STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, November 2025, pp 2724–2743.
Published online in International Academic Press (www.IAPress.org)

Vehicle Routing Problem with Synchronization and Scheduling Constraints of support vehicles

Adil Tahir [1,2,*], Mohamed El Fassi [2], Younes Oujamaa [1], Mohamed Ait Lahcen [1]

[1] *Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco*
[2] *GERAD, Montréal, Québec, Canada*

Abstract Many transportation planning processes in real-world applications are complex and require strong cooperation among various vehicles. When using expensive vehicles, their utilization plays a decisive role in an efficient supply chain. In mining production or civil construction processes, such as mining unloading or road building, the machines are typically mobile, and synchronization between different types of vehicles ensures better use of vehicle fleets, reduces traveled distances, non-productive times, and logistics costs. In this paper, we consider two types of vehicles, called primary and support vehicles. Primary vehicles perform operations and are assisted by at least one support vehicle, with support vehicles scheduled according to a First-Come, First-Served (FCFS) policy. We refer to this practical problem as the vehicle routing problem with synchronization and scheduling constraints of support vehicles. To tackle this problem, we introduce three mixed-integer linear programming models. The first approach involves vehicle routing with synchronization only, breaking each task into several subtasks by duplicating nodes in the graph representation, which produces an equivalent network flow problem. The second model addresses subtasks by adding constraints that determine the assignment of each subtask to a specific primary and support vehicles. The third model incorporates an additional FCFS scheduling constraint for support vehicles. Computational results on 100 real-world instances show that the second model reduces the first model's computational time by 30%. In contrast, the results of the third model indicate that the FCFS constraint for support vehicles has little impact on solution quality and slightly increases computation time, demonstrating the robustness and practical applicability of the scheduling approach.

Keywords Vehicle Routing, Scheduling, Synchronization, Mixed-integer programming

AMS 2010 subject classifications 90C27, 90B06

DOI: 10.19139/soic-2310-5070-2916

1. Introduction

This research is motivated by a problem commonly encountered in mining operations, specifically in vehicle scheduling and routing. It focuses on the pairwise synchronization of operations (e.g., mineral hauling) between two different types of vehicles, both in space and time. The first type of vehicle, called a primary vehicle, is a mining loader that performs loading operations. Since loaders generally cannot transport material (e.g., ore or waste) over long distances, mining trucks, called support vehicles, are required to collect the material during operations. Once fully loaded, these vehicles transport the material to the processing plant, storage facilities, or the waste dump, then return to the loader after unloading. Together, primary and support vehicles must accomplish a set of tasks, each involving the transfer of a specified quantity of ore or waste between a given site and its designated destination. Given the vehicles' capacities and the quantities to be transported, each task is naturally

*Correspondence to: Adil Tahir (Email: tahir@emi.ac.ma). Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

divided into subtasks, with each subtask representing a fraction of the total task to be completed. All subtasks must be completed within a defined time horizon, while minimizing total completion time and balancing the workload across vehicles. The underlying optimization problem can be modeled as a variant of the Vehicle Routing Problem (VRP) incorporating synchronization constraints and support vehicles scheduling constraints, following the first-come, first-served policy. We refer to this new problem as the Vehicle Routing Problem with Synchronization and Scheduling Constraints of Support Vehicles.

The literature on routing problems with synchronization constraints was initially addressed by Drexl [1], where interdependencies arise from routing decisions involving two or more distinct vehicles. Unlike classical Vehicle Routing Problems (VRPs), the VRP with Multiple Synchronization Constraints (VRPMS) involves interdependent routes. A modification in one route may affect others and, in some cases, make them infeasible. To deal with this problem, five different types of synchronization were proposed by Drexl [1]. *Task synchronization* is generally included in any VRP, as delivery tasks or services must be performed exactly once by one or more suitable vehicle(s). *Operation synchronization* concerns the spatial and temporal coordination of tasks, referring to the time offsets required between operations of two or more vehicles at the same or different locations. This concept is relevant to multi-echelon VRPs (Perboli et al. [5]) and to combined VRP-scheduling problems with time windows and additional temporal constraints (Bredström and Rönnqvist [14]). In addition, operation synchronization can generate dynamic time windows, in which the execution of a task depends on the completion of another task. From a temporal viewpoint, three types of operation synchronization are distinguished: pure spatial synchronization, synchronization with precedence constraints, and exact synchronization (e.g., Mankowska et al. [15]). *Resource synchronization* that vehicles compete for common, limited resources (Hempsch and Irnich [16]). Grimault et al. [17] investigate a pickup and delivery problem with resource synchronization; a set of truck routes is needed to serve a set of demands sharing a set of resources; the authors introduce a specific synchronization-based destroy operator for their Adaptive Large Neighborhood Search (ALNS). *Movement Synchronization* occurs when two vehicles must be synchronized in space and time to form a single composite vehicle. There are two types of movement synchronization: movement at the depot and en route. Meisel and Kopfer [4] address a VRP with en route movement synchronization. This problem involves pickup-and-delivery requests that require coordinated operations between active and passive vehicles, which can be combined at each customer location. This problem is solved by the exact branch-and-price procedure proposed by Tilk et al. [8]. *Load synchronization* means that vehicles exchange cargo along their routes or deliver it to the same customer. This occurs in routing problems with split delivery, see e.g., Desaulniers [7], where a customer is served by multiple vehicles, each providing a fraction of the total customer demand. There are three types of load synchronization: fixed, discretized, and continuous. For further details in line with the discussion presented here, Soares et al. [18] clarified the concept of synchronization and proposed a simple mathematical formulation for VRPs with synchronization, in which the different types of synchronization are explicitly modeled and mathematically defined.

Based on this classification, the concept of synchronization is used broadly. Salazar-Aguilar et al. [3] introduce a synchronized arc routing problem inspired by snow plowing operations. In this problem, synchronization occurs along the arcs; the routes must be designed so that street segments with two or more lanes in the same direction are plowed simultaneously by different synchronized vehicles. A non-linear mixed integer formulation and an ALNS procedure are proposed. Quttineh et al. [9] model the military aircraft mission planning problem as a VRP with synchronization constraints, where the target illumination by one aircraft and the attack by another aircraft require exact operation synchronization. Small instances of the problem are solved optimally with a commercial solver CPLEX. Rix et al. [12] present a tactical wood flow model with multiple periods in the context of the Canadian forestry industry, which takes into account synchronization with log loaders. This problem is formulated as a mixed-integer linear program and solved using column generation. Soares et al. [13] also study a full truck-load pickup and delivery problem with multi-vehicle synchronization. Their approach integrates truck-and-loader synchronization to reduce unproductive time, using a fix-and-optimize matheuristic that outperforms traditional MIP solvers on constrained instances. Fink et al. [10] define a VRP with workers and vehicle synchronization. Workers use vehicles to move from one location, such as a gate or a depot, to another. Hence, both workers and vehicles are passive units on their own. The authors solve the problem using a column-generation-based heuristic. Hof and Schneider [11] introduce the Vehicle Routing Problem with Time Windows and Mobile Depots

(VRPTWMD). This problem involves fleets of task vehicles and support vehicles, where the latter act as mobile depots to restore either the load or the fuel capacity of task vehicles used to satisfy customer demands. The authors propose a hybrid heuristic composed of an adaptive large neighborhood search and a path relinking approach, which shows compelling performance on larger instances of the related two-echelon multiple-trip VRP with satellite synchronization, viewed as a special case of the VRPTWMD.

Recently, Soares et al. [19] proposed a robust optimization approach for the Vehicle Routing Problem with Synchronization, taking into consideration the uncertainty in vehicle travel times between customers. This work builds on existing approaches in the literature to develop mathematical models for the Robust VRPMP and a branch-and-cut algorithm to solve more difficult problem instances.

Despite extensive research on VRPs with synchronization constraints and on scheduling problems, to the best of our knowledge, no study has addressed both aspects simultaneously. In this paper, we focus on the mixed-integer linear formulation of exact operation synchronization, i.e., when the offset is zero and synchronization occurs at the same vertex. In this case, a vehicle may wait at a vertex that the other required vehicle arrives to start the operation. The main contributions of this research are as follows:

- For the first time, we introduce a new variant for the VRPMS with a homogeneous fleet, in which the task is disaggregated into subtasks by adding side constraints. These side constraints guarantee that each subtask can be assigned to a single primary-support vehicle.
- The support vehicles are scheduled according to a First-Come-First-Served (FCFS) policy.
- A real-world case study at a major mining company illustrating the practical implementability of the proposed models.

We organize the remainder of this paper as follows. In Section 2, we give a detailed description of the problem. The new mathematical formulations and a theoretical discussion are presented in Section 3. Section 4 presents the computational experiments and associated discussions. Finally, we summarize the work and provide perspectives for future research in Section 5.

2. Problem statement

Table 1. Business rules.

Rule	Description
R-1	Each task can be decomposed into a minimum set of subtasks according to the capacity of the support vehicles.
R-2	Each task is assigned to a unique destination.
R-3	The execution of a subtask requires synchronization in time and location between a primary vehicle and a support vehicle.
R-4	A support vehicle can be allocated to multiple primary vehicles and is not exclusively assigned to any single one.
R-5	Support vehicles may not arrive at a task simultaneously.
R-6	A primary vehicle can start a task as soon as a support vehicle has arrived.
R-7	A primary vehicle must complete a task before moving on to the next one.

Mineral hauling is a fundamental component of the mining supply chain, as it involves transporting ore and waste (sterile material) from various locations to processing or storage facilities. This process involves a homogeneous fleet of vehicles composed of two types: primary vehicles, tasked with loading operations, and support vehicles, assigned to transporting materiel to a specific destination. Each hauling task corresponds to a specific route and requires the simultaneous presence of a primary vehicle and one or more support vehicles to ensure operational feasibility. Primary and support vehicles are not always available. For each vehicle, the days of availability,

capacities, loading and unloading times (for support vehicles), and travel times between different locations are defined.

In this context, the objective of the Vehicle Routing Problem with Synchronization Constraints (VRPSC) is to efficiently plan the daily routes of primary and support vehicles, specifying for each task the vehicles involved and their order of intervention, while minimizing the total execution time and balancing the workload among the available vehicles. This must be done while respecting all business rules listed in Table 1.

