^* and t_i^*, $(i, j) \in A$, be the value of z_{ij} and t_i, respectively, in the optimal solution of *SVRP5*. We define $\hat{G} = (N, \hat{A})$ as the subgraph of *G*, where $\hat{A} = \{(i, j) \in A \,|\, z_{ij}^* = 1\}$.

Proposition 1. $\hat{G}$ is a weakly connected directed acyclic graph.

Proof. $\hat{G}$ is obviously acyclic since any cycle would not satisfy the time consistency imposed by constraints (3.10). Furthermore, since all nodes should be visited by at least one vehicle (constraints (3.7)), the flow conservation (3.8) and constraints (3.9) ensure that there is at least one path in $\hat{G}$ that connects any node in N to the initial and final depots. Then, there is at least one path (passing through one of the depots) connecting any pair of nodes in the equivalent undirected graph of $\hat{G}$.

Proposition 2. It exists an optimal solution of the *SVRP5* where the variables $x_{ij}, (i, j) \in A$ are integers.

Proof. We use the same notation introduced previously and we suppose that z_{ij} and t_i^*, $(i, j) \in A$, are fixed to their optimal value z_{ij}^* and t_i^*, respectively. Because of the constraints (3.10), any feasible flow on $\hat{G}$, which is acyclic, satisfies the starting time t_i^*, $i \in N$, for the service. Hence, the objective is to find a flow that minimizes the total traveling time on $\hat{G}$. We show that this problem is equivalent to a minimum cost flow problem.

For any node $i \in N \backslash \{o, d\}$, we create a virtual node i^*, and $\hat{G}$ is modified as follows:

—a directed arc (i, i^*) is inserted between each node i and its virtual node i^*;

—all directed arcs $(i, j) \in A$ are removed and replaced by a directed arc (i^*, j).

The resulting graph is denoted $G^* = (V, \hat{A})$ (see Fig. 1).

In order to satisfy the service demand, the flow traversing from i to i^* is fixed to n_i $(n_i \le x_{ij} \le n_i)$ and its associated cost is set to 0. The cost of any arc $(i^*, j) \in \hat{A}$ is equal to T_{ij} and $z_{ij}^* \le x_{i^*j} \le \min\{n_i, n_j\} z_{ij}^*$. Hence, finding the minimum cost flow on G^* is equivalent to finding the minimum cost flow on $\hat{G}$ that

Table 1. Number of variables and constraints of the formulations

SVRP	Variables			
	binary	continuous	routing	scheduling
SVRP1	$\lvert N^*\rvert\,(\lvert N^*\rvert - 1)\,\lvert V\rvert$	$\lvert N^*\rvert\,\lvert V\rvert$	$\lvert N^*\rvert\,(\lvert V\rvert + 1) - 2$	$\lvert N^*\rvert(\lvert N^*\rvert - 1)\lvert V\rvert + (\lvert N^*\rvert - 2)\lvert V\rvert + \lvert S\rvert$
SVRP2	$\lvert N\rvert\,(\lvert N\rvert - 1)\,\lvert V\rvert$	$\lvert N\rvert$	$\lvert N\rvert\,(\lvert V\rvert + 1) - 2$	$\lvert N\rvert(\lvert N\rvert - 1)\lvert V\rvert + (\lvert N\rvert - 2)\lvert V\rvert$
SVRP3	$\lvert N\rvert\,(\lvert N\rvert - 1)\,\lvert V\rvert$	$\lvert N\rvert$	$\lvert N\rvert\,(\lvert V\rvert + 1) - 2$	$\lvert N\rvert(\lvert N\rvert - 1)\lvert V\rvert + (\lvert N\rvert - 2)\lvert V\rvert$
SVRP4	$\lvert N\rvert\,(\lvert N\rvert - 1)$	$\lvert N\rvert((\lvert N\rvert - 1)\lvert V\rvert + 1)$	$\lvert N\rvert\,(\lvert V\rvert + 1) - 2$	$\lvert N\rvert(\lvert N\rvert - 1)(\lvert V\rvert + 1) + (\lvert N\rvert - 2)\lvert V\rvert$
SVRP5	$\lvert N\rvert\,(\lvert N\rvert - 1)$	$\lvert N\rvert^2$	$2\,(\lvert N\rvert - 1)$	$\lvert N\rvert\,(\lvert N\rvert - 1)$

With $\lvert N^*\rvert = \lvert N\rvert + \sum_{i=1}^{n}(n_i - 1), \quad \lvert S\rvert = \sum_{i=1}^{n} n_i(n_i - 1)$

satisfies the service demand of each node. The flow traversing each arc $(i^*, j)' \in \hat{A}$ in $\hat{G}$ is indeed equivalent to the flow that traverses arc $(i, j) \in \hat{A}$ in G^*, as shown in Fig. 1.

