Pesquisa Operacional (2023) 43: e270829 p.1-29
doi: 10.1590/0101-7438.2023.043.00270829

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

MATH-HEURISTIC FOR THE CAPACITATED TWO-ECHELON VEHICLE ROUTING PROBLEM

José Carlos Fontoura Guimarães[1*] and Claudio Barbieri da Cunha[2]

Received January 2, 2023 / Accepted May 3, 2023

ABSTRACT. In this paper, we propose a math-heuristic that combines mathematical programming techniques with a heuristic approach based on the Simulated Annealing metaheuristic to solve the Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP). This problem arises in the context of a distribution network that is divided in two levels: satellite facilities that connect customers to fulfilment centers where freight originates. As it is an NP-hard problem, the proposed approach combines a cluster-first route-second math-heuristic in which approaches are more appropriate, particularly for problem sizes that are more commonly found in practice. The results of the experiments with benchmark instances show that such cluster-first route-second math-heuristic approach utilizing package solvers (CPLEX and TSP CONCORDE) can effectively help solving the CVRP for small instances when compared to an exact method**.** The experiments conducted on benchmark instances demonstrated the effectiveness of the proposed "cluster-first, route-second" math-heuristic approach, which utilizes package solvers such as CPLEX and TSP CONCORDE, in solving the CVRP for small instances, outperforming exact methods. This research contributes to demonstrating the potential applications of package solvers on heuristic structures for solving the CVRP. Although the presented math-heuristic has limitations, mainly due to the extensive usage of mathematical programming and the chosen characteristics of the implemented local search operators, it can quickly generate high-quality initial solutions for medium and large instances. By showcasing the "cluster-first, route-second" approach, this paper provides a framework that can be further improved by local search or embedded in other metaheuristics, such as GRASP or tabu search, and has practical implications for various industries.

Keywords: math-heuristic, 2E-CVRP, routing.

*Corresponding author

[1] University of São Paulo, Graduate Program in Transportation Engineering, Escola Politécnica, São Paulo, SP, Brazil – E-mail: jcfguimaraes@usp.br – http://orcid.org/0000-0002-1922-1175

[2] University of São Paulo, Graduate Program in Transportation Engineering, Escola Politécnica, São Paulo, SP, Brazil – E-mail: cbcunha@usp.br – http://orcid.org/0000-0002-9950-2830

1 INTRODUCTION

This article addresses the *two-echelon capacitated vehicle routing problem* (2E-CVRP). In this problem, the distribution network is divided in two levels or layers (*echelons*), as can be seen in Figure 1.

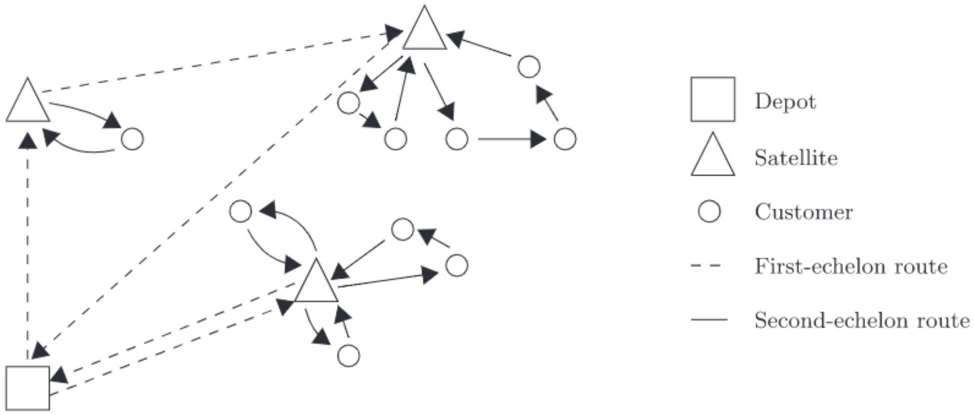


Figure 1 – Example of a 2E-CVRP problem solution.

Source: Sluijk et al. (2022).

The main depot and the intermediate facilities (satellite terminals) correspond to the main distribution structure, from where loads originate, and are usually located away from the large urban centers, configuring the first echelon or tier, denoted by – FE or 1E - with customers are serviced in the second echelon – SE or 2E - as described by Sluijk et al. (2022). The satellites, located at the edges or borders between layers, allow the consolidation and transshipment of products between the vehicles utilized on either echelon. This makes it possible to use larger vehicles in the first tier, which can transfer the loads from the depot to be delivered to the intermediate installations (satellite terminals) at times of lower traffic or that allow the traffic of larger vehicles (for example, in the evening or night), which are then distributed to the delivery points using smaller vehicles, usually more suitable for driving and parking in the most central and congested areas.

As highlighted by Crainic et al. (2009), two-tier systems are increasingly being used in large cities that face serious problems related to vehicle traffic. Examples of two-tier distribution systems include home delivery of e-commerce, distribution of supermarket products, multimodal cargo transport, among others (Sluijk et al., 2022). Solutions described in the literature present exact methods (or algorithms) and heuristic methods. State-of-the-art algorithms can solve only relatively small instances to optimality in reasonable computational time since the 2E-CVRP is of the NP-hard class (Arnold and Sorensen, 2019; Cuda et al., 2015). Heuristic approaches do not necessarily provide the optimal solution to the problem but allow obtaining near-optimal results for a NP-hard problem usually in very short times (Arnold and Sorensen, 2019). A math-heuristic combines heuristic techniques with operations research methods and tools to obtain effective solutions (Boschetti and Maniezzo, 2022).

This paper proposes the development of a math-heuristic based on the Simulated Annealing metaheuristic for the 2E-CVRP that derives from formulation developed by Sluijk et al. (2022).

The 2E-CVRP formulation proposed by the authors allows solving both the first and the second echelons of the routing problem. In this work, we propose a model that comprises a math-heuristic for solving the 2E-CVRP by applying a "cluster-First, route-Second" approach, commonly utilized in constructive heuristics to the CVRP (Prins et. al., 2014). With this approach, customers are first allocated to the satellites thus decomposing the problem into several Capacitated VRPs and it provides the main benefit of being computationally cheaper. However, the routes obtained later can be suboptimal (Singh et al., 2023). In the proposed math–heuristic two CVRP problems are solved after the clustering process (which defines the customers and load assigned to each satellite): the capacitated vehicle routing problem from the central depot to the satellites and the capacitated routing problem within each satellite.

Our motivation stems from the importance of demonstrating possible applications of package solvers in combination with heuristic approach to solve the CVRP. Although our proposed math-heuristic has limitations due to the usage of mathematical programming and chosen specific characteristics of the implemented local search operators, it aims to illustrate how the "cluster-first, route-second" approach can be used to quickly generate good-quality initial solutions that can be further improved by local search. Additionally, our approach can be integrated into a multi-start constructive heuristic such as GRASP (Feo and Resende, 1995) or even in a metaheuristic such as tabu search (Glover, 1990) or VNS (Hansen and Mladenović, 2001).

This article is organized as follows: in the next section the literature review of the problem is presented. The proposed heuristic is described in Section 3 and the results of computational experiments are reported in Section 4. Finally, the conclusions are in Section 5.

2 LITERATURE REVIEW

In the classic vehicle routing problem (VRP), the goal is to determine the route(s) to visit a set of service demand nodes, seeking to minimize the distance or the total cost and ensuring that each node is visited exactly once and the total demand on any route does not exceed the capacity of the respective vehicle. In other words, demands - pickups, deliveries, or service visits - must be allocated to each vehicle of an available fleet, and the routes (or sequence of stops) of each vehicle must minimize the total cost of the service, usually composed of the weighted sum of fixed costs and costs proportional to the distance traveled by vehicles and travel time, subject to constraints such as time windows (time periods in which the deliveries can be made), maximum distance or maximum duration of routes, among others (Cunha, 2006). It is a widely explored problem in the literature. Good recent references that provide a broad view of the state-of-the-art of this problem can be found, for example, in Vidal et al. (2014), Lahyani et al. (2015), Braekers et al. (2016) and Vidal et al. (2020).

The first formal definition of a 2E-VRP appeared in Crainic et al. (2009), where the authors study a rich variant of a 2E-VRP problem with various products and depots, time dependencies and vehicle synchronization, in a broader context of city logistics. More specifically, the problem studied included the selection of routes and the scheduling of the dispatched vehicles, as well as the selection of demand routes from the main terminals, via satellite terminals, to the final customers. The coordination and time-based synchronization of the vehicles at both levels were central elements of the problem. The authors investigated main aspects of the problem and proposed a mathematical formulation and a general methodology for its resolution based on a hierarchical decomposition approach. No computational experiments or application to any practical case have been reported.

Perboli et al. (2011) were the first authors to formally define the term 2E-VRP and propose a mathematical model for the problem with a single deposit from which vehicles depart at the first level. Cuts derived from VRP formulations were proposed to improve the linear relaxation of the mathematical model. Two problem decomposition heuristics that seek to optimally determine the allocation of clients to satellite terminals - allowing the resolution of each of the scripting problems independently - based on the mathematical model (*math-based heuristics*) were also presented. Computational experiments comprised four sets of test instances with up to 50 customers and a single warehouse. The results showed that the *gap of* the mathematical model is small up to 32 customers, increasing in problems with 50 customers and particularly in those with four satellites. The two heuristics based on mixed-integer programming presented good performance both from the computational point of view and from the quality of the solutions obtained.

Jepsen et al. (2013) proposed *an arc-flow-based formulation for* 2E-VRP, while Baldacci et al. (2013) presented a route-based formulation. Both can resolve small problem instances. Marujo et al. (2018) proposed a method to identify the economic (on costs and service levels) and environmental impacts of using motorized cargo tricycles in combination with conventional trucks – acting as mobile depots - as a solution to address the constraints imposed on access of conventional cargo vehicles in densely populated areas. The analytical method in conjunction with Monte Carlo simulation for eco-efficiency evaluation was applied to the megacity of Rio de Janeiro. The results showed that the use of cargo tricycles that replenish from parked trucks serving as mobile warehouses does not result in a significant increase in the time or distance traveled or decrease in the level of service experienced by customers, while greenhouse gas emissions and local air quality pollutants can be significantly reduced using load tricycles and mobile deposits in the last mile distribution.