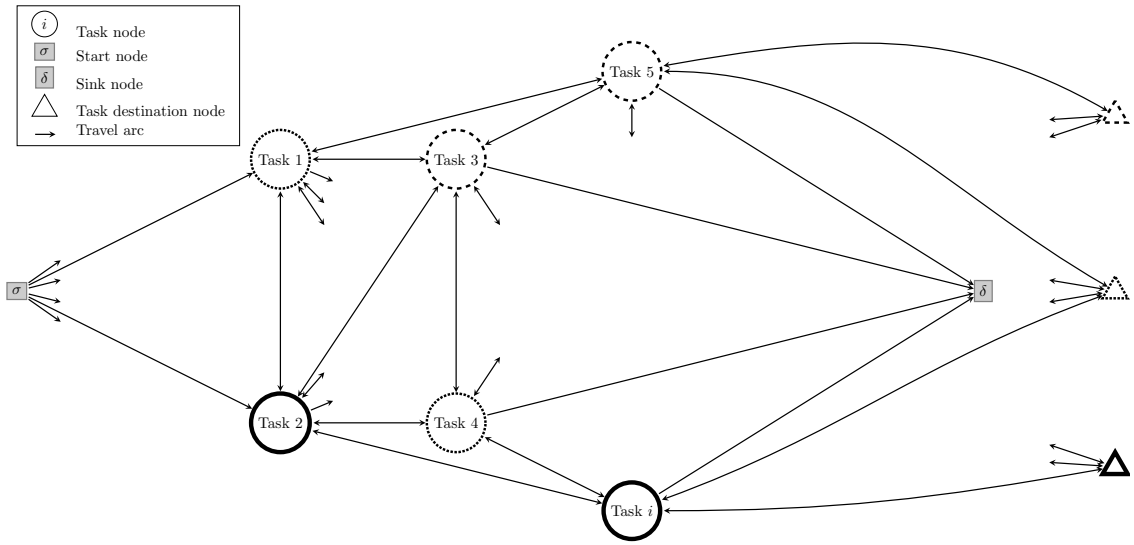

Figure 1. Part of the network illustrating tasks and their associated destinations.

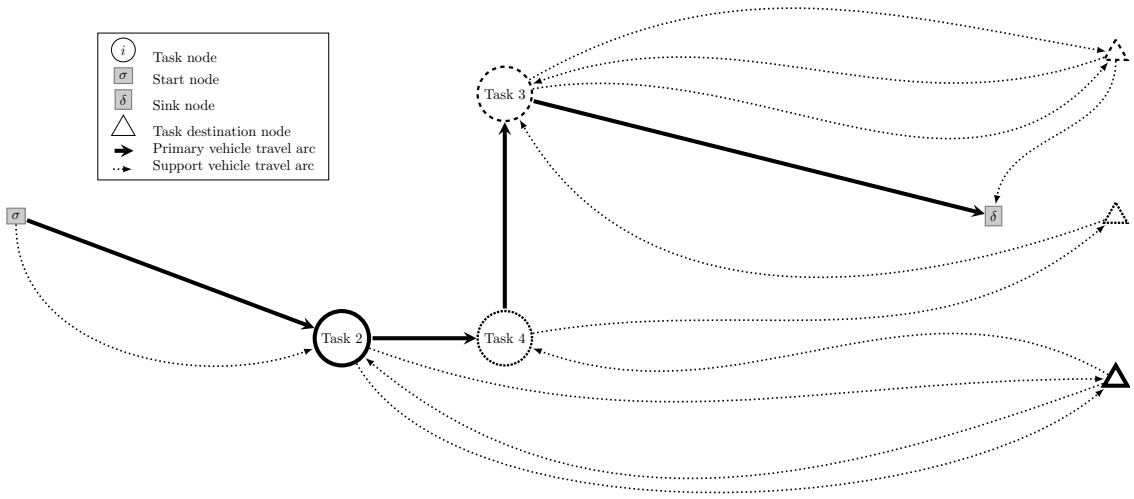

Figure 2. Example of primary and support vehicle routes.

The VRPSC can be formulated on a directed graph $\mathbf{G} = (\mathbf{N}, \mathbf{A})$, where $\mathbf{N}$ denotes the set of nodes and $\mathbf{A}$ the set of arcs corresponding to the movements between nodes. The graph $\mathbf{G}$ is partially shown in Figure 1; for clarity, some arcs are truncated or omitted. In addition to a source node (σ) and a destination node (δ), the graph contains

two types of nodes: circle nodes and triangular nodes. Circles represent tasks performed by vehicles based at the depot, while triangles denote the corresponding unloading destinations. At these locations, materials are unloaded: waste goes to a dump, ore to a processing unit, or either to a storage facility. The functional association between each task and its corresponding destination is illustrated by assigning them a common line style. In this network, the incoming and outgoing arcs of tasks' destination nodes are reserved exclusively for support vehicles, whereas the remaining arcs are shared by both primary and support vehicles.

Figure 2 illustrates an example route involving one primary and one support vehicle for three tasks (circles). The primary vehicle travels between tasks to perform all associated subtasks, while the support vehicle transports materials to their respective destinations.

Since the locations of tasks and their destinations are known in advance, and the travel speeds of each vehicle type are predetermined, the destination nodes are omitted from our models. Instead, we account only for the time required for loading, traveling to the destination, unloading, and returning to the task site.

Table 2. Notation used for modeling the VRPSC variants

Sets and Elements	
$\mathcal{T}$	Set of tasks to be performed.
$\mathcal{H}$	Set of vehicle types: *primary* (p) and *support* (s).
$\mathcal{V}_h$	Set of vehicles of type $h \in \mathcal{H}$ available at the depot.
$\mathcal{V}$	Set of all vehicles, regardless of type.
$\mathbf{R}_i$	Set of subtasks associated with task $i \in \mathcal{T}$.
$\mathbf{N}_\mathcal{T}$	Set of all nodes representing the tasks or subtasks.
$\mathbf{N}$	Set of all nodes in the network.
$\mathbf{A}$	Set of all arcs in the network.
Parameters	
t_{ij}^v	Travel time of vehicle $v \in \mathcal{V}$ between tasks i and j.
d_{ij}^v	Time required for vehicle $v \in \mathcal{V}$ to move from node i to node j.
∂_{ir}^v	Service time of subtask r of task i by vehicle $v \in \mathcal{V}$.
γ_{ij}	Travel time of the support vehicle from task i's destination to task j's site. For each task i, γ_{ii} denotes the travel time from its origin to its destination.
τ	Total time required for a primary vehicle to serve (e.g., load) a support vehicle.
ξ	Service time (e.g., unloading) required at each task destination.
Decision Variables	
s_i^v	Starting time of processing the task associated with node i by vehicle v. (^{1})
y_i^v	Binary variable equal to 1 if vehicle $v \in \mathcal{V}$ is assigned to task i, 0 otherwise. (^{2,3})
z_{nr}^v	Binary variable equal to 1 if vehicle $v \in \mathcal{V}$ executes subtask r of task n, 0 otherwise.(^{2,3})
s_{nr}^v	Start time of subtask r of task n by vehicle v. (^{2,3})
x_{ij}^v	Binary variable equal to 1 if vehicle v travels from node i to node j, 0 otherwise. (^{1,2,3})
w^p	Continuous variable representing the maximum return time to the depot of all primary vehicles used. (^{1,2,3})

Note. (^{1}) Variables used in VRPSC^1; (^{2,3}) variables used in VRPSC^2 and VRPSSC; (^{1,2,3}) variables common to all three models.

3. Mathematical models

In this section, we present three mathematical models developed for the studied problem. The first two models, denoted VRPSC^1 and VRPSC^2, are two variants of the VRPSC. The third model, denoted VRPSSC, extends VRPSC^2 by incorporating scheduling constraints for support vehicles based on a first-come, first-served policy.

Before describing each model in separate subsections, we introduce the notations common to all three formulations, summarized in Table 2.

3.1. VRPSC[1]

In this section, we present the mathematical formulation of the first model, VRPSC[1]. For this model, each task is decomposed into several *subtasks* by duplicating nodes within the network. We start by illustrating the network, highlighting the task decomposition and the arcs used. Next, we define the notations specific to this variant. Finally, we present the complete mathematical formulation of the model.

3.1.1. Graph representation

Let $\mathbf{G} = (\mathbf{N}, \mathbf{A})$ be a directed graph, where $\mathbf{N}$ denotes the set of nodes and $\mathbf{A}$ the set of arcs. The graph $\mathbf{G}$ for VRPSC[1] is partially illustrated in Figure 3. In fact, this graph contains three types of nodes: *start-task*, *subtask*, and *end-task*. *The subtask* nodes are obtained by decomposing the nodes associated with each task into multiple subtasks. The *start-task* and *end-task* nodes are fictitious nodes associated with each *task*: the *start-task* node (denoted $\tilde{\sigma}_i$) represents entry to the task site, while the *end-task* node (denoted $\tilde{\delta}_i$) represents exit from the site. Also, the graph $\mathbf{G}$ contains three types of arcs connecting these different nodes: *travel arcs*, *start-subtask arcs*, and *end-subtask arcs*. *Travel arcs* connect the *end-task* node of one task to the *start-task* nodes of other tasks, as well as the depot to the *start-task* nodes. *Start-subtask arcs* connect a *start-task* node to the *subtask* nodes of the same task. Finally, the arcs connecting the *subtask* nodes of a task to its *end-task* nodes are called *end-subtask arcs*.

It is worth noting that the *start-subtask arcs* and *end-subtask arcs*, which connect the subtasks, are added depending on the type of vehicle, as illustrated in Figure 3. Moreover, the destination nodes of the tasks are not shown in the figure, as they are not explicitly considered in the model.

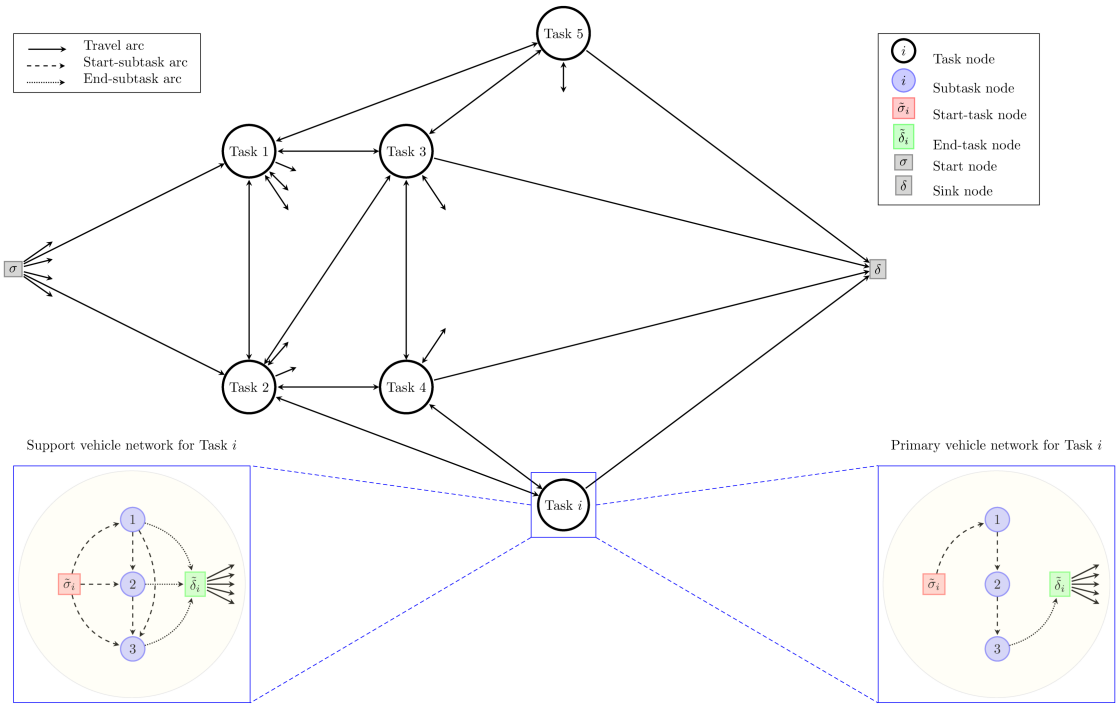

Figure 3. Part of the VRPSC[1] Network.