It is well known that the minimum cost flow problem admits an optimal integer solution when the bounds on the flow are integers, which is the case here. That is to say, if z_{ij} and $(i, j) \in A$, are fixed to their optimal value, it is always possible to find an equivalent optimal solution where the variables x_{ij} are integers. Hence, it exists an optimal solution of the *SVRP5* where the variables x_{ij}, $(i, j) \in A$, are integers.

A direct consequence of this proposition is that the integrality of the flow variables x_{ij}^{v}, $(i, j) \in A$, $v \in V$, in the formulation *SVRP4* can also be relaxed. However, when an integral solution for *SVRP5* can be found directly with a primal or dual simplex algorithm, in the case of *SVRP4* it is necessary to solve the corresponding sub-problem (where z_{ij} and t_i, $(i, j) \in A$, are fixed to their optimal value) with a discrete optimization algorithm. In our tests, CPLEX always returned an integer solution for *SVRP5* and a fractional one for *SVRP4*. Solving in a second step the sub-problem generated by the relaxed version of *SVRP4* takes less than one second and it is more efficient than solving directly *SVRP4*. For this reason, we only present the results for *SVRP4* with its relaxed version, including the computation time required to solve the sub-problem.

In Table 1, the number of variables and the number of constraints of each model are given. Note that the number of routing constraints includes the demand constraints, and the number of nodes in *SVRP1* is higher than in the other models due to the duplication of the nodes with synchronization. Moreover, the number of variables of *SVRP4* corresponds to its relaxed version. From this table, we can see that *SVRP5* has less binary variables and constraints than the other formulations.

4. EXPERIMENTAL RESULTS

The described formulations have been tested and compared with the benchmark proposed by Bredström and Rönnqvist [18] for the HHC problem. These instances are composed of five small instances and five realistic sized instances (see Table 2). Four groups of time windows are considered: Small (S), Medium (M), Large (L), and no time windows (A), in an incremental way, i.e. each time window is contained in the previous time window. The vehicles are available throughout the planning horizon, fixed to 9 hours, and resting time for the drivers is not considered. For the instances (A) with no time window, the whole planning horizon is used as the time window at each node.

We implemented the optimization models in *C++* and solved them with CPLEX 12.6 on a workstation HP Z420 with a processor Intel Xeon E5-1620v2 3.7 10M 1866 4C. We set a time limit of one hour for CPLEX, with one thread, and report the best solution and the final gap (defined as 100 × |*bestnode-bestinteger*|/(1*e* − 10 + |*bestinteger*|)). The results obtained are displayed in Tables 3, 4, and 5 where GAP and CPU refer to the final gap (%) returned by CPLEX and the total processing time (s.), respectively. When the gap is missing, CPLEX was not able to find a feasible solution within the time limit. For each instance, the group of time windows is in parenthesis.

We note that *SVRP4* and *SVRP5* are the only formulations that solved to optimality all instances with 20 nodes within the time limit, and most of the instances with 50 nodes. The models *SVRP1*, *SVRP2*, and *SVRP3* struggle to find at least one feasible solution for instances with more than 50 nodes.

Table 2. Characteristics of the instances from Bredström and Rönnqvist [4]

Instance	$\|N\|$	$\|V\|$	$\|S\|$
1	20	4	2
2	20	4	2
3	20	4	2
4	20	4	2
5	20	4	2
6	50	10	5
7	50	10	5
8	50	10	5
9	80	16	8
10	80	16	8

Table 3. CPLEX Gap and processing time for the instances of 20 nodes

Instance	SVRP1		SVRP2		SVRP3		SVRP4		SVRP5	
	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s
1(S)	0.00	1.566	0.00	0.418	0.00	0.361	0.00	0.022	0.00	0.013
1(M)	0.00	3.447	0.00	3.509	0.00	3.276	0.00	0.037	0.00	0.024
1(L)	0.00	51.343	0.00	0.790	0.00	0.910	0.00	0.047	0.00	0.019
1(A)	19.27	>3600	17.4	>3600	10.73	>3600	0.00	5.247	0.00	3.014
2(S)	0.00	0.111	0.00	0.029	0.00	0.035	0.00	0.017	0.00	0.009
2(M)	0.00	1.320	0.00	0.342	0.00	0.537	0.00	0.044	0.00	0.026
2(L)	0.00	157.935	0.00	9.463	0.00	4.305	0.00	0.109	0.00	0.050
2(A)	0.00	445.032	0.00	57.342	0.00	84.030	0.00	0.571	0.00	0.140
3(S)	0.00	2.032	0.00	0.541	0.00	0.433	0.00	0.030	0.00	0.014
3(M)	0.00	10.132	0.00	0.648	0.00	1.492	0.00	0.039	0.00	0.025
3(L)	0.00	23.165	0.00	9.049	0.00	1.220	0.00	0.085	0.00	0.055
3(A)	0.00	642.489	0.00	312.455	0.00	539.679	0.00	2.528	0.00	0.694
4(S)	0.00	30.222	0.00	7.640	0.00	4.794	0.00	0.039	0.00	0.024
4(M)	0.00	45.188	0.00	18.266	0.00	12.946	0.00	0.173	0.00	0.057
4(L)	16.89	3600.040	0.00	972.477	0.00	1050.450	0.00	2.004	0.00	1.598
4(A)	25.38	>3600	18.12	>3600	17.76	>3600	0.00	36.133	0.00	9.574
5(S)	0.00	2.239	0.00	0.211	0.00	0.121	0.00	0.027	0.00	0.028
5(M)	0.00	0.558	0.00	0.744	0.00	0.420	0.00	0.031	0.00	0.015
5(L)	0.00	23.017	0.00	0.593	0.00	0.250	0.00	0.047	0.00	0.048
5(A)	16.68	>3600	9.48	>3600	12.15	>3600	0.00	13.089	0.00	4.396