Li et al. (2021) addressed the problem of route optimization in a series of synchronized express service steps (line haul, pickup and delivery), in which satellite terminals are classified in two types: satellite terminals for cargo pickup, called source satellites, which serve only senders; and satellite terminals for cargo deliveries, so-called destination satellites, serve only the receivers. At the second level, a heterogeneous fleet of small-capacity trucks meets senders' requests and collects cargo for transport to a source satellite. Upon arriving at the destination satellite terminal – connected to a source satellite through the intercity line haul - the loads are transshipped and loaded into trucks scheduled to arrive on time to the recipients. In the first level, the loads are loaded in large capacity trucks and transported between the satellite terminals. To ensure the on time express services, synchronization constraints on the source satellites and destination satellites must be assured. The goal is to find the set of routes that minimizes the sum of costs. The authors proposed a mixed-integer programming model in which satellite bi-synchronization constraints provide an innovative method for formulating two-level network routing problems. A strategy was also developed based on the ALNS meta-heuristic (*"adaptive large neighborhood search"*) that uses different operators of destruction and repair of solutions. The experiments involved small (three source satellite terminals, three destination and up to 9 pickup or delivery points) and large (17 source and destination satellite terminals and 120 pickup and delivery points). The exact method was able to solve only part of the small instances, whereas ALNS was able to obtain quality solutions (deviations of less than 0.5% for those instances) in reduced processing times. In a related subsequent work, Li et al. (2022) addressed the 2E-CVRP with constraints on simultaneous grouping and simultaneous pickup and delivery, a new variant of the classic 2E-VRP problem in which customers from the same administrative region are served by vehicles from the same satellite, to ensure service consistency, and the pickup and delivery are carried out simultaneously at the second level. To optimally solve the problem, a path-based formulation was proposed, and a specialized branch-and-cut-and-price algorithm was developed, which also allows to solve two related variants: the 2E-VRP with grouping constraints and the 2E-VRP with simultaneous collection and delivery. Computational experiments with three groups of test instances were conducted to evaluate the performance of the algorithm proposed in the problems. The results show that the proposed dominance rule can significantly reduce the number of labels generated and all valid inequalities have a major impact on the robustness of the path-based model, making it competitive when compared to existing exact algorithms.

Liu and Jiang (2021) describe a variable neighborhood search (VNS) algorithm to solve the 2E-VRPSDP, a variant of the two-echelon vehicle routing problem with simultaneous pickups and deliveries in which both feasible and infeasible solutions can be explored. Arnold and Sorensen, 2019 demonstrate the importance of well-implemented local search and how they are sufficient to create heuristics to obtain high-quality solutions in short processing time.

Good reviews on the 2E-VRP can be found in Cuda et al. (2015) and Sluijk et al. (2022). The latter comprises 40 new articles that have been published in a relatively short time frame since the previous review; additionally, it includes variants of problems, inspired by real world applications, which have been studied to fill the gap between theoretical and canonical models and practical applications. The authors identified several challenging extensions of the 2E-VRP, including multiple warehouses, time windows, simultaneous pickup and delivery operations, heterogeneous fleet of vehicles at each of the two levels, and the inclusion of delivery alternatives such as direct warehouse deliveries, without going through any satellite, for customers located near the origin of the cargo. The synchronization of routes between the two levels and split deliveries at the first level are important aspects highlighted by the authors, especially for distribution systems for e-commerce. Finally, it is also important to highlight that the literature on the subject also comprises the multi-level distribution network design problem, which also considers decisions regarding the location of the facilities. Among them, in Winkenbach et al (2015), the total distance covered in the itineraries is made through a simplified analytical method derived from Daganzo (2004) to reduce the overall complexity of the problem. The work of Janjevic et al. (2021) is a good reference on the state-of-the-art models of localization and routing in multi-layered networks.

3 PROPOSED MODEL

The proposed model comprises a math-heuristic for solving the 2E-CVRP by applying a “cluster-first route-second” method. With this approach, customers are first allocated to the satellites, thus decomposing the problem into several Capacitated VRPs. Two CVRP problems are solved after the clustering process (which defines the customers and load assigned to each satellite): the capacitated vehicle routing problem from the central depot to the satellites and the capacitated routing problem within each satellite. The math-heuristic steps (Figure 2) are:

- Step 1: builds an initial cluster of demand nodes around the existing satellites respecting the existing capacity constraints.
- Step 2: solves the first echelon CVRP based on the clusters obtained in Step 1.
- Step 3: solves the second echelon CVRP from clusters formed in Step 1.
- Step 4: and Step 5 solve the 2E-CVRP by grouping all demand nodes in each single satellite if capacity constraints allow. These two steps are intended to tackle scenarios where the optimal solution is located within a single satellite, and Step 1 fails to detect a unique cluster..
- Steps 6 and 7 aim to determine the solution with the lowest cost among the results obtained in Step 3 for each individual satellite, as well as the results obtained in Step 5 (FE and SE) for each single satellite
- In Step 8, the chosen solution from Step 7 is subjected to the Simulated Annealing metaheuristic with the aim to improve it...

3.1 Math-heuristic model

The model comprises eight steps and aims to solve the 2E-CVRP using a “cluster-first, route-second” approach. Figure 2 depicts all the steps of the math-heuristic, while its variables and parameters are defined in Table 1.

Table 1 – Math-heuristic parameters and variables.

Variable	Definition	From	Input to
C	Set of demand nodes.	problem	All steps
S	Set of satellites.	problem	All steps
$\mathrm{h_s}$	Handling cost at satellite s.	problem	Step 2
$\mathrm{m_s}$	Maximum number of city freighters per satellite.	problem	Step 3
$\mathrm{d_{ij}}$	Distance (or cost) to cover the arc from i to j.	problem	All steps
$\mathrm{x_{ij}}$	Binary decision variable, xij = 1 if demand node i is assigned to satellite cluster j, 0 otherwise.	Step 1	Step 2
$\mathrm{city\_freighter_s}$	Number of city freighters required per each satellite cluster s$\in S$.	Step 1	-
$\mathrm{z_j}$	Binary decision variable, $\mathrm{z_j}$ = 1 if satellite j has 1 or more demand nodes assigned to its cluster (satellite j is active), 0 otherwise.	Step 1	Step 2
$\mathrm{load\_depot_s}$	Load assigned to depot s$\in S$.	Step 1	Step 2
$\mathrm{demand_i}$	Demand required by i.	problem	Steps 1,3
M1	Maximum number of vehicles available in the first echelon.	problem	Step 2
M2	Maximum number of vehicles available in the second echelon.	problem	Step 3
A1	$A1$=$\{(i,j) \mid i,j \in \{0\} \cup S,\ i \neq j\}$ = set of the first echelon arcs connecting the main depot $\{0\}$ to satellite $j \in S$.	problem	Step 2
A2	$A2 = \{(i,j) \mid i,j \in S \cup C,\ i \neq j\}$ = set of the second echelon arcs, connecting satellite $i \in S$ with demand node $j \in C$.	problem	Step 3
K1	Set of first echelon vehicles	problem	Step 2
K2	Set of second echelon vehicles	problem	Step 3
Q1	Capacity of first echelon vehicles	problem	Step 2
Q2	Capacity of second echelon vehicles	problem	Steps 1,3
$\mathrm{x_{ijk}}$	Binary decision variable, x_{ijk} = 1 indicates if vehicle $k \in K1$ uses arc $(i,j) \in A1$, 0 otherwise.	Step 2	Step 6
$\mathrm{u_{sk}}$	Subtour elimination variable in Step 1	-	-
$\mathrm{w_{sk}}$	Load transported from main depot $\{0\}$ to satellite $s \in S$ by vehicle $k \in K1$.	Step 2	Step 6
$\mathrm{y_{ijs}}$	Binary variable, $y_{ijs} = 1$ if vehicle dispatched from $s \in S$ uses arc$(i,j) \in A2$, 0 otherwise.	-	Step 6
$\mathrm{f_{ijs}}$	Load transported from s$\in S$ using arc$(i,j) \in A2$.	-	Step 6

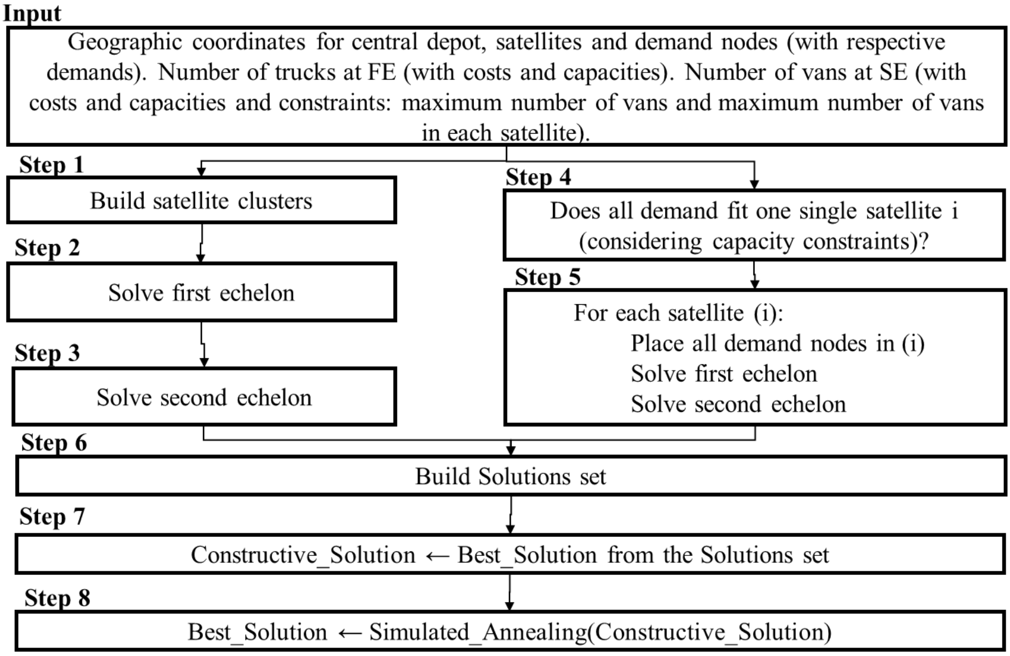

Input
Geographic coordinates for central depot, satellites and demand nodes (with respective demands). Number of trucks at FE (with costs and capacities). Number of vans at SE (with costs and capacities and constraints: maximum number of vans and maximum number of vans in each satellite).
Step 1
Build satellite clusters
Step 2
Solve first echelon
Step 3
Solve second echelon
Step 4
Does all demand fit one single satellite i (considering capacity constraints)?
Step 5
For each satellite (i):
Place all demand nodes in (i)
Solve first echelon
Solve second echelon
Step 6
Build Solutions set
Step 7
Constructive_Solution ← Best_Solution from the Solutions set
Step 8
Best_Solution ← Simulated_Annealing(Constructive_Solution)

Figure 2 – Description of proposed model.