For the primary vehicle, the *start-subtask arcs* and *end-subtask arcs* are sequential, since the subtasks are identical. Therefore, the execution order is not important, and the primary vehicle can perform them in any order.

Formally, we define $\mathbf{N}$ and $\mathbf{A}$ as follows: $\mathbf{N} = \bigcup_{i\in\mathcal{T}} \left(\mathbf{N}_i \cup \left\{\tilde{\sigma}_i, \tilde{\delta}_i\right\}\right) \cup \{\sigma, \delta\}$, where $\mathbf{N}_i$ is the set of nodes representing the subtasks of task i, and $\mathbf{N}_\mathcal{T} = \bigcup_{i\in\mathcal{T}} \mathbf{N}_i$ is the set of all nodes representing subtasks. And

$$\mathbf{A} = \left\{ \begin{array}{lll} (\sigma, \tilde{\sigma}_i) & \forall i \in \mathbf{N}_\mathcal{T}, & \text{(arcs from the depot to the fictitious } \textit{start-task} \text{ nodes)} \\ \bigcup \ (\tilde{\delta}_i, \delta) & \forall i \in \mathbf{N}_\mathcal{T}, & \text{(arcs from the fictitious } \textit{end-task} \text{ nodes to the depot)} \\ \bigcup \ (\tilde{\sigma}_i, n) & \forall i \in \mathcal{T},\ \forall n \in \mathcal{N}_i, & \text{(arcs from the fictitious } \textit{start-task} \text{ node to the } \textit{subtask} \text{ nodes)} \\ \bigcup \ (j, k) & \forall i \in \mathcal{T},\ \forall j, k \in \mathcal{N}_i | k > j, & \text{(arcs between the } \textit{subtask} \text{ nodes of each task)} \\ \bigcup \ (n, \tilde{\delta}_i) & \forall i \in \mathcal{T},\ \forall n \in \mathcal{N}_i, & \text{(arcs from the } \textit{subtask} \text{ nodes to the fictitious } \textit{end-task} \text{ node)} \\ \bigcup \ (\tilde{\delta}_i, \tilde{\sigma}_j) & \forall i, j \in \mathcal{T},\ i \neq j. & \text{(inter-task transition arcs)} \end{array} \right\}$$

Service and Travel times of each vehicle type are defined as follows:

$$d_{ij}^v = \begin{cases} t_{ij}^v & \text{if} v \in \mathcal{V}_p,\ (i,j) \in \mathbf{A},\ i,j \notin \mathbf{N}_\mathcal{T}, \\ t_{ij}^v & \text{if} v \in \mathcal{V}_s,\ (i,j) \in \mathbf{A},\ i = \sigma, \\ \gamma_{ij} - \gamma_{ii} & \text{if} v \in \mathcal{V}_s,\ (i,j) \in \mathbf{A},\ i,j \notin \mathbf{N}_\mathcal{T},\ i \neq \sigma, \\ 0 & \text{otherwise.} \end{cases} \qquad \partial_{ir}^v = \begin{cases} \tau & \text{if} v \in \mathcal{V}_p,\ i \in \mathcal{T},\ r \in \mathbf{N}_i, \\ \tau + \xi + 2\gamma_{ii} & \text{if} v \in \mathcal{V}_s,\ i \in \mathcal{T},\ r \in \mathbf{N}_i, \\ 0 & \text{otherwise.} \end{cases}$$

It should be noted that $d_{ij}^v = \gamma_{ij} - \gamma_{ii}$ for $v \in \mathcal{V}_s$ $(i,j) \in \mathbf{A}$, with $i,j \notin \mathbf{N}_\mathcal{T}$ and $i \neq \sigma$, since the support vehicle v must move directly to task j with travel time γ_{ij} after completing the last subtask of task i, γ_{ii} is subtracted due to the assumption that the support vehicle returns to the site of task i after each subtask.

*3.1.2. **Mathematical Formulation***

The mathematical formulation of the **VRPSC**[1] is given by the following model:

$$\min_{x,s,w} \quad w^p \tag{1}$$

$$\text{s.t.:} \quad \sum_{(n,j)\in\mathbf{A}} x^v_{nj} - \sum_{(j,n)\in\mathbf{A}} x^v_{jn} = 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N} \setminus \{\sigma, \delta\} \tag{2}$$

$$\sum_{(\sigma,j)\in\mathbf{A}} x^v_{\sigma j} - \sum_{(j,\delta)\in\mathbf{A}} x^v_{j\delta} = 0, \quad \forall v \in \mathcal{V} \tag{3}$$

$$\sum_{(j,n)\in\mathbf{A}} x^v_{jn} \leq 1, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N} \setminus \{\delta\} \tag{4}$$

$$s^v_\delta - w^p \leq 0, \quad \forall v \in \mathcal{V}_p \tag{5}$$

$$\sum_{v\in\mathcal{V}_p} \sum_{(j,n)\in\mathbf{A}} x^v_{jn} = 1, \quad \forall n \in \mathbf{N} \setminus \{\sigma, \delta\} \tag{6}$$

$$\sum_{v\in\mathcal{V}_s} \sum_{(j,n)\in\mathbf{A}} x^v_{jn} = 1, \quad \forall n \in \mathbf{N}_\mathcal{T} \tag{7}$$

$$s^v_j - \left(s^v_i + d^v_{ij} - M_1 \cdot (1 - x^v_{ij})\right) \geq 0, \quad \forall v \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \mid i \notin \mathbf{N}_\mathcal{T} \tag{8}$$

$$s^v_j - \left(s^v_r + \partial^v_{ir} - M_1 \cdot (1 - x^v_{rj})\right) \geq 0, \quad \forall v \in \mathcal{V}, \forall (r,j) \in \mathbf{A} \mid r \in \mathbf{N}_i, i \in \mathcal{T} \tag{9}$$

$$\left|s^v_n - s^{v'}_n\right| - M_2 \cdot \left(2 - \sum_{(j,n)\in\mathbf{A}} \left(x^v_{jn} + x^{v'}_{jn}\right)\right) \leq 0, \quad \forall v \in \mathcal{V}_p,\ \forall v' \in \mathcal{V}_s, \forall n \in \mathbf{N}_\mathcal{T} \tag{10}$$

$$x^v_{ij} \in \{0,1\}, \quad \forall e \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \tag{11}$$

$$s^v_\sigma = 0. \quad s^v_n \geq 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N} \tag{12}$$

$$w^p \geq 0. \tag{13}$$

The objective function (1), combined with the constraint (5), aims to minimize the maximum return time of the primary vehicles to the depot. This formulation also ensures that all tasks are completed in minimal time while balancing the workload among the primary vehicles. Constraints (2) and (3) are flow conservation constraints. Constraint (4) enforces that each vehicle traverses at most one outgoing arc from each node. Constraint (6) ensures that each node, except the depot nodes, is covered by exactly one primary vehicle. Similarly, constraint (7) requires that each node associated with a subtask be covered by a single support vehicle. Constraints (8) and (9) are temporal scheduling constraints: a vehicle can visit node j only after visiting node i (the previous node in its route) and completing the associated subtask, considering both the service time at node i and the travel time from node i to node j. Constraints (10) ensure synchronization between a primary vehicle and a support vehicle when they perform the same subtask. This strict synchronization is only activated if both vehicles are actually assigned to the same node. Finally, constraints (11), (12), and (13) respectively ensure the integrality and non-negativity of the decision variables.

3.2. VRPSC^2

In this section, we introduce the second mathematical model, which addresses a vehicle routing problem with synchronization constraints. Unlike the first model, subtasks are managed through specific constraints and variables rather than node duplication.

Formally, we define $\mathbf{N}$ and $\mathbf{A}$ as follows: $\mathbf{N} = \mathbf{N}_{\mathcal{T}} \cup \{\sigma, \delta\}$, where $\mathbf{N}_{\mathcal{T}} = \{i \mid i \in \mathcal{T}\}$ is the set of nodes representing the tasks.

$$\mathbf{A} = \left\{ \begin{array}{lll} (\sigma, n) & \forall n \in \mathbf{N}_{\mathcal{T}}, & \text{(arcs from the depot to task nodes)} \\ \bigcup (n, \delta) & \forall n \in \mathbf{N}_{\mathcal{T}}, & \text{(arcs from task nodes to the depot)} \\ \bigcup (n_1, n_2) & \forall n_1, n_2 \in \mathbf{N}_{\mathcal{T}},\ n_1 \neq n_2. & \text{(arcs between task nodes)} \end{array} \right\}$$

Service and Travel times of each vehicle type are defined as follows :

$$d_{ij}^v = \begin{cases} t_{ij}^v & \text{if} v \in \mathcal{V}_p \text{ and} (i,j) \in \mathbf{A}, \\ t_{ij}^v & \text{if} v \in \mathcal{V}_s \text{ and} (i,j) \in \mathbf{A}, i = \sigma, \\ \gamma_{ij} - \gamma_{ii} & \text{if} v \in \mathcal{V}_s \text{ and} (i,j) \in \mathbf{A}, i \in \mathbf{N}_{\mathcal{T}}, \\ 0 & \text{otherwise.} \end{cases} \qquad \partial_{ir}^v = \begin{cases} \tau & \text{if} v \in \mathcal{V}_p,\ i \in \mathcal{T},\ r \in \mathbf{R}_i, \\ \tau + \xi + 2\gamma_{ii} & \text{if} v \in \mathcal{V}_s,\ i \in \mathcal{T},\ r \in \mathbf{R}_i, \\ 0 & \text{otherwise.} \end{cases}$$

3.2.1. *Mathematical Formulation*

The mathematical formulation of the VRPSC^2 is presented as follows.