We can also remark that the behavior of *SVRP3* is quite similar to *SVRP2*. Although *SVRP3* avoids the use of a big M, this does not necessary lead to a more robust formulation. However, the introduction of the variables z_{ij}, $(i, j) \in A$, really helps CPLEX to find feasible solutions. Table 6 shows the percentage of feasible solutions found for each class of instances.

The *SVRP4* and *SVRP5* have a very similar behavior. From these experiments, we cannot say which formulation is better. However, Bredström and Rönnqvist [18] with their Branch and Price did not prove optimality of the instances 6, 7 and 8 with Small (S), Medium (M), and Large (L) time windows while *SVRP4* and *SVRP5* only fail with the instance 8(L). Besides no feasible solution for instance 10(S) has been found by their algorithm. Hence, these two formulations seem competitive with the exact approach proposed by Bredström and Rönnqvist [18].

Table 4. CPLEX Gap and processing time for the instances of 50 nodes

Instance	SVRP1		SVRP2		SVRP3		SVRP4		SVRP5	
	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s
6(S)	11.79	>3600	6.20	>3600	7.24	>3600	0.00	0.455	0.00	0.118
6(M)	17.79	>3600	15.51	>3600	13.04	>3600	0.00	18.045	0.00	3.059
6(L)	19.33	>3600	19.89	>3600	17.53	>3600	0.00	607.000	0.00	210.358
6(A)	–	>3600	19.83	>3600	25.61	>3600	9.74	>3600	8.98	>3600
7(S)	18.34	>3600	11.71	>3600	6.93	>3600	0.00	3.559	0.00	2.326
7(M)	–	>3600	21.21	>3600	25.62	>3600	0.00	188.089	0.00	44.382
7(L)	–	>3600	–	>3600	–	>3600	0.00	1753.660	0.00	842.265
7(A)	–	>3600	–	>3600	–	>3600	18.22	>3600	16.97	>3600
8(S)	–	>3600	18.18	>3600	10.73	>3600	0.00	14.744	0.00	26.716
8(M)	–	>3600	–	>3600	–	>3600	0.00	2873.040	0.00	242.437
8(L)	–	>3600	–	>3600	–	>3600	4.17	>3600	2.30	>3600
8(A)	–	>3600	–	>3600	–	>3600	10.14	>3600	11.06	>3600

Table 5. CPLEX Gap and processing time for the instances of 80 nodes

Instance	SVRP1		SVRP2		SVRP3		SVRP4		SVRP5	
	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s	GAP, %	CPU, s
9(S)	–	>3600	24.88	>3600	22.41	>3600	2.49	>3600	1.41	>3600
9(M)	–	>3600	–	>3600	–	>3600	5.92	>3600	5.78	>3600
9(L)	–	>3600	–	>3600	–	>3600	10.93	>3600	9.80	>3600
9(A)	–	>3600	–	>3600	–	>3600	20.16	>3600	21.07	>3600
10(S)	–	>3600	21.68	>3600	18.77	>3600	0.00	1910.190	0.00	147.560
10(M)	–	>3600	–	>3600	–	>3600	3.31	>3600	4.84	>3600
10(L)	–	>3600	–	>3600	–	>3600	9.86	>3600	8.76	>3600
10(A)	–	>3600	54.09	>3600	61.07	>3600	18.33	>3600	19.39	>3600

Table 6. Percentage of feasible solutions found by instance size

Size	MIP1	MIP2	MIP3	MIP4	MIP5
	%				
20	100.00	100.00	100.00	100.00	100.00
50	33.33	58.33	58.33	100.00	100.00
80	0.00	37.50	25.00	100.00	100.00

CONCLUSIONS

In this paper, we propose three new formulations for the vehicle routing problem with exact synchronization: *SVRP3*, *SVRP4,* and *SVRP5.* These three formulations are compared against two other formulations proposed in the literature (*SVRP1* and *SVRP2*). The computational tests, made with a state of the art commercial solver, show that *SVRP4* and *SVRP5* leads to a more scalable formulation since it allows to solve larger instances than the other ones.