3.1.1 Step 1 – Build initial satellite clusters

This step attempts to cluster the demand nodes around each satellite by applying a k-medoids approach (Kaufman and Rousseeuw, 2005). It will split the demand nodes into s satellite clusters, where the number s is the number of satellites. Step 1 consists of an implementation in CPLEX solver of the following mathematical model:

$$Objective\ Function = \ min \ \sum_{i \in C} \sum_{j \in S} d_{ij}^2 * x_{ij} \tag{1}$$

Model constraints:

$$\sum_{j \in S} x_{ij} = 1, \ \forall \ i \in C \tag{2}$$

$$\sum_{i \in C} demand_i * \ x_{is} \leq Q2 * \ m_s, \ \forall s \in S \tag{3}$$

$$city\_freighter_s - 1 \leq \frac{(\sum_{i \in C} demand_i \ * \ x_{is})}{Q2}, \forall s \in S \tag{4}$$

$$\frac{(\sum_{i \in C} demand_i * \ x_{is})}{Q2} \leq \ city\_freighter_s \ , \ \forall s \in S \tag{5}$$

$$\sum_{s \in S} city\_freighter_s \leq M2 \tag{6}$$

$$x_{ij} \leq z_j, \; \forall\, i \in C, \;\; \forall\, j \in S \tag{7}$$

$$z_j \in \{0,1\} \;, \quad \forall\, j \in S \tag{8}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, i \in C, \;\; \forall\, j \in S \tag{9}$$

$$city\_freighter_s \in \mathbb{N}, \;\; \forall\, s \in S \tag{10}$$

The objective function (1) minimizes the sum of the square of the distances from each demand node to the satellites (the square of the distance is used to increase the penalty and reduce the sensibility of the formulation when applied on the solver if a node has approximately the same distance from the satellites). Constraints (2) ensure that all demand nodes must be assigned to only one satellite cluster, while constraints (3) impose that the total satellite cluster demand must respect satellite capacity. Constraints (4), (5) and (6) bound the total number of city freighters at the second echelon. Constraint (7) states that a satellite cluster j is opened if there is at least one demand node assigned to it. Expressions (8), (9) and (10) correspond to the non-negativity constraints.

3.1.2 Step 2 – Solve the CVRP for the first echelon

The objective of this step is to solve the first echelon routes - given the demand clusters around the satellites from Step 1 - so that the satellites can be loaded from the main depot with the required demand at minimum cost. The formulation of Step 2 is based on Sluijk et al. (2022), being the main difference the inclusion on the constraint (18) to allow finding a solution to the FE before the SE is solved by constraining the total demand in each satellite according to the clusters obtained in Step 1. Step 2 consists of an implementation in CPLEX solver of the following mathematical model:

$$Objective\,function = \; min \sum_{k \in K1} \sum_{(i,j) \in A1} d_{ij} x_{ijk} + \sum_{k \in K1} \sum_{s \in S} h_s * w_{sk} \tag{11}$$

Model constraints:

$$\sum_{k \in K1} \sum_{i \in \{S\}} x_{0ik} \leq M1, \;\; \forall\, i \in S \tag{12}$$

$$\sum_{(i,j) \in A1} x_{ijk} = \sum_{(i,j) \in A1} x_{jik}, \; \forall\, i \; \in \{0\} \cup S, \; \forall\, k \in K1 \tag{13}$$

$$\sum_{(i,j) \in A1} x_{ijk} \leq 1, \; \forall\, i \; \in \{0\} \cup S, \; \forall\, k \in K1 \tag{14}$$

$$u_{ik} - \; u_{jk} + \; |S| \, x_{ijk} \leq |S| - 1, \; \forall\, i,j \in S, \; i \; \neq j, \; \forall\, k \in K1 \tag{15}$$

$$w_{sk} \leq Q1 \sum_{(j,s) \in A1} x_{jsk}, \; \forall\, s \in S, \;\; \forall\, k \in K1 \tag{16}$$

$$\sum_{s \in S} w_{sk} \leq Q1, \; \forall\, k \in K1 \tag{17}$$

$$\sum_{k\in K1} w_{sk} = load\_\ depot_s,\ \forall\ s \in opened\ satellites \tag{18}$$

$$u_{sk} \in Z^+, \quad \forall\ s \in S,\ \forall\ k \in K1 \tag{19}$$

$$w_{sk} \geq 0, \quad \forall\ s \in S,\ \forall\ k \in K1 \tag{20}$$

$$x_{ijk} \in \{0,1\}, \quad \forall\ (i,j) \in A1,\ \forall k \in K1 \tag{21}$$

The objective function (11) aims to minimize the route costs from the main depot to the satellites and the handling costs. Constraints (12) limit the number of available vehicles at the first echelon. Constraints (13) assure the flow of conservation at the first echelon. Constraints (14) impose that each satellite can be visited at most once by a first echelon vehicle. Constraints (15) are the subtour elimination constraints. Constraints (16) and (17) establish the first echelon vehicles capacity constraints, while constraints (18) assure that Step 1 cluster demands must be respected. Equations (19), (20) and (21) are the non-negativity constraints.

3.1.3 Step 3 – Solve the CVRP for the second echelon

The objective of this step is to solve the second echelon routes - given the demand clusters around the satellites from Step 1 - so that the demand nodes can be serviced from each satellite at minimum cost. The formulation of Step 3 is based on Sluijk et al. (2022) being the main difference the inclusion on the constraint (23) to allow finding a solution to the SE after the FE is solved by constraining the total demand in each satellite according to the clusters obtained in Step 1. Step 3 consists of an implementation in CPLEX of the following mathematical model (this step solves the second echelon as a monolithic block):

$$Objective\ function = \ min \sum_{s\in S} \sum_{(i,j)\in A2} d_{ij} y_{ijs} \tag{22}$$

Model constraints:

$$\sum_{c\in C} f_{scs} = load\_depot_s\ ,\ \forall\ s \in S \tag{23}$$

$$f_{sjs_2} = 0,\ \forall\ (s,j) \in A2,\ \forall s,\ s_2 \in S, s_{\ 2} \neq s \tag{24}$$

$$y_{ijs_2} = 0,\ f_{ijs_2} = 0\ ,\ \forall (i,j) \in A2,\ \forall s_2 \in S,\ s_2 \neq s,\ \forall s \in current\ opened\ satellites, \forall\ i\ and\ \ j\ assigned\ to\ s \tag{25}$$

$$\sum_{(i,j)\in A2} y_{ijs} = \sum_{(i,j)\in A2} y_{jis},\ \forall\ i\ \in S \cup C,\ \ \forall\ s \in S \tag{26}$$

$$\sum_{(s,j)\in A2} y_{sjs} \leq \ m_s,\ \ \forall\ s \in S \tag{27}$$

$$\sum_{(s,j)\in A2} y_{sjs} \leq \ M2 \tag{28}$$

$$\sum_{s\in S} \sum_{(i,j)\in A2} y_{ijs} = 1,\ \forall\ j \in C \tag{29}$$

$$\sum_{s\in S}\sum_{(j,i)\in A2} f_{jis} - \sum_{s\in S}\sum_{(i,j)\in A2} f_{ijs} = demand_i, \ \forall\, i \in C \tag{30}$$

$$f_{ijs} \leq Q2\, y_{ijs}\ , \quad \forall\,(i,j)\in A2, \forall\, s\in S \tag{31}$$

$$f_{ijs} \geq 0, \quad \forall\,(i,j)\in A2, \ \forall\, s\in S \tag{32}$$

$$x_{ijk} \in \{0,1\}, \quad \forall\,(i,j)\in A1, \ \forall k\in K1 \tag{33}$$

$$y_{ijs} \in \{0,1\}, \quad \forall\,(i,j)\in A2, \ \forall s\in S \tag{34}$$

The objective function (22) aims to minimize the route costs from the satellites to the demand nodes. Constraints (23) assure that the total load from each satellite must respect the satellite cluster load from Step 1. Constraints (24) and (25) impose that there is no flow between satellites. Constraints (26) assure the flow conservation in the second echelon. Constraints (27) limit the number of vehicles per satellite. Constraints (28) limit the number of vehicles in the second echelon. Constraints (29) assure that each demand node is visited exactly once. Constraints (30) establish that the net flow at each demand node is equal to its demand. Constraints (31) assure that the flow from each satellite must comply with the second echelon vehicle capacity constraints. Equations (32), (33) and (34) are the non-negativity constraints.

3.1.4 Steps 4 and 5 – Group demand in single cluster

In Step 4, if capacity constraints allow – second echelon freighters capacity, total number of second echelon freighters available, number of freighters allowed per satellite and satellite capacity – the demand is assigned for every single satellite. Then Step 5 applies Step 2 and Step 3 for every single cluster identified in Step 4. These two steps are designed to address the situation where the optimum solution resides within a single satellite and Step 1 does not find a single cluster.

3.1.5 Steps 6 and 7 – Build solution set and obtain initial feasible solution

Step 6 and Step 7 identify the minimum cost solution between the solution (FE and SE) obtained in Step 3 and each solution (FE and SE) obtained in Step 5 for each different satellite.