$$\min_{x,y,s,w} \quad w^p$$

$$\text{s.t.:} \quad (2) - (4) \tag{14}$$

$$s_{\delta 1}^v - w^p \leq 0, \quad \forall v \in \mathcal{V}_p \tag{15}$$

$$\sum_{(j,n) \in \mathbf{A}} x_{jn}^v - y_n^v = 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_{\mathcal{T}} \tag{16}$$

$$\sum_{v \in \mathcal{V}_p} y_n^v = 1, \quad \forall n \in \mathbf{N}_{\mathcal{T}} \tag{17}$$

$$z_{nr}^v - y_n^v \leq 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_{\mathcal{T}}, r \in \mathbf{R}_n \tag{18}$$

$$y_n^v - \sum_{r \in \mathbf{R}_n} z_{nr}^v \leq 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_{\mathcal{T}} \tag{19}$$

$$\sum_{v \in \mathcal{V}_p} z_{nr}^v - \sum_{v' \in \mathcal{V}_s} z_{nr}^{v'} = 0, \quad \forall n \in \mathbf{N}_{\mathcal{T}}, r \in \mathbf{R}_n \tag{20}$$

$$\sum_{v \in \mathcal{V}_s} \sum_{r \in \mathbf{R}_n} z_{nr}^v = |\mathbf{R}_n|, \quad \forall n \in \mathbf{N}_{\mathcal{T}} \tag{21}$$

$$s_{n(r+1)}^v - (s_{nr}^v + \partial_{nr}^v z_{nr}^v) \geq 0, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_{\mathcal{T}}, r \in \mathbf{R}_n \tag{22}$$

$$s_{ja}^v - \left(s_i^v + d_{ij}^v - M_1 \cdot (1 - x_{ij}^v)\right) \geq 0, \quad \forall v \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \mid i = \sigma, a = 1 \tag{23}$$

$$s_j^v - \left(s_{ib}^v + d_{ij}^v - M_1 \cdot (1 - x_{ij}^v)\right) \geq 0, \quad \forall v \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \mid j = \delta, b = |\mathbf{R}_i| + 1 \tag{24}$$

$$s_{ja}^v - \left(s_{ib}^v + d_{ij}^v - M_1 \cdot (1 - x_{ij}^v)\right) \geq 0, \quad \forall v \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \mid i, j \notin \{\sigma, \delta\}, b = |\mathbf{R}_i| + 1, a = 1 \tag{25}$$

$$\left|s_{nr}^v - s_{nr}^{v'}\right| - M_2 \cdot \left(2 - z_{nr}^v - z_{nr}^{v'}\right) \leq 0, \quad \forall (v, v') \in \mathcal{V}_p \times \mathcal{V}_s, \forall n \in \mathbf{N}_{\mathcal{T}}, r \in \mathbf{R}_n \tag{26}$$

$$y_n^v, z_{nr}^v \in \{0,1\}, \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_\mathcal{T}, r \in \mathbf{R}_n \tag{27}$$
$$x_{ij}^v \in \{0,1\}, \quad \forall v \in \mathcal{V}, \forall (i,j) \in \mathbf{A} \tag{28}$$
$$s_{nr}^v \geq 0 \quad \forall v \in \mathcal{V}, \forall n \in \mathbf{N}_\mathcal{T}, r \in \mathbf{R}_n \tag{29}$$
$$s_\sigma^v = 0, \quad \forall v \in \mathcal{V}. \tag{30}$$
$$w^p \geq 0. \tag{31}$$

The objective function and the constraint (14) are the same as in the VRPSC^1 model. Constraints (15) calculate the maximum return time of the primary vehicles to the depot. Constraint (16) links the x and y variables: if a vehicle v traverses an arc entering node n, then $y_n^v = 1$, indicating that this vehicle covers a subset of the subtasks associated with task n. Constraint (17) ensures that each node corresponding to a task is covered by exactly one primary vehicle, which executes all subtasks associated with that task. Constraints (18) and (19) enforce that the variables z_{nr}^v are zero whenever $y_n^v = 0$. In other words, if vehicle v does not cover node n, it executes no subtask associated with that node. Conversely, if a vehicle executes no subtask of a given task, it does not cover that task. Constraints (20) and (21) ensure that all subtasks are executed by both a primary and a support vehicle. Constraints (22) schedule the subtasks r of a task n: a vehicle cannot execute two subtasks simultaneously. Constraints (23), (24) and (25) ensure that a vehicle can only visit node j after visiting node i and completing the subtasks associated with i. For a primary vehicle, this means completing all subtasks of i, while a support vehicle must complete at least one. Synchronization is handled using s_{ib}^v, representing the end of the last subtask of i, and s_{ja}^v, representing the start of the first subtask of j, and is activated only if the vehicle executes at least one subtask in both i and j. Constraints (26) ensure synchronization between a primary vehicle and a support vehicle executing the same subtask r. Finally, constraints (27), (28), (29), (30), and (31) enforce the integrality and non-negativity of the decision variables.

3.3. VRPSSC

This section introduces the mathematical model for the vehicle routing problem that includes synchronization and scheduling constraints for support vehicles. It mainly extends the VRPSC^2 model.

$$\min_{x,y,s,w} \quad w^p$$
$$\text{s.t.:} \quad (14) - (30) \tag{32}$$
$$s_{nr}^v - \left(s_{nr}^{v'} + M_2(1 - z_{nr}^v)\right) \leq 0, \quad (v, v') \in \mathcal{V}_s \times \mathcal{V}_s, \forall n \in \mathbf{N}_\mathcal{T}, r \in \mathbf{R}_n \tag{33}$$

Constraint (33) ensures support vehicles are assigned to subtasks following a **first-come, first-served** policy. For each subtask r of a task n, it selects the support vehicle that becomes available first.

3.4. Theoretical discussion

In this section, we compare some weaknesses of the first model, VRPSC^1, with the second, VRPSC^2. The latter is a more compact model based on fewer variables. Table 5 illustrates the size difference between VRPSC^1 and VRPSC^2 for some representative real-world instances. The compactness of VRPSC^2 arises from its underlying graph structure, which is built on a smaller set of nodes and arcs compared to VRPSC^1. This reduction is due to a more aggregated time-space representation that removes redundant transitions. Consequently, the feasible solution space is captured more efficiently, with fewer decision variables and constraints. Conversely, VRPSC^1 uses an extended network with dummy nodes that explicitly list all possible transitions between service and synchronization events. While this offers greater modeling flexibility, it also significantly increases the model size and computational time. The main theoretical implication is that VRPSC^2 strikes a better trade-off between expressiveness and computational efficiency. Its smaller graph reduces the number of binary variables and constraints, demonstrating the efficiency differences between the two models.

Given that the linear relaxations of the three models are relatively weak, we introduce several valid inequalities to strengthen the relaxation and improve computational performance.

$$\sum_{n\in\mathbf{N}_\mathcal{T}}\left(\sum_{r\in\mathbf{R}_n}\partial^v_{nr}\right)\cdot y^v_n+\sum_{(i,j)\in\mathbf{A}}d^v_{ij}\cdot x^v_{ij}\le s^v_\delta,\quad\forall v\in\mathcal{V}_p \tag{34}$$

$$s^v_{nb}+\sum_{(n,j)\in\mathbf{A}}d^v_{nj}\cdot x^v_{nj}+\left(d^v_{j\delta}+\sum_{r\in\mathbf{R}_j}\partial^v_{jr}\right)\cdot x^v_{j\delta}\le s^v_\delta,\quad\forall v\in\mathcal{V}_p,\forall n\in\mathbf{N}_\mathcal{T} \tag{35}$$

$$\sum_{(i,n)\in\mathbf{A}}\left(d^v_{\sigma i}+\sum_{r\in\mathbf{R}_i}\partial^v_{ir}+d^v_{in}\right)\cdot x^v_{in}\le s^v_{na},\quad\forall v\in\mathcal{V}_p,\forall n\in\mathbf{N}_\mathcal{T} \tag{36}$$

$$s^v_{na}+\sum_{r\in\mathbf{R}_n}\partial^v_{nr}\le s^v_{nb},\quad\forall v\in\mathcal{V}_p,\forall n\in\mathbf{N}_\mathcal{T} \tag{37}$$

Where $b=|\mathbf{R}_n|+1, a=1$.

Inequalities (34) and (35) define a lower bound on the vehicles' return time to the depot, denoted by s^v_δ. Inequalities (34) ensure that the return time to the depot for each primary vehicle is greater than or equal to the sum of travel and service times. In contrast, inequalities (35) compute a bound for s^v_δ when the vehicle returns to the depot via node i followed by node j. Inequalities (36) define a lower bound on the start time of the first subtask of each task, based on its predecessors. Inequalities (37) define a lower bound on the completion time of the last subtask executed by a primary vehicle, which must execute all subtasks associated with the task.

4. Computational experiments

In this section, we present computational results comparing the performance of the proposed models on 100 real-world instances from a Moroccan mining company. At the processing site, the company operates 10 zones where loading operations of either ore or waste material must be performed before transportation to designated destinations. All tasks must be executed by a fleet of up to five loaders and eight trucks. The travel times of primary and support vehicles between the zones and their respective destinations, as well as the loading and unloading durations, were measured directly on-site. We assume that the constants M_1 and M_2 are fixed to a 24-hour time horizon due to operational rules in the mining industry. If the number of problem subtasks increases, these values can be adjusted through benchmarking. A time limit of 10 minutes was set for each model.

For clarity of exposition, the instances are labeled according to the format: $\sum_{i\in\mathcal{T}}|\mathbf{R}_i|-|\mathcal{T}|-|\mathcal{V}_p|-|\mathcal{V}_s|$, where $\sum_{i\in\mathcal{T}}|\mathbf{R}_i|$ denotes the total number of subtasks, $|\mathcal{T}|$ represents the number of tasks containing at least one subtask, $|\mathcal{V}_p|$ is the number of primary vehicles, and $|\mathcal{V}_s|$ is the number of support vehicles.

The MILPs are implemented in GUROBI version 12.0 via the Python 11.0 API. The tests were performed on a Dell 64-bit Oracle Linux machine (Intel i7, four CPU cores, 3.40 GHz).

4.1. Numerical Results

In this section, we present the numerical results of the three models. For each model, we provide the computational time (Time), the optimality gap in % (Gap), the objective value expressed in hours (obj), the number of primary vehicles used (P.V.), and finally the number of support vehicles used (S.V.). Detailed results of the experiments are reported in Tables 3, 4, 6, and 7.