The proposed comparative tests allow identifying clearly the most efficient formulations for the SVRP. In particular, it seems that separating the flow variables from the scheduling variables through the use of intermediate binary variables (the variables z), leads to more robust formulations. Indeed, we show that using such formulations permits to relax the integrality of the flow variables, which result to an improved convergence towards an optimal solution.

However, even with the best proposed formulation, CPLEX was not able to solve to optimality most instances with more than 80 nodes within the time limit of one hour. Base on the formulation *SVRP5*, it would be interesting to investigate more efficient exact methods, such as a dedicated Branch-and-Bound or a Branch-and-Price. The challenge is to identify the optimal directed acyclic graph $\hat{G}$ in G since the flow can be directly deduced from this graph.

We also consider expanding our formulations to other class of vehicle routing problems with synchronization. In particular, when synchronization occurs on separate nodes or between heterogeneous vehicles. Applications to the snow-plowing problem and to the road marking problem are also part of our future work.

ACKNOWLEDGMENTS

The authors are very grateful to CONACYT for its support through a full time scholarship. The authors also thank David Bredström and Mikael Rönnqvist for sharing their instances.

REFERENCES

1. G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," Manage. Sci. **6**, No. 1, 80–91 (1959).
2. B. L. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer Science, New York, 2008).
3. P. Toth and D. Vigo, *The Vehicle Routing Problem* (Soc. Industrial and Applied Mathematics, Philadelphia, USA, 2001).
4. G. Laporte, "The vehicle routing problem: an overview of exact and approximate algorithms," Eur. J. Operat. Res. **59**, 345–358 (1992).
5. M. Drexl, "Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints," Transport. Sci. **46**, 297–316 (2012).
6. A. Amaya, A. Langevin, and M. Trépanier, "The capacitated arc routing problem with refill points," Operat. Res. Lett. **35**, 45–53 (2007).
7. M. A. Salazar-Aguilar, A. Langevin, and G. Laporte, "The synchronized arc and node routing problem: application to road marking," Comput. Operat. Res. **40**, 1708–1715 (2013).
8. S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," Transport. Sci. **40**, 455–472 (2006).
9. M. A. Salazar-Aguilar, A. Langevin, and G. Laporte, "Synchronized arc routing for snow plowing operations," Comput. Operat. Res. **39**, 1432–1440 (2012).
10. I. Ioachim, J. Desrosiers, F. Soumis, and N. Bélanger, "Fleet assignment and routing with schedule synchronization constraints," Eur. J. Operat. Res. **119**, 75–90 (1999).
11. N. Bélanger, G. Desaulniers, F. Soumis, and J. Desrosiers, "Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues," Eur. J. Operat. Res. **175**, 1754–1766 (2006).
12. T. G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems," Transport. Sci. **43**, 432–454 (2009).
13. Y. Li, A. Lim, and B. Rodrigues, "Manpower allocation with time windows and job-teaming constraints," Naval Res. Logistics **52**, 302–311 (2005).
14. A. Dohn, E. Kolind, and J. Clausen, "The manpower allocation problem with time windows and job teaming constraints: a branch-and-price approach," Comput. Operat. Res. **36**, 1145–1157 (2009).
15. B.-I. Kim, J. Koo, and J. Park, "The combined manpower-vehicle routing problem for multi-staged services," Expert Syst. Appl. **37**, 8424–8431 (2010).
16. R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm Intell. **1**, 33–57 (2007).
17. P. Eveborn, P. Flisberg, and M. Rönnqvist, "Laps care—an operational system for staff planning of home care," Eur. J. Operat. Res. **171**, 962–976 (2006).
18. D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," Eur. J. Operat. Res. **191**, 19–31 (2008).
19. D. Bredström and M. Rönnqvist, "A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints," Tech. Rep. (Dep. Finance and Management Sci., Norwegian School of Economics and Business Administration, 2007).
20. N. Labadie, C. Prins, and Y. Yang, "Iterated local search for a vehicle routing problem with synchronization constraints," in *Proceedings of the ICORES 2014* (Angers, France, 2015), pp. 257–263.
21. B. D. Rosa, G. Improta, G. Ghiani, and R. Musmanno, "The arc routing and scheduling problem with transshipment," Transport. Sci. **36**, 301–313 (2002).
22. A. del Pia and C. Filippi, "A variable neighborhood descent algorithm for a real waste collection problem with mobile depots," Int. Trans. Operat. Res. **13**, 125–141 (2006).