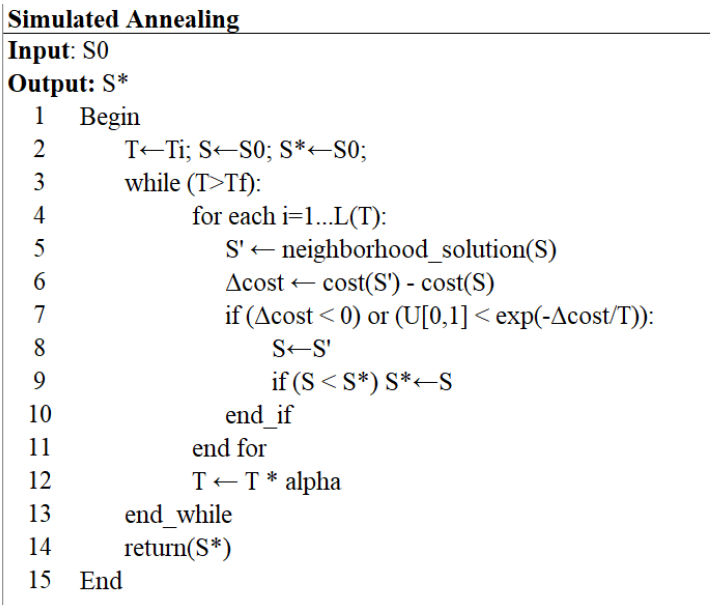
3.1.6 Step 8 – Simulated Annealing

This step executes the Simulated Annealing (SA) metaheuristic to the chosen solution in Step 7 as detailed in the pseudocode depicted in Figure 3.

S0 is the initial feasible solution obtained from Step 7. S* is the best solution and S is the incumbent solution during the SA iterations. T is the temperature parameter and Ti the initial process temperature. L(T) is the SA iterations parameter and alpha the temperature reduction rate. The neighborhood solution (S') is obtained by either two local search operators applied on the incumbent solution S: move or swap.

The move operator relocates a demand node from its cluster to another different cluster if capacity constraints allow (total number of vehicles available and vehicle capacities at the second echelon, maximum number of vehicles per satellite and satellite capacity).

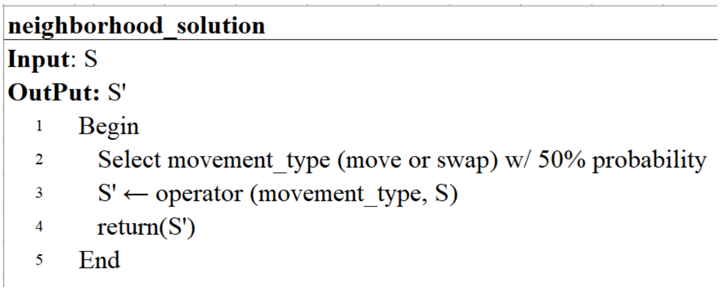
The swap operator exchanges the satellites of two demand nodes in different satellites if capacity constraints allow (total number of vehicles available at the second echelon, maximum number of vehicles per satellite and satellite capacity).



```
Simulated Annealing
Input: S0
Output: S*
Begin
    T←Ti; S←S0; S*←S0;
    while (T>Tf):
        for each i=1...L(T):
            S' ← neighborhood_solution(S)
            Δcost ← cost(S') - cost(S)
            if (Δcost < 0) or (U[0,1] < exp(-Δcost/T)):
                S←S'
                if (S < S*) S*←S
            end_if
        end for
        T ← T * alpha
    end_while
    return(S*)
End
```

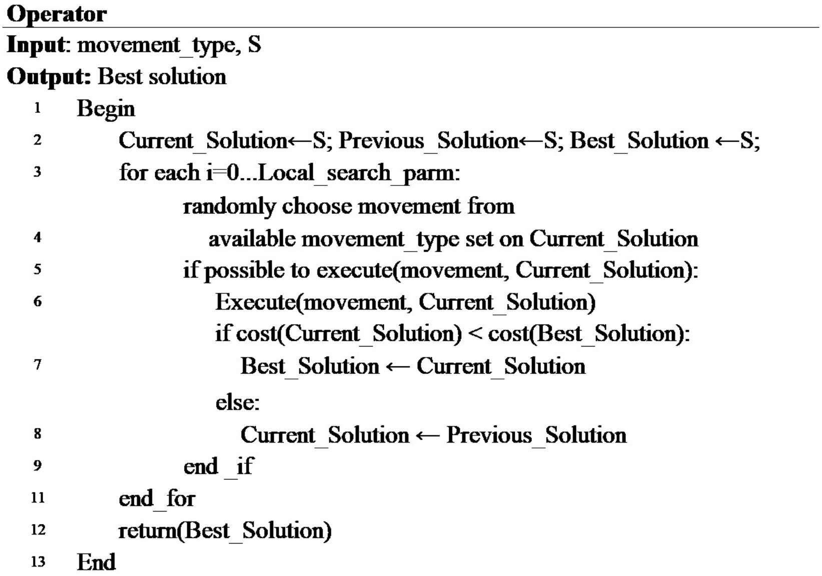
Figure 3 – Simulated Annealing algorithm.

Operators move and swap are selected randomly with probability of 50% as described by the pseudocode in Figure 4. The local search operators (move or swap) perform as described by the pseudocode in Figure 5.



```
neighborhood_solution
Input: S
OutPut: S'
Begin
  Select movement_type (move or swap) w/ 50% probability
  S' ← operator (movement_type, S)
  return(S')
End
```

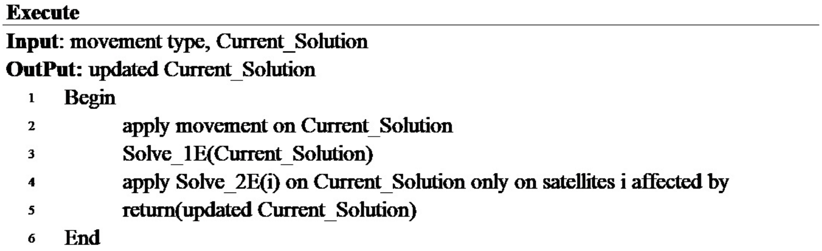
Figure 4 – Local search operator selection.



```
Operator
Input: movement_type, S
Output: Best solution
Begin
    Current_Solution←S; Previous_Solution←S; Best_Solution ←S;
    for each i=0...Local_search_parm:
        randomly choose movement from
            available movement_type set on Current_Solution
        if possible to execute(movement, Current_Solution):
            Execute(movement, Current_Solution)
            if cost(Current_Solution) < cost(Best_Solution):
                Best_Solution ← Current_Solution
            else:
                Current_Solution ← Previous_Solution
        end _if
    end_for
    return(Best_Solution)
End
```

Figure 5 – Local search.

Inside the local search routine, the new solution - after either the move or swap operations - is obtained by applying Step 2 (3.2.2) and Step 3 (3.2.3) as shown by the pseudocode presented by Step 3 and Step 4 in Figure 6.



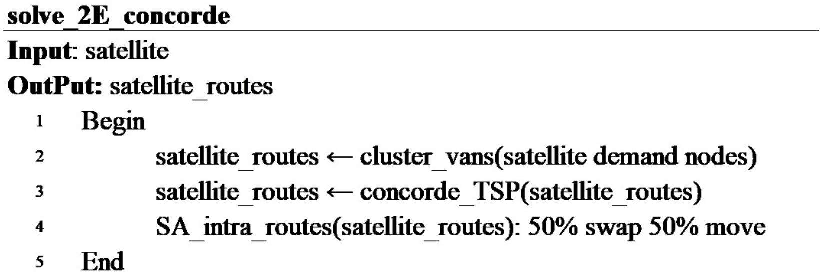
```
Execute
Input: movement type, Current_Solution
OutPut: updated Current_Solution
1  Begin
2          apply movement on Current_Solution
3          Solve_1E(Current_Solution)
4          apply Solve_2E(i) on Current_Solution only on satellites i affected by
5          return(updated Current_Solution)
6  End
```

Figure 6 – Obtain new solution after local search operators.

3.2 Second echelon math-heuristic variation

An alternative math-heuristic to solve the second echelon (Step 3) was also developed using the TSP Concorde algorithm libraries (Applegate et al., 2003) as described by the pseudocode in Figure 7.

In this math-heuristic, set V corresponds to the set of satellite vehicles, whose cardinality |V| is equal to the round up to the nearest integer of the division of the total satellite demand by the



```
solve_2E_concorde
Input: satellite
OutPut: satellite_routes
Begin
        satellite_routes ← cluster_vans(satellite demand nodes)
        satellite_routes ← concorde_TSP(satellite_routes)
        SA_intra_routes(satellite_routes): 50% swap 50% move
End
```

Figure 7 – Alternative math-heuristic for Step 3.

freighter capacity parameter. The mathematical modeling of the *cluster_vans* step of the math-heuristic is:

$$Objective\ Function = \ min\ \sum_{j \in V} z_j \tag{35}$$

Model constraints:

$$\sum_{j \in V} x_{ij} = 1,\ \forall\ i \in C \tag{36}$$

$$x_{ij} \leq z_j,\ \forall\ i \in C,\ \forall\ i \in S \tag{37}$$

$$\sum_{i \in C} demand_i * x_{ij} \leq freighter\_\ capacity,\ \forall\ j \in V \tag{38}$$

$$z_j \in \{0,1\}\ ,\quad \forall\ j \in S \tag{39}$$

$$x_{ij} \in \{0,1\},\ \ \forall\ i \in C\ ,\ \forall\ j \in V \tag{40}$$

The objective function (35) aims to cluster the demand nodes in the minimum required set of vehicles. Constraints (36) state that each demand node must be allocated to a single second echelon vehicle. Constraints (37) assure that a route (vehicle) cluster j is opened if there is at least one demand node assigned to it. Constraints (38) impose that the total load allocated to a vehicle must be within freighter capacity. Equations (39) and (40) are the non-negativity constraints. Once the demand nodes are assigned to routes - the vehicle clusters are identified - the initial routes are obtained by applying the TSP Concorde algorithm to each vehicle cluster. Finally, within each satellite cluster a Simulated Annealing process as described in 3.2.6 is applied between the routes (or vehicles): demand nodes are moved or swapped between the same satellite cluster routes with 50% probability.