The comparative analysis of the VRPSC^1 and VRPSC^2 models, conducted over 89 solved instances, shows that VRPSC^2 significantly outperforms VRPSC^1 across all performance metrics. The data in Tables 3 and 4 reveal consistent improvements in both solution quality and computational efficiency. Statistical analysis using paired t-tests confirms a clear performance advantage for VRPSC^2, with results indicating high statistical significance. On

average, VRPSC^2 decreases computation time by 30.4% (381.7s vs. 265.6s; $p < 0.001$; $d = 0.55$) while enhancing solution quality by 6.4% (3.487 vs. 3.265; $p < 0.001$; $d = 0.58$). The observed effect sizes, classified as moderate to large based on Cohen's standards (according to d value), further support the practical significance of these improvements.

Figure 4 demonstrates the computational advantage of VRPSC^2, exhibiting consistently lower computational times throughout the entire range of instance complexity. This advantage is especially significant for large-scale instances, where VRPSC^2 reduces the average computational time from 533.28 s to 10.40 s for 80-subtask instances, representing a 98.1% reduction. Figure 5 further corroborates these findings, emphasizing VRPSC^2's enhanced capacity to attain solution optimality in larger-scale instances.

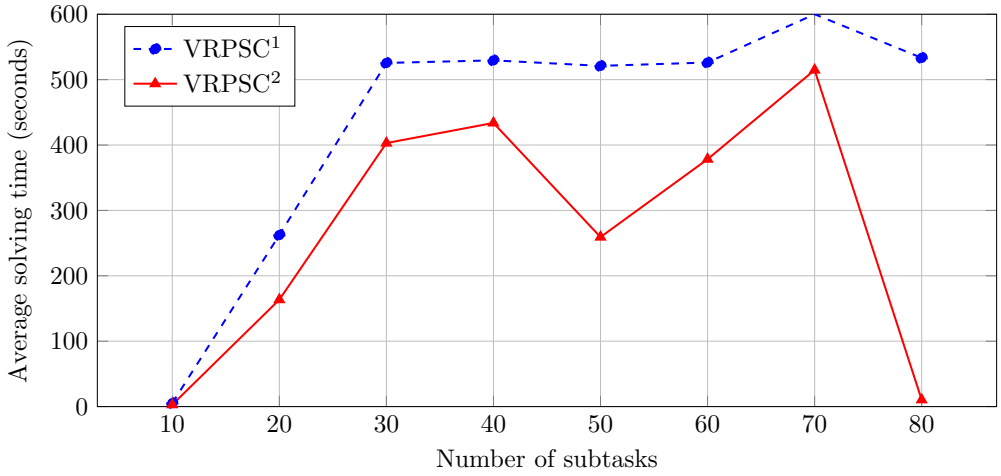

Figure 4. Average solving time by number of subtasks for VRPSC^1 and VRPSC^2.

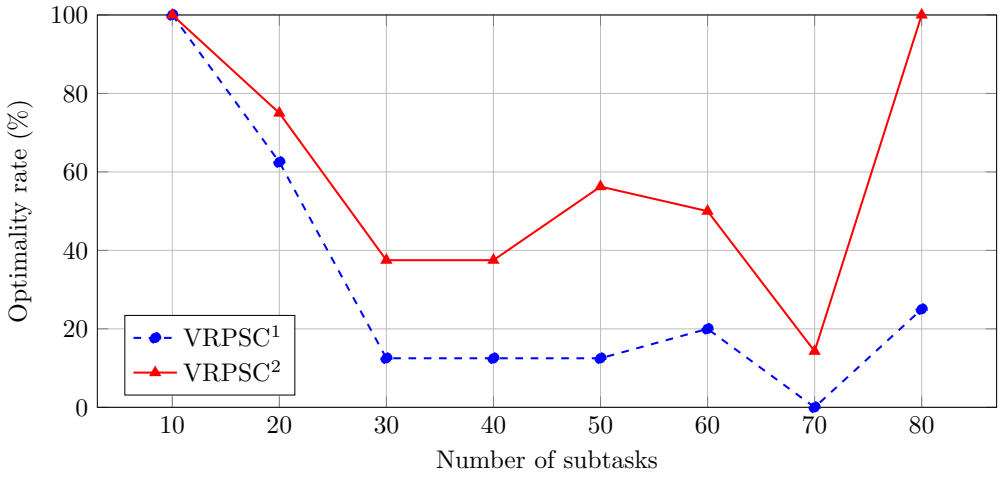

Figure 5. Success rate (optimality proven) by number of subtasks for VRPSC^1 and VRPSC^2

Although both models perform similarly on small instances—achieving minimal computational times and 100% success rates for 10 subtasks—a pronounced divergence emerges as problem complexity increases. VRPSC^1 exhibits notable performance degradation, frequently failing to converge within the allotted 600-second time limit: its success rate drops to 62.5% for 20 subtasks and falls to 12.5% for instances with 30–50 subtasks. In contrast,

Table 3. Results of VRPSC[1] and VRPSC[2] (Part 1).

	VRPSC[1]					VRPSC[2]				
Instance	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**
10-4-1-2	**0.1**	**0.000**	**2.25**	1	2	0.3	**0.000**	**2.25**	1	2
10-4-1-3	**0.1**	**0.000**	**2.12**	1	3	**0.1**	**0.000**	**2.12**	1	3
10-4-2-2	1.3	**0.000**	**1.83**	2	2	**0.8**	**0.000**	**1.83**	2	2
10-4-2-3	1.0	**0.000**	**1.37**	2	3	**0.6**	**0.000**	**1.37**	2	3
10-6-2-4	**0.4**	**0.000**	**1.23**	2	4	0.5	**0.000**	**1.23**	2	4
10-6-2-5	**0.3**	**0.000**	**1.23**	2	5	**0.3**	**0.000**	**1.23**	2	5
10-6-3-4	48.6	**0.000**	**1.69**	2	2	**21.6**	**0.000**	**1.69**	2	2
10-6-3-5	**0.4**	**0.000**	**0.90**	3	5	0.5	**0.000**	**0.90**	3	5
10-8-3-6	1.4	**0.000**	**0.95**	3	6	**1.0**	**0.000**	**0.95**	3	6
10-8-3-7	**0.9**	**0.000**	**0.95**	3	7	1.6	**0.000**	**0.95**	3	7
10-8-4-6	1.3	**0.000**	**0.78**	4	6	**1.2**	**0.000**	**0.78**	4	6
10-8-4-7	**0.9**	**0.000**	**0.78**	4	7	1.2	**0.000**	**0.78**	4	7
10-10-4-7	**2.7**	**0.000**	**0.78**	4	7	3.6	**0.000**	**0.78**	4	7
10-10-4-8	7.0	**0.000**	**0.78**	4	7	**2.9**	**0.000**	**0.78**	4	7
10-10-5-7	16.5	**0.000**	**0.68**	5	7	**6.4**	**0.000**	**0.68**	5	7
10-10-5-8	**1.8**	**0.000**	**0.65**	5	8	3.4	**0.000**	**0.65**	5	8
20-4-1-2	0.9	**0.000**	**4.00**	1	2	**0.6**	**0.000**	**4.00**	1	2
20-4-1-3	0.3	**0.000**	**3.53**	1	3	**0.2**	**0.000**	**3.53**	1	3
20-4-2-2	600.0	0.059	**3.57**	2	2	**19.2**	**0.000**	**3.57**	2	2
20-4-2-3	600.0	0.144	2.57	2	3	**73.3**	**0.000**	**2.55**	2	3
20-6-2-4	228.6	**0.000**	**2.07**	2	4	**65.7**	**0.000**	**2.07**	2	4
20-6-2-5	1.8	**0.000**	**2.06**	2	5	**0.7**	**0.000**	**2.06**	2	5
20-6-3-4	149.7	**0.000**	**2.06**	3	4	**5.1**	**0.000**	**2.06**	3	4
20-6-3-5	**0.7**	**0.000**	**2.06**	3	5	**0.7**	**0.000**	**2.06**	3	5
20-8-3-6	9.9	**0.000**	**1.53**	3	6	**3.6**	**0.000**	**1.53**	3	6
20-8-3-7	**1.8**	**0.000**	**1.53**	3	7	3.3	**0.000**	**1.53**	3	7
20-8-4-6	600.0	0.096	1.36	4	6	600.0	**0.082**	**1.34**	4	6
20-8-4-7	**600.0**	**0.024**	**1.26**	4	7	**600.0**	**0.024**	**1.26**	4	7
20-10-4-7	**600.0**	**0.064**	**1.25**	4	7	**600.0**	**0.064**	**1.25**	4	7
20-10-4-8	17.7	**0.000**	**1.17**	4	8	**16.3**	**0.000**	**1.17**	4	8
20-10-5-7	600.0	**0.068**	**1.18**	5	7	**600.0**	**0.068**	**1.18**	5	7
20-10-5-8	186.2	**0.000**	**1.10**	5	8	**27.6**	**0.000**	**1.10**	5	8
30-4-1-2	6.8	**0.000**	**5.50**	1	2	**2.0**	**0.000**	5.51	1	2
30-4-1-3	1.8	**0.000**	**4.96**	1	3	**0.4**	**0.000**	**4.96**	1	3
30-4-2-2	600.0	0.314	5.22	2	2	**19.2**	**0.000**	**5.20**	2	2
30-4-2-3	600.0	0.162	3.70	2	3	**280.8**	**0.000**	**3.59**	2	3
30-6-2-4	600.0	0.103	3.02	2	4	**600.0**	**0.032**	**2.80**	2	4
30-6-2-5	600.0	0.004	2.72	2	5	**2.3**	**0.000**	**2.71**	2	5
30-6-3-4	600.0	0.293	2.90	3	4	**600.2**	**0.208**	**2.59**	3	4
30-6-3-5	600.0	0.177	2.49	3	5	**600.0**	**0.060**	**2.18**	3	5
30-8-3-6	600.0	0.126	2.14	3	6	**600.0**	**0.074**	**2.02**	3	6
30-8-3-7	600.0	0.051	1.97	3	7	**600.0**	**0.021**	**1.91**	3	7
30-8-4-6	**600.0**	**0.171**	**1.93**	4	6	600.0	0.179	1.95	4	6
30-8-4-7	600.0	0.140	1.86	4	7	**600.0**	**0.075**	**1.73**	4	7
30-10-4-7	600.0	0.131	1.76	4	7	**600.0**	**0.110**	**1.72**	4	7
30-10-4-8	600.0	0.089	1.68	4	8	**600.0**	**0.032**	**1.58**	4	8
30-10-5-7	600.0	0.101	1.69	5	7	**600.0**	**0.084**	**1.66**	5	7
30-10-5-8	600.0	0.050	1.60	5	8	**142.9**	**0.000**	**1.52**	5	8

Table 4. Results of VRPSC[1] and VRPSC[2] (Part 2).