4 COMPUTATIONAL EXPERIMENTS

The proposed models were implemented in C++ using CPLEX 22.1.0. The second echelon math-heuristic variation (section 3.3) employed Concorde TSP C libraries (Applegate et al., 2003). The math-heuristics were tested with instances used in Sluijk et al. (2022) and Voigt et al. (2022) as given in https://homepage.univie.ac.at/ulrich.breunig/research/. The best-known solutions (column BKS), in all the result tables presented below, marked with an asterisk are optimal solutions. Non marked best-known solutions have not been proven optimal (Breunig et al., 2016).

The chosen instances can be considered small - up to 100 customers - and medium - from 101 to 350 customers (Alesiani et al., 2022). The problem instances were solved using an Intel i7 computer with 32GB of memory running Linux operating system. Steps 1-3 and 5 correspond to mathematical models that were implemented in CPLEX as described respectively above in subsections 3.1.1, 3.1.2 and 3.1.3, respectively.

An exact model, solving concurrently the first and second echelon, implemented in Gurobi, using the mathematical formulation presented in Sluijk et al. (2022) was developed to certify the implemented math-heuristic model– since it uses part of the original formulation – and to check the processing time of the math-heuristic when compared to the exact monolithic approach. Table 2 and Table 3 present the results for the exact model. The solver time limit was set to 36,000 seconds (10 hours). Column 'Qty Sat' contains the number of satellites in the instance; column 'Qty Clt' the number of demand nodes; column 'Dmd' the total second echelon demand; column 'Qty Tr' the number of first echelon vehicles; column 'Truck Cap' the capacity of each first echelon vehicle; column 'Max Van/Sat' the maximum number of second echelon vehicles per satellite; columns 'Qty Vans' the total number of second echelon vehicles; column 'Van Cap' the capacity of each second echelon vehicle; column 'BKS' the cost of the best-known solution for the test instance; column 'Sol' the cost obtained from the exact model; column runtime (solver execution time in seconds); column 'Gap %' the solver gap after the model execution (or execution time limit being reached) and column deviation the percentage difference from the cost obtained from the exact model and the best-known solution for the instance.

The math-heuristic test instances results are presented in: Table 4 (test instances from Set#2), in Table 5 (test instances from Set#3), Table 6 (test instances from Set#4), Table 7 (test instances from Set#5) and Table 8 (test instances from Set#6). The results presented are an average of five different executions per set of test instances. Column 'Qty Sat' contains the number of satellites in the instance; column 'Qty Clt' the number of demand nodes; column 'Dmd' the total second echelon demand; column 'Qty Tr' the number of first echelon vehicles; column 'Truck Cap' the capacity of each first echelon vehicle; column 'Max Van/Sat' the maximum number of second echelon vehicles per satellite; columns 'Qty Vans' the total number of second echelon vehicles; column 'Van Cap' the capacity of each second echelon vehicle; column 'BKS' the cost of the best-known solution for the test instance; column 'SA' the cost obtained from the math-heuristic; column runtime (running time in seconds); column 'deviation' the percentage difference from the cost obtained from the math-heuristic and the best-known solution for the instance. Instance sets named Set#2, Set#3 and Set#4 were solved using the math-heuristic for the second echelon as described in 3.2.3.

The following parameters were used in the test instances to stay within the computational time comparable to the methods analyzed in Sluijk et al. (2022): solver gap limit equal 0.005% and solver time limit = 600 seconds, local search iterations limit L(T) = 10 and simulated annealing iterations limit = 10,simulated annealing initial (Ti) temperature = 1000 and final temperature (Tf) = 0.01, temperature reduction ratio (alpha)=0.9 and simulated annealing time limit = 1800 seconds. The size of the test instances contained in datasets Set#5 and Set#6 required the use of the math-heuristic described in 3.3 since no feasible quality solution was obtained after three hours of processing with the second echelon math-heuristic described in 3.2.2. The following parameters were used for instance sets Set#5 and Set#6 to ensure the computational time comparable to the methods analyzed in Sluijk et al., 2022: solver gap limit = 0.005% and solver time limit = 600 seconds, local search iterations limit = 10 and simulated annealing iterations limit = 10, simulated annealing initial temperature = 1000 and final temperature = 0.01, temperature reduction ration (alpha) = 0.9, simulated annealing process time limit = 3,600 seconds. The used intra-route parameters are local search iterations limit = 50 and simulated annealing iterations limit=50, simulated annealing initial temperature=1,000 and final temperature= 0.01, temperature reduction ration (alpha) = 0.9, simulated annealing process time limit = 300 seconds. Table 9 presents a summary of the results obtained for all the test instances. Column ‘Qty Sat.’ denotes the number of satellites in the instance; column ‘Qty. Clt’ the number of demand nodes; columns delta the difference from the cost obtained from the math-heuristic and the best-known solution for the instance. Finally, columns ‘t(s)’ correspond to the execution time in seconds.

Table 2 – Results obtained by the exact model using the formulation by Sluijk et al. (2022).

	Qty	Qty		Qty	Truck	Max	Qty				run	Gap	Dev
Set #2	Sat	Clt.	Dmd	Tr	Cap	Van/Sat	Vans	Van Cap	BKS	Sol.	time(s)	%	%
Set2a_E-n22-k4-s6-17.dat	2	21	22500	3	15000	4	4	6000	417.07*	417.07	8	0.0	0.0
Set2a_E-n22-k4-s8-14.dat	2	21	22500	3	15000	4	4	6000	384.96*	384.96	23	0.0	0.0
Set2a_E-n22-k4-s9-19.dat	2	21	22500	3	15000	4	4	6000	470.60*	470.60	187	0.0	0.0
Set2a_E-n22-k4-s10-14.dat	2	21	22500	3	15000	4	4	6000	371.50*	371.50	22	0.0	0.0
Set2a_E-n22-k4-s11-12.dat	2	21	22500	3	15000	4	4	6000	427.22*	427.22	102	0.0	0.0
Set2a_E-n22-k4-s12-16.dat	2	21	22500	3	15000	4	4	6000	392.78*	392.78	272	0.0	0.0
Set2a_E-n33-k4-s1-9.dat	2	32	29370	3	20000	4	4	8000	730.16*	754.67	1240	0.0	0.0
Set2a_E-n33-k4-s2-13.dat	2	32	29370	3	20000	4	4	8000	714.63*	745.38	3600	0.0	0.0
Set2a_E-n33-k4-s3-17.dat	2	32	29370	3	20000	4	4	8000	707.48*	762.15	1577	0.0	0.0
Set2a_E-n33-k4-s4-5.dat	2	32	29370	3	20000	4	4	8000	778.74*	778.74	3600	0.0	0.0
Set2a_E-n33-k4-s7-25.dat	2	32	29370	3	20000	4	4	8000	756.85*	756.85	3600	0.0	0.0
Set2a_E-n33-k4-s14-22.dat	2	32	29370	3	20000	4	4	8000	779.05*	779.05	732	0.0	0.0
Set2c_E-n51-k5-s2-17.dat	2	50	777	4	400	5	5	160	601.39	612.75	36021	4.06	1.9
Set2c_E-n51-k5-s4-46.dat	2	50	777	4	400	5	5	160	702.33	702.33	36010	4.50	0.0
Set2c_E-n51-k5-s6-12.dat	2	50	777	4	400	5	5	160	567.42	567.42	36011	1.34	0.0
Set2c_E-n51 -k5-s11-19.dat	2	50	777	4	400	5	5	160	617.42	627.99	36022	3.64	1.7
Set2c_E-n51-k5-s27-47.dat	2	50	777	4	400	5	5	160	530.76	530.75	10052	0.00	0.0
Set2c_E-n51-k5-s32-37.dat	2	50	777	4	400	5	5	160	752.59	805.44	36010	5.93	7.0
Set2c_E-n51-k5-s2-4-17-46.dat	4	50	777	4	400	5	5	160	601.39	644.35	36017	6.18	7.1
Set2c_E-n51 -k5-s6-12-32-37.dat	4	50	777	4	400	5	5	160	567.42	567.42	36014	2.09	0.0
Set2c_E-n51-k5-s11-19-27-47.dat	4	50	777	4	400	5	5	160	530.76	530.76	36016	1.04	0.0

Table 3 – Results obtained by the exact model with Sluijk et al. (2022) formulation.

Set #3	Qty Sat	Qty Clt	Dmd	Qty. Tr	Truck Cap	Max Van/Sat	Qty Vans	Van Cap	BKS	Sol.	run time (s)	Gap %	Dev %
Set3_E-n22-k4-s13-14.dat	2	21	22500	3	15000	4	4	6000	526.15*	526.15	131	0,00	0.0
Set3_E-n22-k4-s13-16.dat	2	21	22500	3	15000	4	4	6000	521.09*	521.09	149	0,00	0.0
Set3_E-n22-k4-s13-17.dat	2	21	22500	3	15000	4	4	6000	496.38*	496.38	101	0,00	0.0
Set3_E-n22-k4-s14-19.dat	2	21	22500	3	15000	4	4	6000	497.80*	497.80	557	0,00	0.0
Set3_E-n22-k4-s17-19.dat	2	21	22500	3	15000	4	4	6000	512.80*	512.80	940	0,00	0.0
Set3_E-n22-k4-s19-21.dat	2	21	22500	3	15000	4	4	6000	520.42*	520.42	867	0,00	0.0
Set3_E-n33-k4-s16-22.dat	2	32	29370	3	20000	4	4	8000	672.17*	672.17	3600	0,10	0.0
Set3_E-n33-k4-s16-24.dat	2	32	29370	3	20000	4	4	8000	666.02*	666.02	3600	0,00	0.0
Set3_E-n33-k4-s19-26.dat	2	32	29370	3	20000	4	4	8000	680.36*	680.36	3600	0,04	0.0
Set3_E-n33-k4-s22-26.dat	2	32	29370	3	20000	4	4	8000	680.36*	680.36	3600	0,10	0.0
Set3_E-n33-k4-s24-28.dat	2	32	29370	3	20000	4	4	8000	670.43*	670.43	3600	0,07	0.0
Set3_E-n33-k4-s25-28.dat	2	32	29370	3	20000	4	4	8000	650.58*	650.58	3600	0,05	0.0
Set3_E-n51-k5-s12-18.dat	2	50	777	3	400	5	5	160	690.59	690.59	36002	1,63	0.0
Set3_E-n51-k5-s12-41.dat	2	50	777	3	400	5	5	160	683.05	689.81	36011	2,86	1.0
Set3_E-n51-k5-s12-43.dat	2	50	777	3	400	5	5	160	710.41	710.41	36017	0,80	0.0
Set3_E-n51-k5-s39-41.dat	2	50	777	3	400	5	5	160	728.54	751.73	36008	4,58	3.2
Set3_E-n51-k5-s40-41.dat	2	50	777	3	400	5	5	160	723.75	735.71	36001	3,95	1.7
Set3_E-n51-k5-s40-43.dat	2	50	777	3	400	5	5	160	752.15	757.77	36002	3,04	0.7

Table 4 – Set#2 Results obtained by the math-heuristic.

Set #2	Qty Sat	Qty Clt.	Dmd	Qty Tr.	Truck Cap	Max Van/Sat	Qty Vans	Van Cap	BKS	SA	Run time (s)	Dev %
Set2a_E-n22-k4-s6-17.dat	2	21	22500	3	15000	4	4	6000	417.07*	417.07	864	0.0
Set2a_E-n22-k4-s8-14.dat	2	21	22500	3	15000	4	4	6000	384.96*	384.96	1087	0.0
Set2a_E-n22-k4-s9-19.dat	2	21	22500	3	15000	4	4	6000	470.60*	470.60	1167	0.0
Set2a_E-n22-k4-s10-14.dat	2	21	22500	3	15000	4	4	6000	371.50*	371.50	966	0.0
Set2a_E-n22-k4-s11-12.dat	2	21	22500	3	15000	4	4	6000	427.22*	429.39	15	0.5
Set2a_E-n22-k4-s12-16.dat	2	21	22500	3	15000	4	4	6000	392.78*	422.39	5	7.5
Set2a_E-n33-k4-s1-9.dat	2	32	29370	3	20000	4	4	8000	730.16*	754.67	437	3.4
Set2a_E-n33-k4-s2-13.dat	2	32	29370	3	20000	4	4	8000	714.63*	745.38	164	4.3
Set2a_E-n33-k4-s3-17.dat	2	32	29370	3	20000	4	4	8000	707.48*	762.15	372	7.7
Set2a_E-n33-k4-s4-5.dat	2	32	29370	3	20000	4	4	8000	778.74*	778.74	347	0.0
Set2a_E-n33-k4-s7-25.dat	2	32	29370	3	20000	4	4	8000	756.85*	756.85	2260	0.0
Set2a_E-n33-k4-s14-22.dat	2	32	29370	3	20000	4	4	8000	779.05*	779.05	371	0.0
Set2b_E-n51-k5-s2-17.dat	2	50	777	4	400	5	5	160	597.49*	630.60	1214	5.5
Set2b_E-n51-k5-s4-46.dat	2	50	777	4	400	5	5	160	530.76*	530.76	930	0.0
Set2b_E-n51-k5-s6-12.dat	2	50	777	4	400	5	5	160	554.81*	559.28	3768	0.8
Set2b_E-n51 -k5-s11-19.dat	2	50	777	4	400	5	5	160	581.64*	585.95	1261	0.7
Set2b_E-n51-k5-s27-47.dat	2	50	777	4	400	5	5	160	538.22*	538.22	3030	0.0
Set2b_E-n51-k5-s32-37.dat	2	50	777	4	400	5	5	160	552.28*	552.28	3340	0.0
Set2c_E-n51-k5-s2-17.dat	4	50	777	4	400	5	5	160	601.39	607.73	3054	1.1
Set2c_E-n51-k5-s4-46.dat	4	50	777	4	400	5	5	160	702.33	702.34	3468	0.0
Set2c_E-n51-k5-s6-12.dat	4	50	777	4	400	5	5	160	567.42	586.21	1230	3.3
Set2c_E-n51 -k5-s11-19.dat	2	50	777	4	400	5	5	160	617.42	626.50	1216	1.5
Set2c_E-n51-k5-s27-47.dat	2	50	777	4	400	5	5	160	530.76	530.76	1169	0.0
Set2c_E-n51-k5-s32-37.dat	2	50	777	4	400	5	5	160	752.59	762.54	1800	1.3
Set2b_E-n51-k5-s2-4-17-46.dat	2	50	777	4	400	5	5	160	530.76*	530.76	2114	0.0
Set2b_E-n51 -k5-s6-12-32-37.dat	2	50	777	4	400	5	5	160	531.92*	564.45	2401	6.1
Set2b_E-n51-k5-s11-19-27-47.dat	2	50	777	4	400	5	5	160	527.63*	564.62	2401	7.0
Set2c_E-n51-k5-s2-4-17-46.dat	4	50	777	4	400	5	5	160	601.39	617.66	2404	2.7
Set2c_E-n51 -k5-s6-12-32-37.dat	4	50	777	4	400	5	5	160	567.42	584.90	2404	3.1
Set2c_E-n51-k5-s11-19-27-47.dat	4	50	777	4	400	5	5	160	530.76	530.76	2101	0.0

Table 5 – Set#3 Results obtained by the math-heuristic.

Set #3	Qty Sat	Qty Clt.	Dmd	Qty. Tr.	Truck Cap	Max Van/Sat	Qty Vans	Van Cap	BKS	SA	Run time (s)	Dev %
Set3_E-n22-k4-s13-14.dat	2	21	22500	3	15000	4	4	6000	526.15*	536.32	7	1.93
Set3_E-n22-k4-s13-16.dat	2	21	22500	3	15000	4	4	6000	521.09*	522.54	12	0.3
Set3_E-n22-k4-s13-17.dat	2	21	22500	3	15000	4	4	6000	496.38*	496.38	1468	0.00
Set3_E-n22-k4-s14-19.dat	2	21	22500	3	15000	4	4	6000	498.80*	530.61	98	6.59
Set3_E-n22-k4-s17-19.dat	2	21	22500	3	15000	4	4	6000	512.80*	530.61	48	3.47
Set3_E-n22-k4-s19-21.dat	2	21	22500	3	15000	4	4	6000	520.42*	527.58	1994	1.38
Set3_E-n33-k4-s16-22.dat	2	32	29370	3	20000	4	4	8000	672.17*	703.32	1258	4.63
Set3_E-n33-k4-s16-24.dat	2	32	29370	3	20000	4	4	8000	666.02*	703.32	1027	5.60
Set3_E-n33-k4-s19-26.dat	2	32	29370	3	20000	4	4	8000	680.36*	680.37	455	0.00
Set3_E-n33-k4-s22-26.dat	2	32	29370	3	20000	4	4	8000	680.36*	680.37	835	0.00
Set3_E-n33-k4-s24-28.dat	2	32	29370	3	20000	4	4	8000	670.43*	690.11	1203	2.94
Set3_E-n33-k4-s25-28.dat	2	32	29370	3	20000	4	4	8000	650.58*	690.11	792	6.08
Set3_E-n51-k5-s12-18.dat	2	50	777	3	400	5	5	160	690.59	697.45	3785	0.99
Set3_E-n51-k5-s12-41.dat	2	50	777	3	400	5	5	160	683.05	707.96	6358	3.65
Set3_E-n51-k5-s12-43.dat	2	50	777	3	400	5	5	160	710.41	710.41	1800	0.00
Set3_E-n51-k5-s39-41.dat	2	50	777	3	400	5	5	160	728.54	729.28	1241	0.10
Set3_E-n51-k5-s40-41.dat	2	50	777	3	400	5	5	160	723.75	736.44	1200	1.75
Set3_E-n51-k5-s40-43.dat	2	50	777	3	400	5	5	160	752.15	769.98	1422	2.37

Table 6 – Set#4 results obtained by the math-heuristic.

	Qty			Qty		Max					Run	Dev
Set #4	Sat	Qty Clt.	Dmd	Tr	Truck Cap.	Van/Sat	Qty Vans	Van Cap	BKS	SA	time (s)	%
Set4a_Instance50-30.dat	3	50	20206	3	12500	3	6	5000	121163*	130872	52	8.01
Set4a_Instance50-31.dat	3	50	28153	3	12500	3	6	5000	149031*	153375	622	2.91
Set4a_Instance50-32.dat	3	50	20206	3	12500	3	6	5000	119905*	133844	18	11.63
Set4a_Instance50-33.dat	3	50	20153	3	12500	3	6	5000	150832*	155079	22	2.82
Set4a_Instance50-34.dat	3	50	20206	3	12500	3	6	5000	123396*	136450	57	10.58
Set4a_Instance50-35.dat	3	50	28153	3	12500	3	6	5000	171842*	179454	1235	4.43
Set4a_Instance50-36.dat	3	50	20206	3	12500	3	6	5000	122895*	131895	58	7.32
Set4a_Instance50-37.dat	5	50	28153	3	12500	3	6	5000	152873*	165148	55	8.03
Set4a_Instance50-38.dat	5	50	20206	3	12500	3	6	5000	116920*	134038	9	14.64
Set4a_Instance50-39.dat	5	50	28153	3	12500	3	6	5000	152091*	166168	6	9.26
Set4a_Instance50-10.dat	2	50	20206	3	12500	4	6	5000	140765*	142482	2632	1.2
Set4a_Instance50-11.dat	2	50	28153	3	12500	4	6	5000	204743*	208355	2989	1.76
Set4a_Instance50-12.dat	2	50	20206	3	12500	4	6	5000	120946*	123949	176	2.48
Set4a_Instance50-13.dat	2	50	28153	3	12500	4	6	5000	148179*	166433	2131	12.32
Set4a_Instance50-14.dat	2	50	20206	3	12500	4	6	5000	139364*	140412	1992	0.75
Set4a_Instance50-15.dat	2	50	28153	3	12500	4	6	5000	148992*	167089	2465	12.15
Set4a_Instance50-16.dat	2	50	20206	3	12500	4	6	5000	138919*	141094	835	1.57
Set4a_Instance50-17.dat	2	50	28153	3	12500	4	6	5000	208848*	211737	1339	1.38
Set4a_Instance50-18.dat	2	50	20206	3	12500	4	6	5000	122767*	122829	773	0.05
Set4a_Instance50-19.dat	3	50	28153	3	12500	3	6	5000	156465*	160116	101	2.33
Set4a_Instance50-20.dat	3	50	20206	3	12500	3	6	5000	127298*	147286	8	15.70
Set4a_Instance50-21.dat	3	50	28153	3	12500	3	6	5000	157782*	160274	107	1.58
Set4a_Instance50-22.dat	3	50	20206	3	12500	3	6	5000	128182*	143012	11	11.57
Set4a_Instance50-23.dat	3	50	28153	3	12500	3	6	5000	180734*	188014	204	4.03
Set4a_Instance50-24.dat	3	50	20206	3	12500	3	6	5000	128269*	133007	11	3.69
Set4a_Instance50-25.dat	3	50	28153	3	12500	3	6	5000	152240*	157474	33	3.44
Set4a_Instance50-26.dat	3	50	20206	3	12500	3	6	5000	116747*	134193	42	14.94
Set4a_Instance50-27.dat	3	50	28153	3	12500	3	6	5000	148156*	150282	1815	1.43
Set4a_Instance50-28.dat	3	50	20206	3	12500	3	6	5000	121045*	135017	8	11.54
Set4a_Instance50-29.dat	3	50	28153	3	12500	3	6	5000	172200*	185289	209	7.60
Set4a_Instance50-40.dat	5	50	20206	3	12500	2	6	5000	119942*	137199	87	14.39
Set4a_Instance50-41.dat	5	50	28153	3	12500	2	6	5000	166796*	192650	116	15.50
Set4a_Instance50-42.dat	5	50	20206	3	12500	2	6	5000	119454*	137684	99	15.26
Set4a_Instance50-43.dat	5	50	28153	3	12500	2	6	5000	143967*	162348	260	12.77
Set4a_Instance50-44.dat	5	50	20206	3	12500	2	6	5000	104514*	123848	16	18.50
Set4a_Instance50-45.dat	5	50	28153	3	12500	2	6	5000	145095*	162930	37	12.29
Set4a_Instance50-46.dat	5	50	20206	3	12500	2	6	5000	108879*	114407	40	5.08
Set4a_Instance50-47.dat	5	50	28153	3	12500	2	6	5000	158729*	185889	157	17.11