Instance	Time	Gap	Obj	P.V.	S.V.	Time	Gap	Obj	P.V.	S.V.
40-4-1-2	68.7	**0.000**	**7.68**	1	2	**2.9**	**0.000**	**7.68**	1	2
40-4-1-3	2.7	**0.000**	**6.62**	1	3	**0.6**	**0.000**	**6.62**	1	3
40-4-2-2	600.0	0.452	7.38	2	2	**106.0**	**0.000**	**7.18**	2	2
40-4-2-3	600.0	0.342	5.23	2	3	**498.3**	**0.000**	**5.01**	2	3
40-6-2-4	600.0	0.204	4.21	2	4	**600.0**	**0.090**	**3.68**	2	4
40-6-2-5	600.0	0.012	3.39	2	5	**600.0**	**0.009**	**3.38**	2	5
40-6-3-4	600.0	0.240	3.66	3	4	**600.0**	**0.190**	**3.43**	3	4
40-6-3-5	600.1	0.051	2.93	3	5	**600.0**	**0.018**	**2.83**	3	5
40-8-3-6	600.0	0.194	2.94	3	6	**600.0**	**0.016**	**2.52**	3	6
40-8-3-7	600.0	0.061	2.64	3	7	**81.1**	**0.000**	**2.48**	3	7
40-8-4-6	600.0	0.196	2.71	4	6	**600.0**	**0.128**	**2.50**	4	6
40-8-4-7	-	-	-	-	-	**251.3**	**0.000**	**2.18**	4	7
40-10-4-7	-	-	-	-	-	**600.1**	**0.150**	**2.26**	4	7
40-10-4-8	600.0	0.241	2.53	4	8	**600.0**	**0.040**	**2.00**	4	8
40-10-5-7	600.0	0.239	2.47	5	7	**600.0**	**0.227**	**2.29**	5	7
40-10-5-8	600.0	0.168	2.26	5	8	**600.0**	**0.041**	**1.96**	5	8
50-4-1-2	9.6	**0.000**	**8.30**	1	2	**1.1**	**0.000**	**8.30**	1	2
50-4-1-3	6.1	**0.000**	**8.08**	1	3	**0.7**	**0.000**	**8.08**	1	3
50-4-2-2	600.0	0.455	7.93	2	2	**118.1**	**0.000**	**7.86**	2	2
50-4-2-3	600.0	0.312	6.28	2	3	**600.0**	**0.144**	**5.42**	2	3
50-6-2-4	600.0	0.275	5.68	2	4	**600.0**	**0.131**	**4.74**	2	4
50-6-2-5	600.1	0.148	4.87	2	5	**600.0**	**0.048**	**4.36**	2	5
50-6-3-4	600.0	0.202	5.20	3	4	**600.0**	**0.151**	**4.89**	3	4
50-6-3-5	600.0	0.158	4.93	3	5	**600.0**	**0.037**	**4.31**	3	5
50-8-3-6	600.0	0.023	4.36	3	6	**19.5**	**0.000**	**4.26**	2	6
50-8-3-7	600.0	0.023	4.36	3	7	**10.7**	**0.000**	**4.26**	2	7
50-8-4-6	600.0	0.043	4.45	4	6	**17.0**	**0.000**	**4.26**	2	6
50-8-4-7	600.0	0.156	5.05	4	7	**14.8**	**0.000**	**4.26**	2	7
50-10-4-8	600.0	0.044	2.95	4	8	**41.3**	**0.000**	**2.83**	4	8
50-10-5-7	600.0	0.190	3.48	5	7	**600.0**	**0.031**	**2.91**	5	7
50-10-5-8	600.1	0.057	2.99	5	8	**65.0**	**0.000**	**2.83**	4	8
60-4-1-3	30.8	**0.000**	**9.53**	1	3	**1.4**	**0.000**	**9.53**	1	3
60-4-2-2	-	-	-	-	-	**429.7**	**0.000**	**9.83**	2	2
60-4-2-3	600.0	0.079	6.98	2	3	**288.2**	**0.000**	**6.70**	2	3
60-6-2-4	600.0	0.134	6.28	2	4	**600.0**	**0.007**	**5.48**	2	4
60-6-2-5	431.1	**0.000**	**5.44**	2	5	**9.8**	**0.000**	**5.44**	2	5
60-6-3-4	600.0	0.326	5.65	3	4	**600.0**	**0.247**	**5.06**	3	4
60-6-3-5	600.0	0.326	5.65	3	5	**600.0**	**0.106**	**4.26**	3	5
60-8-3-6	600.0	0.289	5.44	3	6	**600.0**	**0.087**	**4.25**	3	6
60-8-3-7	600.0	0.193	4.81	3	7	**54.0**	**0.000**	**3.88**	3	7
60-8-4-6	600.0	0.336	5.83	4	6	**600.0**	**0.063**	**4.14**	3	6
70-4-2-3	-	-	-	-	-	**2.8**	**0.000**	**8.38**	2	3
70-6-2-4	-	-	-	-	-	**600.0**	**0.142**	**6.67**	2	4
70-6-2-5	-	-	-	-	-	**600.0**	**0.014**	**5.80**	2	5
70-6-3-4	600.1	0.349	8.08	3	4	**600.0**	**0.231**	**6.84**	3	4
70-6-3-5	600.0	0.271	7.22	3	5	**600.0**	**0.047**	**5.52**	3	5
70-8-3-6	-	-	-	-	-	**600.0**	**0.099**	**4.73**	3	6
70-8-4-6	-	-	-	-	-	**600.0**	**0.127**	**4.88**	4	6
80-8-3-6	-	-	-	-	-	**16.4**	**0.000**	**7.47**	3	6
80-8-3-7	-	-	-	-	-	**12.1**	**0.000**	**7.47**	3	7
80-8-4-6	-	-	-	-	-	**10.9**	**0.000**	**7.47**	3	6
80-8-4-7	333.1	**0.000**	**7.47**	4	7	**2.2**	**0.000**	**7.47**	4	7
AVG	381.7	0.10	3.49	2.9	5.1	**273.6**	**0.04**	**3.58**	2.8	5.1

VRPSC^2 demonstrates strong robustness, maintaining success rates of 75% at 20 subtasks and 37.5–56.25% for instances with 30–50 subtasks.

Analysis of performance distributions further supports these findings: VRPSC^2 outperforms VRPSC^1 on 45 instances in terms of computational efficiency (5:1 ratio) and on 49 instances regarding solution quality (16:1 ratio). For example, in instance 20-4-2-3, VRPSC^1 reaches the time limit with an optimality gap of 0.144%, while VRPSC^2 solves the same instance in 73.3 seconds with a zero gap. In large-scale instances, VRPSC^2 consistently achieves objective values that are equal to or better than VRPSC^1's while significantly reducing computational time. The scalability of VRPSC^2 is especially clear in these cases, where it attains a 100% success rate for 80-subtask problems compared to 25% for VRPSC^1. Notably, the similarity in fleet configurations between the two models suggests that the performance improvements of VRPSC^2 are due to enhancements in the model formulation rather than structural changes to the solutions. This is supported by Table 5, which reports the number of constraints and variables for some instances. This table indicates that VRPSC^2 consistently reduces the number of constraints and variables, demonstrating the efficiency and effectiveness of the improved formulation.

Table 5. Number of constraints and variables in VRPSC^1 and VRPSC^2.

Instance	VRPSC^1			VRPSC^2		
	Constraints	Variables		Constraints	Variables	
		Binary	Continuous		Binary	Continuous
10-10-5-8	3544	1690	417	3215	1742	300
20-10-5-8	5079	2120	547	4285	1872	430
30-10-5-8	6796	2718	677	5355	2002	560
40-10-5-8	8682	3472	807	6425	2132	690
50-10-5-8	11881	5438	937	7495	2262	820
60-8-4-6	9917	4838	781	5318	1440	711
70-6-3-5	10902	6446	673	3912	976	633
80-8-4-7	22269	14100	1079	7607	1804	1002

Sensitivity analysis regarding fleet size, as reported in Tables 3 and 4, reveals that increasing the number of vehicles (primary or support) generally reduces objective values, reflecting improved route optimization. This effect is particularly pronounced for small instances (10–20 subtasks), where additional vehicles accelerate task completion. For larger-scale instances (30–80 subtasks), the marginal benefit decreases, suggesting resource saturation and highlighting the importance of optimal fleet sizing.

In summary, the full experimental results—including tabular data, computational times, success rates, and statistical validation—demonstrate that VRPSC^2 is a more efficient model. It offers better scalability, greater computational efficiency, and increased robustness, which is particularly important for medium to large-scale instances where operational requirements are most stringent.

Tables 6 and 7 present the computational results for the VRPSC^2 and VRPSSC models. For most small instances, both models achieve identical objective values with very low computational time. For medium and large instances, the two models perform comparably in terms of the number of vehicles used. However, the VRPSSC model exhibits a slight increase in objective value and computational time, particularly for larger instances. The average values (AVG) confirm that VRPSC^2 achieves a slightly lower mean computational time (273.6 s vs. 281.4 s) and an almost identical average optimality gap compared to VRPSSC, while the average objective value and number of vehicles remain unchanged.

These results indicate that scheduling support vehicles under a First-Come, First-Served (FCFS) policy yields only a marginal reduction in solution quality. Nonetheless, this strategy provides a more structured and operationally realistic sequencing of support vehicle activities, enhancing the robustness and practical relevance of the model in real-world settings where vehicle arrival order critically affects task coordination and overall system efficiency.

Table 6. Results of VRPSC^2 and VRPSSC (Part 1).