Table 6 – Set#4 results obtained by the math-heuristic. (cont.)

	Qty			Qty		Max					Run	Dev
Set #4	Sat	Qty Clt.	Dmd	Tr	Truck Cap.	Van/Sat	Qty Vans	Van Cap	BKS	SA	time (s)	%
Set4a_Instance50-48.dat	5	50	20206	3	12500	2	6	5000	108221*	130914	38	20.97
Set4a_Instance50-49.dat	5	50	28153	3	12500	2	6	5000	143488*	159766	48	11.34
Set4a_Instance50-50.dat	5	50	20206	3	12500	2	6	5000	108316*	113166	194	4.48
Set4a_Instance50-51.dat	5	50	28153	3	12500	2	6	5000	139803*	157056	15	12.34
Set4a_Instance50-52.dat	5	50	20206	3	12500	2	6	5000	112569*	119445	110	6.11
Set4a_Instance50-6.dat	2	50	28153	3	12500	2	6	5000	127999*	130620	56	2.05
Set4a_Instance50-7.dat	2	50	28153	3	12500	4	6	5000	145860*	166545	76	14.18
Set4a_Instance50-8.dat	2	50	20206	3	12500	4	6	5000	136376*	136927	2118	0.40
Set4a_Instance50-9.dat	2	50	28153	3	12500	4	6	5000	145025*	165233	304	13.93

Table 7 – Set#5 results obtained by the math-heuristic with alternative 2nd echelon model (subsection 3.3).

				Qty.	Truck	Max					Run	Dev
Set #5	Qty Sat	Qty Clt	Dmd	Tr	Cap	Van/Sat	Qty Vans	Van Cap	BKS	SA	time(s)	%
Set5_100-5-1.dat	5	100	5	100	1583	5	528	32	1564.46*	1735.34	9555	10.92
Set5_100-5-1b.dat	5	100	5	100	1583	5	528	15	1108.77	1276.45	9130	15.12
Set5_100-5-2.dat	5	100	5	100	1558	5	520	32	1016.32*	1155.92	9818	13.74
Set5_100-5-2b.dat	5	100	5	100	1558	5	520	15	782.25	944.64	9058	20.76
Set5_100-5-3.dat	5	100	5	100	1562	5	521	30	1045.29*	1208.49	10012	15.61
Set5_100-5-3b.dat	5	100	5	100	1562	5	521	16	828.54	1005.20	8815	21.32
Set5_100-10-1.dat	10	100	10	100	1610	5	537	35	1124.93	1197.17	19188	6.4
Set5_100-10-1b.dat	10	100	10	100	1610	5	537	18	916.25	1024.80	16254	11.85
Set5_100-10-2.dat	10	100	10	100	1536	5	512	33	1002.15	1156.35	18807	15.39
Set5_100-10-2b.dat	10	100	10	100	1536	5	512	18	774.11	926.17	16451	19.64
Set5_100-10-3b.dat	10	100	10	100	1540	5	514	32	854.88	942.15	17624	10.21
Set5_200-10-1.dat	10	200	10	100	1540	5	514	17	1556.95	1674.06	21560	7.52
Set5_200-10-1b.dat	10	200	10	200	3098	5	1033	62	1187.62	1325.38	20349	11.60
Set5_200-10-2.dat	10	200	10	200	3098	5	1033	30	1365.74	1601.90	19043	17.29
Set5_200-10-2b.dat	10	200	10	200	3101	5	1034	63	1002.85	1207.60	17841	20.42
Set5_200-10-3.dat	10	200	10	200	3101	5	1034	30	1793.99	2092.17	24854	16.62
Set5_200-10-3b.dat	10	200	10	200	3077	5	1026	63	1197.90	1453.68	18925	21.35

Table 8 – Set#6 results obtained by the math-heuristic with alternative 2nd echelon model (subsection 3.3).

				Qty.	Truck	Max	Qty	Van			Run	Dev
Set #6	Qty Sat	Qty Clt	Dmd	Tr.	Cap.	Van/Sat	Vans	Cap	BKS	SA	time(s)	%
Set6_A-n51-4.dat	4	50	777	2	640	50	50	160	652.00*	742.34	5877	13.86
Set6_B-n51-4.dat	4	50	777	2	640	50	50	160	563.98*	627.67	9361	11.29
Set6_C-n51-4.dat	4	50	777	2	640	50	50	160	689.18*	792.89	7399	15.05
Set6_A-n76-4.dat	4	75	1364	3	560	75	75	140	985.95*	1082.04	9358	9.75
Set6_B-n76-4.dat	4	75	1364	3	560	75	75	140	792.73*	916.76	9430	15.65
Set6_C-n76-4.dat	4	75	1364	3	560	75	75	140	1054.89*	1150.43	9448	9.06
Set6_A-n101-4.dat	4	100	1458	4	448	100	100	112	1194.17*	1255.64	9397	5.1
Set6_B-n101-4.dat	4	100	1458	4	448	100	100	112	939.83*	1027.80	9458	9.36
Set6_C-n101-4.dat	4	100	1458	4	448	100	100	112	1302.16	1435.89	9464	10.27
Set6_A-n51-5.dat	5	50	777	2	640	50	50	160	663.41*	718.15	5678	8.25
Set6_B-n51-5.dat	5	50	777	2	640	50	50	160	549.23*	623.50	6496	13.52
Set6_C-n51-5.dat	5	50	777	2	640	50	50	160	723.12*	811.46	9878	12.22
Set6_A-n76-5.dat	5	75	1364	3	560	75	75	140	979.48*	1037.86	10077	5.96
Set6_B-n76-5.dat	5	75	1364	3	560	75	75	140	784.44*	842.78	10006	7.44
Set6_C-n76-5.dat	5	75	1364	3	560	75	75	140	1115.32*	1209.79	10028	8.47
Set6_A-n101-5.dat	5	100	1458	4	448	100	100	112	1211.68*	1351.48	10145	11.54
Set6_B-n101-5.dat	5	100	1458	4	448	100	100	112	969.13*	1134.30	10090	17.04
Set6_C-n101-5.dat	5	100	1458	4	448	100	100	112	1305.82	1541.74	9539	18.07
Set6_A-n51-6.dat	6	50	777	2	640	50	50	160	662.51*	704.67	6179	6.36
Set6_B-n51-6.dat	6	50	777	2	640	50	50	160	556.32*	625.73	6425	12.48
Set6_C-n51-6.dat	6	50	777	2	640	50	50	160	697.00*	768.79	6148	10.30
Set6_A-n76-6.dat	6	75	1364	3	560	75	75	140	970.20*	1055.11	10691	8.75
Set6_B-n76-6.dat	6	75	1364	3	560	75	75	140	774.17*	868.04	10605	12.13
Set6_C-n76-6.dat	6	75	1364	3	560	75	75	140	1060.52	1212.65	10200	14.34
Set6_A-n101-6.dat	6	100	1458	4	448	100	100	112	1156.62	1260.22	11008	8.96
Set6_B-n101-6.dat	6	100	1458	4	448	100	100	112	960.76*	1081.42	10659	12.56
Set6_C-n101-6.dat	6	100	1458	4	448	100	100	112	1284.48	1515.38	10773	17.98

Table 9 – Average results obtained by the math-heuristic.

	Qty Clt.	Qty Sat	Dev %	Runtime (s)
Set#2	21(a)	2	1.34	684
	32(b)	2	2.56	659
	50(b)	2	1.18	2257
	50(c)	2	1.19	1990
	50(b)	4	4.38	2305
	50(c)	4	1.93	2303
Set#3	21	2	2.28	605
	32	2	3.21	928
	50	2	1.48	2634
Set#4	50	2	4,94	1376
	50	3	6,98	256
	50	5	12,38	80
Set#5	100	5	16.25	9398
	100	10	12.7	17665
	200	10	15.80	20428
Set#6	50	4	13.40	7546
	50	5	11.33	7350
	50	6	9.71	6250
	75	4	11.48	9412
	75	5	7.29	10037
	75	6	11.74	10498
	100	4	8.3	9440
	100	5	15.55	9924
	100	6	13.16	10813

5 CONCLUDING REMARKS

The main objective of the work was to develop a math-heuristic to obtain quality solution in computational time comparable to the methods analyzed in Sluijk et al. (2022). The math-heuristic allowed to high quality, optimum or near optimum solutions with faster processing time than the exact model for 30 and 50 customers instances when the number of maximum vehicles per satellite equals the second echelon number of available vehicles and the number of available vehicles in the second echelon is between two and five.