	VRPSC^2					VRPSSC				
Instance	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**
10-4-1-2	0.3	**0.000**	**2.25**	1	2	**0.1**	**0.000**	**2.25**	1	2
10-4-1-3	0.1	**0.000**	**2.12**	1	3	**0.1**	**0.000**	**2.12**	1	3
10-4-2-2	0.8	**0.000**	**1.83**	2	2	**0.7**	**0.000**	1.90	2	2
10-4-2-3	**0.6**	**0.000**	**1.37**	2	3	**0.6**	**0.000**	**1.37**	2	3
10-6-2-4	**0.5**	**0.000**	**1.23**	2	4	**0.5**	**0.000**	**1.23**	2	4
10-6-2-5	**0.3**	**0.000**	**1.23**	2	5	**0.3**	**0.000**	**1.23**	2	5
10-6-3-4	**21.6**	**0.000**	**1.7**	2	2	24.2	**0.000**	**1.70**	2	2
10-6-3-5	0.5	**0.000**	**0.90**	3	5	**0.4**	**0.000**	**0.90**	3	5
10-8-3-6	**1.0**	**0.000**	**0.95**	3	6	1.4	**0.000**	**0.95**	3	6
10-8-3-7	1.6	**0.000**	**0.95**	3	7	**1.5**	**0.000**	**0.95**	3	7
10-8-4-6	**1.2**	**0.000**	**0.78**	4	6	1.7	**0.000**	**0.78**	4	6
10-8-4-7	**1.2**	**0.000**	**0.78**	4	7	1.8	**0.000**	**0.78**	4	7
10-10-4-7	**3.6**	**0.000**	**0.78**	4	7	3.8	**0.000**	**0.78**	4	7
10-10-4-8	**2.9**	**0.000**	**0.78**	4	7	4.0	**0.000**	**0.78**	4	7
10-10-5-7	**6.4**	**0.000**	**0.68**	5	7	6.8	**0.000**	**0.68**	5	7
10-10-5-8	**3.4**	**0.000**	**0.65**	5	8	4.1	**0.000**	**0.65**	5	8
20-4-1-2	0.6	**0.000**	**4.00**	1	2	**0.5**	**0.000**	**4.00**	1	2
20-4-1-3	**0.2**	**0.000**	**3.53**	1	3	0.3	**0.000**	**3.53**	1	3
20-4-2-2	19.2	**0.000**	**3.57**	2	2	**2.6**	**0.000**	3.64	2	2
20-4-2-3	73.3	**0.000**	**2.55**	2	3	**30.5**	**0.000**	2.58	2	3
20-6-2-4	65.7	**0.000**	**2.07**	2	4	**19.3**	**0.000**	2.08	2	4
20-6-2-5	**0.7**	**0.000**	**2.06**	2	5	1.4	**0.000**	**2.06**	2	5
20-6-3-4	**5.1**	**0.000**	**2.06**	3	4	35.0	**0.000**	2.07	3	4
20-6-3-5	**0.7**	**0.000**	**2.06**	3	5	1.7	**0.000**	**2.06**	3	5
20-8-3-6	**3.6**	**0.000**	**1.53**	3	6	142.7	**0.000**	**1.53**	3	6
20-8-3-7	**3.3**	**0.000**	**1.53**	3	7	10.5	**0.000**	**1.53**	3	7
20-8-4-6	**600.0**	**0.082**	**1.34**	4	6	600.0	0.115	1.39	4	6
20-8-4-7	**600.0**	**0.031**	**1.27**	4	7	**600.0**	**0.031**	**1.27**	4	7
20-10-4-7	**600.0**	**0.064**	**1.25**	4	7	600.0	0.100	1.30	4	7
20-10-4-8	**16.3**	**0.000**	**1.17**	4	8	49.0	**0.000**	**1.17**	4	8
20-10-5-7	**600.0**	**0.068**	**1.18**	5	7	**600.0**	**0.068**	**1.18**	5	7
20-10-5-8	**27.6**	**0.000**	**1.10**	5	8	183.0	**0.000**	**1.10**	5	8
30-4-1-2	2.0	**0.000**	**5.51**	1	2	**1.4**	**0.000**	**5.51**	1	2
30-4-1-3	0.4	**0.000**	**4.96**	1	3	**0.2**	**0.000**	**4.96**	1	3
30-4-2-2	19.2	**0.000**	**5.20**	2	2	**10.0**	**0.000**	5.22	2	2
30-4-2-3	280.8	**0.000**	**3.59**	2	3	**4.7**	**0.000**	3.63	2	3
30-6-2-4	600.0	**0.032**	**2.80**	2	4	**525.1**	**0.000**	2.82	2	4
30-6-2-5	**2.3**	**0.000**	**2.71**	2	5	4.8	**0.000**	**2.71**	2	5
30-6-3-4	600.2	0.208	2.59	3	4	**600.1**	**0.150**	**2.54**	3	4
30-6-3-5	**600.0**	**0.060**	**2.18**	3	5	600.0	0.068	2.20	3	5
30-8-3-6	**600.0**	**0.074**	**2.02**	3	6	600.0	0.101	2.08	3	6
30-8-3-7	**600.0**	0.021	1.91	3	7	**600.1**	**0.016**	**1.90**	3	7
30-8-4-6	**600.0**	**0.179**	**1.95**	4	6	600.0	0.188	1.97	4	6
30-8-4-7	600.0	0.075	1.73	4	7	**600.0**	**0.064**	**1.71**	4	7
30-10-4-7	600.0	0.110	1.72	4	7	**600.0**	**0.105**	**1.71**	4	7
30-10-4-8	600.0	0.032	1.58	4	8	**600.0**	**0.019**	**1.56**	4	8
30-10-5-7	**600.0**	**0.084**	**1.66**	5	7	600.0	0.126	1.74	5	7
30-10-5-8	**142.9**	**0.000**	**1.52**	5	8	600.0	0.160	1.81	5	8

Table 7. Results of VRPSC^2 and VRPSSC (Part 2).

	VRPSC^2					VRPSSC				
Instance	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**	**Time**	**Gap**	**Obj**	**P.V.**	**S.V.**
40-4-1-2	2.9	**0.000**	**7.68**	1	2	**1.7**	**0.000**	**7.68**	1	2
40-4-1-3	**0.6**	**0.000**	**6.62**	1	3	**0.6**	**0.000**	**6.62**	1	3
40-4-2-2	106.0	**0.000**	**7.18**	2	2	**52.5**	**0.000**	7.23	2	2
40-4-2-3	498.3	**0.000**	**5.01**	2	3	**3.1**	**0.000**	5.13	2	3
40-6-2-4	**600.0**	**0.090**	**3.68**	2	4	600.0	0.092	3.69	2	4
40-6-2-5	600.0	0.009	**3.38**	2	5	**76.6**	**0.000**	3.40	2	5
40-6-3-4	**600.0**	**0.190**	**3.43**	3	4	600.0	0.206	3.50	3	4
40-6-3-5	**600.0**	**0.018**	**2.83**	3	5	600.0	0.064	2.97	3	5
40-8-3-6	**600.0**	0.016	2.52	3	6	**600.0**	**0.008**	**2.50**	3	6
40-8-3-7	**81.1**	**0.000**	**2.48**	3	7	114.0	**0.000**	**2.48**	3	7
40-8-4-6	**600.0**	**0.128**	**2.50**	4	6	600.0	0.142	2.54	4	6
40-8-4-7	**251.3**	**0.000**	**2.18**	4	7	600.0	0.060	2.32	4	7
40-10-4-7	**600.1**	**0.150**	**2.26**	4	7	600.0	0.203	2.41	4	7
40-10-4-8	**600.0**	**0.040**	**2.00**	4	8	600.0	0.169	2.31	4	8
40-10-5-7	600.0	0.227	2.29	5	7	**600.0**	**0.145**	**2.20**	5	7
40-10-5-8	600.0	0.041	1.96	5	8	**600.0**	**0.031**	**1.94**	5	8
50-4-1-2	**1.1**	**0.000**	**8.30**	1	2	2.9	**0.000**	**8.30**	1	2
50-4-1-3	0.7	**0.000**	**8.08**	1	3	**0.4**	**0.000**	**8.08**	1	3
50-4-2-2	118.1	**0.000**	**7.86**	2	2	**13.2**	**0.000**	7.88	2	2
50-4-2-3	600.0	**0.144**	**5.42**	2	3	**22.7**	**0.000**	5.59	2	3
50-6-2-4	**600.0**	**0.131**	**4.74**	2	4	600.0	0.149	4.84	2	4
50-6-2-5	**600.0**	**0.048**	**4.36**	2	5	600.0	0.074	4.48	2	5
50-6-3-4	**600.0**	**0.151**	**4.89**	3	4	600.0	0.160	4.94	3	4
50-6-3-5	**600.0**	**0.037**	**4.31**	3	5	600.0	0.042	4.33	3	5
50-8-3-6	**19.5**	**0.000**	**4.26**	2	6	102.6	**0.000**	**4.26**	2	6
50-8-3-7	**10.7**	**0.000**	**4.26**	2	7	127.0	**0.000**	**4.26**	2	7
50-8-4-6	**17.0**	**0.000**	**4.26**	2	6	52.8	**0.000**	**4.26**	2	6
50-8-4-7	**14.8**	**0.000**	**4.26**	2	7	106.8	**0.000**	**4.26**	2	7
50-10-4-8	**41.3**	**0.000**	**2.83**	4	8	600.0	0.038	2.93	4	8
50-10-5-7	**600.0**	**0.031**	**2.91**	5	7	600.0	0.135	3.26	5	7
50-10-5-8	**65.0**	**0.000**	**2.83**	4	8	600.0	0.010	2.86	5	8
60-4-1-3	1.4	**0.000**	**9.53**	1	3	**1.1**	**0.000**	**9.53**	1	3
60-4-2-2	429.7	**0.000**	**9.83**	2	2	**238.8**	**0.000**	**9.88**	2	2
60-4-2-3	288.2	**0.000**	**6.70**	2	3	**11.3**	**0.000**	**6.70**	2	3
60-6-2-4	600.0	0.007	**5.48**	2	4	**551.6**	**0.000**	5.51	2	4
60-6-2-5	**9.8**	**0.000**	**5.44**	2	5	19.7	**0.000**	**5.44**	2	5
60-6-3-4	600.0	0.247	5.06	3	4	**600.0**	**0.240**	**5.01**	3	4
60-6-3-5	**600.0**	**0.106**	**4.26**	3	5	600.0	0.179	4.64	3	5
60-8-3-6	600.0	0.087	4.25	3	6	**600.0**	**0.076**	**4.20**	3	6
60-8-3-7	**54.0**	**0.000**	**3.88**	3	7	600.0	0.194	4.81	3	7
60-8-4-6	**600.0**	**0.063**	**4.14**	3	6	600.0	0.187	4.77	4	6
70-4-2-3	**2.8**	**0.000**	**8.38**	2	3	7.3	**0.000**	**8.38**	2	3
70-6-2-4	**600.0**	**0.142**	**6.67**	2	4	600.0	0.211	7.25	2	4
70-6-2-5	**600.0**	**0.014**	**5.80**	2	5	600.0	0.024	5.86	2	5
70-6-3-4	**600.0**	**0.231**	**6.84**	3	4	600.0	0.242	6.94	3	4
70-6-3-5	**600.0**	**0.047**	**5.52**	3	5	600.0	0.076	5.69	3	5
70-8-3-6	**600.0**	**0.099**	**4.73**	3	6	600.0	0.163	5.09	3	6
70-8-4-6	600.0	0.127	4.88	4	6	**600.0**	**0.103**	**4.75**	4	6
80-8-3-6	**16.4**	**0.000**	**7.47**	3	6	36.4	**0.000**	**7.47**	3	6
80-8-3-7	**12.1**	**0.000**	**7.47**	3	7	229.2	**0.000**	**7.47**	3	7
80-8-4-6	**10.9**	**0.000**	**7.47**	3	6	55.4	**0.000**	**7.47**	3	6
80-8-4-7	**2.2**	**0.000**	**7.47**	4	7	44.1	**0.000**	**7.47**	4	7
AVG	**273.6**	**0.04**	3.58	2.8	5.1	281.4	0.05	3.63	2.8	5.1

To confirm the statistical validity of our observations, a paired t-test was conducted to evaluate the significance of the differences in objective values between VRPSC^2 and VRPSSC. The results revealed a statistically significant difference ($t(99) = -3.88$, $p < 0.001$), with VRPSC^2 producing slightly better solutions (mean difference $= -0.05$). The effect size, measured using Cohen's d (-0.388), indicates a small to moderate effect magnitude.

These findings are consistent with the visual evidence shown in Figure 6, where the close alignment of the performance curves and the substantial overlap of confidence intervals emphasize the functional similarity between the two formulations. Although the difference is statistically significant, its moderate magnitude suggests that VRPSSC maintains nearly equivalent optimization performance as VRPSC^2, while potentially offering additional operational benefits not explicitly acknowledged in this comparative analysis.

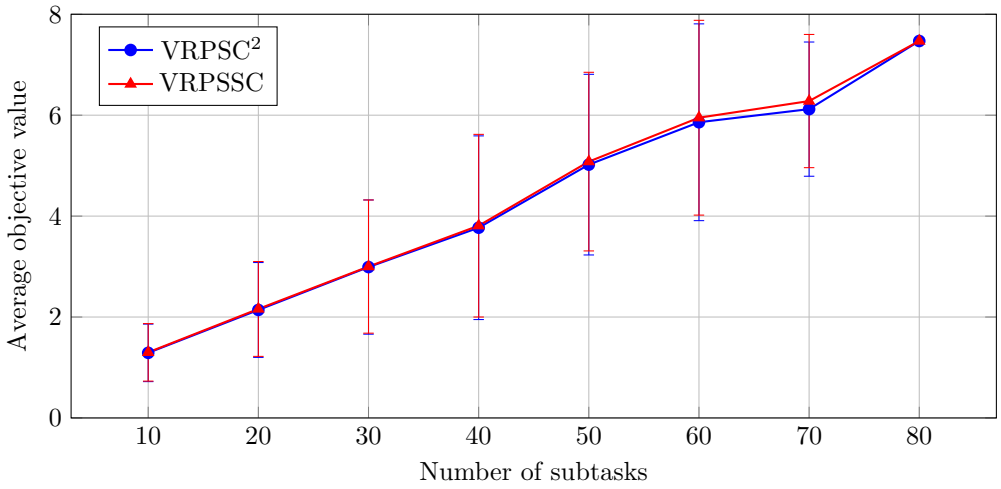

Figure 6. Average objective values by number of subtasks for VRPSC^2 and VRPSSC.

The scalability analysis of VRPSC^1, VRPSC^2, and VRPSSC across all 100 instances reveals significant computational challenges as problem complexity increases. The three models demonstrate robust performance on small instances but struggle with larger configurations. A clear pattern emerges when examining instances that hit the 10-minute time limit: computational complexity rises sharply with the ratio of subtasks (TS) to the total number of vehicles available (TV). Table 8 presents a detailed analysis across ranges of this ratio, considering time limit and complete failures ("-") in VRPSC^1 as scalability limitations.

Table 8. Scalability Analysis by TS/TV Ratio Ranges.

Ratio (TS/T.V)	**Time Limit Rate (%)**		
Range	**VRPSC^1**	**VRPSC^2**	**VRPSSC**
< 1.5	0	0	0
$1.5 - 2.5$	45.5	36.4	45.5
$2.5 - 3.5$	58.8	58.8	58.8
> 3.5	82.0	44.3	44.3
Total instances	**65.0**	**41.0**	**42.0**

From Table 8, two major trends can be observed. First, VRPSC^2 emerges as the most robust approach, particularly for ratios exceeding 3.5. This model reduces the proportion of instances reaching the time limit by 46% compared to VRPSC^1 (44.3% versus 82.0%), thereby maintaining a success rate of 55.7%, whereas VRPSC^1 achieves only 18%. Second, VRPSSC demonstrates comparable performance to VRPSC^2, representing a significant improvement over VRPSC^1. Its main advantage lies in superior stability for intermediate ratios.

Furthermore, the exponential degradation in VRPSC^1 performance with increasing ratios indicates that computational complexity grows faster than the availability of vehicle resources. The 65% time-limit failure rate observed across all 100 instances clearly illustrates its limited scalability, particularly beyond a ratio of 2.5. These findings substantiate the hypothesis that exact optimization approaches reach intrinsic performance limits when ratios exceed 3.5, as even the most advanced formulations (VRPSC^2 and VRPSSC) succeed in finding optimal solutions in only about 55% of cases within 10 minutes. Overall, the results underscore the need for heuristic or decomposition-based strategies to effectively address large-scale instances characterized by high task-to-vehicle ratios.

5. Conclusion

This paper investigates a variant of the Vehicle Routing Problem with Synchronization Constraints (VRPSC) and introduces three new models: VRPSC^1, VRPSC^2, and VRPSSC. The first two models differ in how tasks are disaggregated: VRPSC^1 duplicates nodes, whereas VRPSC^2 uses additional variables and constraints to allocate subtasks. The VRPSSC model further incorporates a First-Come, First-Served (FCFS) scheduling constraint for support vehicles. Computational experiments conducted on 100 real-world instances show that VRPSC^2 achieves an approximately 30% reduction in solution time compared to VRPSC^1, while consistently providing solutions with a gap below 0.25% in less than 10 minutes. Regarding VRPSSC, the results indicate that adding the FCFS constraint for support vehicles maintains solution quality and computational time comparable to those of VRPSC^2.

This approach improves planning, robustness, and scalability, making it suitable for practical applications such as construction, urban logistics, or transportation, where perfect synchronization and an *FCFS* policy are desirable. Its practical utility could be further enhanced by integrating the model into a decision-support system with a graphical interface, enabling planners to efficiently generate feasible plans within ten minutes and respond dynamically to disruptions, such as truck breakdowns or driver unavailability.

However, the study has limitations. It relies on the assumption of perfect synchronization, which may not hold in dynamic environments, and on the use of a commercial solver (Gurobi) to solve the MIP-based formulation, which may limit scalability for very large instances without dedicated heuristics or acceleration strategies.

Future research should explore several promising directions. First, developing metaheuristics, such as Adaptive Large Neighborhood Search, would enable solving larger problem instances. The model could also be extended to handle dynamic settings with real-time task arrivals and to incorporate uncertainty in travel or service times. Another critical improvement involves explicitly modeling and minimizing vehicle waiting times to significantly enhance practical applicability. This is especially important for operations with heterogeneous fleets, where effectively managing delays during subtask execution would markedly improve the models' realism and operational efficiency.

REFERENCES

1. M. Drexl, *Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints*, Transportation Science, vol. 46, no. 3, pp. 297–316, 2012.
2. T. Gschwind, *A comparison of column-generation approaches to the Synchronized Pickup and Delivery Problem*, European Journal of Operational Research, vol. 247, no. 1, pp. 60–71, 2015.
3. M.A. Salazar-Aguilar, A. Langevin, and G. Laporte, *The synchronized arc and node routing problem: Application to road marking*, Computers & Operations Research, vol. 40, no. 7, pp. 1708–1715, 2013.
4. F. Meisel and H. Kopfer, *Synchronized Routing of Active and Passive Means of Transport*, OR Spectrum, vol. 36, pp. 65–89, 2013.
5. G. Perboli, R. Tadei, and D. Vigo, *The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics*, Transportation Science, vol. 45, no. 3, pp. 364–380, 2009.
6. C.E. Cortés, M. Matamala, and C. Contardo, *The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method*, European Journal of Operational Research, vol. 200, no. 3, pp. 711–724, 2010.
7. G. Desaulniers, *Branch-and-Price-and-Cut for the Split-Delivery Vehicle Routing Problem with Time Windows*, Operations Research, vol. 58, no. 1, pp. 179–192, 2010.
8. C. Tilk, N. Bianchessi, M. Drexl, S. Irnich, and F. Meisel, *Branch-and-Price-and-Cut for the Active-Passive Vehicle-Routing Problem*, Transportation Science, vol. 52, no. 4, pp. 300–319, 2018.

9. N.-H. Quttineh, T. Larsson, K. Lundberg, and K. Holmberg, *Military aircraft mission planning: a generalized vehicle routing model with synchronization and precedence*, EURO Journal on Transportation and Logistics, vol. 2, pp. 1–30, 2013.
10. M. Fink, G. Desaulniers, M. Frey, F. Kiermaier, R. Kolisch, and F. Soumis, *Column generation for vehicle routing problems with multiple synchronization constraints*, European Journal of Operational Research, vol. 272, no. 2, pp. 699–711, 2019.
11. J. Hof and M. Schneider, *Intraroute Resource Replenishment with Mobile Depots*, Transportation Science, vol. 55, no. 3, pp. 660–686, 2021.
12. G. Rix, L.-M. Rousseau, and G. Pesant, *A column generation algorithm for tactical timber transportation planning*, Journal of the Operational Research Society, vol. 66, no. 2, pp. 278–287, 2014.
13. R. Soares, A. Marques, P. Amorim, and J. Rasinmäki, *Multiple vehicle synchronisation in a full truck-load pickup and delivery problem: A case-study in the biomass supply chain*, European Journal of Operational Research, vol. 277, no. 1, pp. 174–194, 2019.
14. D. Bredström and M. Rönnqvist, *Combined vehicle routing and scheduling with temporal precedence and synchronization constraints*, European Journal of Operational Research, vol. 191, no. 1, pp. 19–31, 2008.
15. D.S. Mankowska, F. Meisel, and C. Bierwirth, *The home health care routing and scheduling problem with interdependent services*, Health Care Management Science, vol. 17, no. 1, pp. 15–30, 2014.
16. C. Hempsch and S. Irnich, *Vehicle Routing Problems with Inter-Tour Resource Constraints*, in *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, pp. 421–444, 2008.
17. A. Grimault, N. Bostel, and F. Lehuédé, *An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization*, Computers & Operations Research, vol. 88, pp. 1–14, 2017.
18. R. Soares, A. Marques, P. Amorim, and S.N. Parragh, *Synchronisation in vehicle routing: Classification schema, modelling framework and literature review*, European Journal of Operational Research, vol. 313, pp. 817–840, 2024.
19. Soares, Ricardo and Parragh, Sophie and Marques, Alexandra and Amorim, Pedro *The Robust Vehicle Routing Problem With Synchronization: Models and Branch-And-Cut Algorithms* Networks, vol. 86, pp. 296–324, 2025.