As the number of clients and satellites grow, and the number of second echelon vehicles restriction by satellite gets stricter, the search neighborhood grows and the math-heuristic with the chosen parameters to keep the processing time comparable to the approaches in Sluijk et al. (2022) did not reach good solutions, although all under 16.25% apart from the best-known solution. This is expected since the designed model relies heavily on the mathematical programming model resolution for the first echelon and second echelon (reason also that has driven the approach described in item 3.3 for Set#5 and Set#6 instances). Additionally, the two local implemented search operators (move and swap) between satellites and between routes for the alternative second echelon math-heuristic are intentionally restricted for movements that maintain the capacity restrictions, which prevents temporarily accepting unfeasible solutions to explore broader neighborhoods (Liu and Jiang, 2021) specially when the solutions are very tight (trucks are almost full and demand sizes do not allow the desired movements). The small number of local implemented operators (only two) also explains the poor results for medium and large instances (Arnold and Sorensen, 2019).

The results on benchmark instances show that such "cluster-first, route-second" math-heuristic approach utilizing package solvers (CPLEX and TSP CONCORDE) can effectively help solving the CVRP for small instances when compared to an exact method. The contribution of this paper lies in demonstrating the potential applications of package solvers to solve the CVRP on heuristic structures.

While the math-heuristic presented in this paper has certain limitations in solving medium and large instances, it serves to illustrate how the 'cluster-first, route-second' approach can be utilized to develop initial construction heuristic frameworks. This is due to the simplification of the original problem and the subsequent ability to gradually replace mathematical models with heuristic structures, leading to improved scalability. This work also motivates future research aimed at enhancing the simulated annealing approach by developing more efficient local search engines, effective destroy and repair methods together with local search operators to explore several neighborhoods (e.g., pruning or temporarily accepting unfeasible solutions) or alternatively devising population-based heuristics that are all state-of-the-art approaches while taking advantage of the current existing package solvers processing capacity.

References

ALESIANI F, ERMIS G & GKIOTSALITIS, K. 2022. Constrained Clustering for the Capacitated Vehicle Routing Problem (CC-CVRP). *Applied Artificial Intelligence* **36**(1): 1995658. DOI: 10.1080/08839514.2021.1995658

APPLEGATE DL, BIXBY RE, CHVATAL V & COOK WJ. 2003. Concorde-03.12.19, Available at: http://www.math.uwaterloo.ca/tsp/data/index.html (accessed in 20-December-2022).

ARNOLD F & SORENSEN K. 2019. Knowledge-guided local search for the vehicle routing problem. *Computers and Operations Research*, **105**:32–46. DOI: 0.1016/j.cor.2019.01.002

BOSCHETTI MA & MANIEZZO V. 2022. Matheuristics: using mathematics for heuristic design. *4OR*, **20**:173–208. DOI: 0.1007/s10288-022-00510-8

BRAEKERS K, RAMAEKERS K & VAN NIEUWENHUYSE I. 2016. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, **99**:300-313. DOI: 10.1016/j.cie.2015.12.007

BREUNIG U. 2-Echelon Capacitated Vehicle Routing Problem Instances, Available at: https: //homepage.univie.ac.at/ulrich.breunig/research/ (accessed in 20-Dec-2022).

BREUNIG U, SCHMID V, HARTL R & VIDAL T. 2016. A large neighbourhood-based heuristic for two-echelon routing problems, *Computers and Operations Research*, **76**: 208-225. DOI: 10.1016/j.cor.2016.06.014

CRAINIC TG, RICCIARDI N & STORCHI G. 2009. Models for Evaluating and Planning City Logistics Systems. *Transportation Science*, **743**(4): 432-454. DOI: 10.1287/trsc.1090.0279

CUDA R, GUASTAROBA G & SPERANZA MG. 2015. A survey on two-echelon routing problems. *Computers & Operations Research* **55**:185-199. DOI: 10.1016/j.cor.2014.06.008

CUNHA CB. 2006. *Contribuição à Modelagem de Problemas em Logística e Transportes*. Tese (Livre-Docência) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes.

DAGANZO CF. 1984. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation Science*, **18**(4): 331–350. DOI: 10.1287/trsc.18.4.331

FEO TA & RESENDE MGC. 1995. Greedy Randomized Adaptive Search Procedures, *Journal of Global Optimization*, **6**(2): 109–133. doi:10.1007/BF01096763.

GILLETT B & JG JOHNSON. 1976. Multi-terminal vehicle-dispatch algorithm. *Omega*, **4**: 711-718. DOI: 10.1016/0305-0483(76)90097-9.

GLOVER F. 1990. Tabu Search: A tutorial, *Interfaces*: 79–94. doi:10.1287/inte.20.4.74

HANSEN P & MLADENOVIĆ N. 2001. Variable neighborhood search: Principles and applications, *European Journal of Operational Research*, **130**(3): 449–467. doi:10.1016/S0377-2217(00)00100-4.

HUANG Y, SAVELSBERGH M & ZHAO, L. 2018. Designing logistics systems for home delivery in densely populated urban areas. *Transportation Research Part B*, **115**:95–125. DOI: 0.1016/j.trb.2018.07.006

JANJEVIC M, MERCHÁN D & WINKENBACH M. 2021. Designing multi-tier, multi-service-level, and multi-modal last-mile distribution networks for omni-channel operations. *European Journal of Operational Research* **294**(3): 1059-1077. DOI: 10.1016/j.ejor.2020.08.043

JEPSEN M, SPOORENDONK S & ROPKE S. 2015. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science* **47**(1): 23-37. DOI: 10.1287/trsc.1110.0399

KAUFMAN L & ROUSSEEUW P. J. 2005. *Finding groups in data: An introduction to cluster analysis*. Wiley series in probability and statistics. New York: Wiley-Interscience.

LAHYANI R, KHEMAKHEM M & SEMET F. 2015. Rich Vehicle Routing Problems: From a taxonomy to a definition. *European Journal of Operation Research¸* **241**(1), pp.1-14. DOI: 10.1016/j.ejor.2014.07.048

LI H, WANG H, CHEN J & BAI M. 2021. Two-echelon vehicle routing problem with satellite bi-synchronization. *European Journal of Operational Research* **288**(3): 775-793. DOI: 10.1016/j.ejor.2020.06.019

LI J, XU M & SUN P. 2022. Two-echelon capacitated vehicle routing problem with grouping constraints and simultaneous pickup and delivery. *Transportation Research Part B: Methodological* **162**:261-291. DOI: 10.1016/j.trb.2022.06.003

LIU D, DENG Z, MAO X, YANG Y & KAISAR EI. 2020. Two-Echelon Vehicle-Routing Problem: Optimization of Autonomous Delivery Vehicle-Assisted E-Grocery Distribution, *IEEE Access*, **8**: 108705-108719. DOI: 10.1109/ACCESS.2020.3001753

LIU R & JIANG S. 2021. A variable neighborhood search algorithm with constraint relaxation for the two-echelon vehicle routing problem with simultaneous delivery and pickup demands. *Soft Computing*, **26**: 8879–8896. DOI: 10.1007/s00500-021-06692-3

MARA S & KUO RJ. 2018. Location-routing problem: a classification of recent research, *International Transactions in Operational Research* **28**: 2941-2983. DOI: 10.1111/itor.12950

MARUJO LG, GOES GV, D'AGOSTO MA, FERREIRA, A. F., WINKENBACH M & BANDEIRA RA. 2018. Assessing the sustainability of mobile depots: The case of urban freight distribution in Rio de Janeiro. *Transportation Research Part D: Transport and Environment*, **62**: 256–267. DOI: 10.1016/j.trd.2018.02.022

PERBOLI G, TADEI R & VIGO D. 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. *Transportation Science*, **45**(3): 364-380. DOI: 10.1287/trsc.lll0.0368

PRINS C, LACOMME P & PRODHON C. 2014. Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C*, **40**: 179-200. DOI: 10.1016/j.trc.2014.01.011

SINGH B, OBERFICHTNER L & IVLIEV S. 2023. Heuristics for a cash-collection routing problem with a cluster-first route-second approach. *Annals of Operations Research*, **322**: 413–440. DOI: 0.1007/s10479-022-04883-1

SLUIJK N, AM FLORIO, J KINABLE, N DELLAERT & T.V. WOENSEL. 2022. Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, **23** (in-proof). DOI: 10.1016/j.ejor.2022.02.022

VIDAL T, CRAINIC TG, GENDREAU M & PRINS C. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operation Research*, **234**: 658–673. DOI: 10.1016/j.ejor.2013.09.045

VIDAL T, LAPORTE G & MATL P. 2020. A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research* **286**(2): 401-416. DOI: 10.1016/j.ejor.2019.10.010

VOIGT S, FRANK M, FONTAINE P & KUHN H. 2022. Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research* **146**: 105856. DOI: 10.1016/j.cor.2022.105856

XINCHAO Z. 2011. Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing* **11**: 1827–1836. DOI: 10.1016/j.asoc.2010.05.029

WANG K, SHAO Y & ZHOU W. 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D* **57**: 262–276. DOI: 10.1016/j.trd.2017.09.018

WINKENBACH M, KLEINDORFER PR & SPINLER S. 2015. Enabling Urban Logistics Services at La Poste through Multi-Echelon Location-Routing. *Transportation Science*, **50**(2): 520–540. DOI: 10.1287/trsc.2015.0624

ZHOU L, BALDACCI R & VIGO D. 2018. A Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options Arising in the Last Mile Distribution. *European Journal of Operational Research* **265**: 765–778. DOI: 0.1016/j.ejor.2017.08.011

How to cite

GUIMARÃES JCF & CUNHA CB. 2023. Math-heuristic for the capacitated two-echelon vehicle routing problem. *Pesquisa Operacional*, **43**: e270829. doi: 10.1590/0101-7438.2023.043.00270829.