Comput Optim Appl (2015) 61:159–187
DOI 10.1007/s10589-014-9713-5

A multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem

Juan Carlos Rivera · H. Murat Afsar · Christian Prins

Received: 22 October 2013 / Published online: 19 November 2014

Abstract The multitrip cumulative capacitated vehicle routing problem (mt-CCVRP) is a non-trivial extension of the classical CVRP: the goal is to minimize the sum of arrival times at demand nodes and each vehicle may perform several trips. Applications of this NP-hard problem can be found in disaster logistics and maintenance operations. Contrary to the CVRP, the cost of a solution varies if a trip is reversed or if its rank in a multitrip is changed. Moreover, evaluating local search moves in constant time is not obvious. This article presents a mixed integer linear program (MILP), a dominance rule, and a hybrid metaheuristic: a multi-start iterated local search (MS-ILS) calling a variable neighborhood descent with $O(1)$ move evaluations. On three sets of instances, MS-ILS obtains good solutions, not only on the mt-CCVRP, but also on the cumulative CVRP where it competes with four existing algorithms. Moreover, the metaheuristic retrieves the optimal solutions of the MILP, which can be computed for small instances using a commercial solver.

Keywords Multitrip cumulative capacitated vehicle routing problem · Disaster logistics · Iterated local search · Variable neighborhood descent

1 Introduction

In the last decade, several disasters around the world have caused millions of victims and massive destructions of infrastructure and environment. From 2003 to 2012,

---

J. C. Rivera · H. M. Afsar · C. Prins (✉)
ICD-LOSI, UMR CNRS 6281, Troyes University of Technology (UTT), Troyes, France
e-mail: christian.prins@utt.fr

J. C. Rivera
e-mail: jkrivera@gmail.com

H. M. Afsar
e-mail: murat.afsar@utt.fr

according to the International Federation of Red Cross and Red Crescent Societies (IFRC) [40], 670 disasters occurred on average every year, which have caused 115 thousand deaths and 216 million of affected people. Seismic events have induced the greatest number of deaths, on average 67,818 per year, while flood events have affected the largest number of people, on average 108 million per year. The enormous scale of these disasters, which can be checked in databases like EM-DAT, AirDisaster.com, Natural Disaster Reference Database, The British Association for Immediate Care, Social Studies Network for Disaster Prevention in Latin America, Disaster Resource Guide or DisasterRelief.org, has brought attention to the need for methodology and technology for effectively managing relief supply chains. Although a great deal of research and technology is available for industrial supply chains, Beamon [10] notes that the challenges associated with managing a humanitarian relief chain following a large-scale emergency are often quite different.

Altay and Green III [2] define disaster operations as the set of activities performed before, during, and after a disaster, with the goal of preventing loss of human life, reducing its impact on the economy, and returning to a state of normalcy. IFRC [1] defines a disaster as a sudden, calamitous event that seriously disrupts the functioning of a community or society and causes human, material, and economic or environmental losses that exceed the ability of a community or society to cope with using its own resources. The combination of hazards, vulnerability and inability to reduce the potential negative consequences of a risk is a classical cause of disasters.

IFRC classifies disaster types as natural hazards, which can be geophysical, hydrological, climatological or biological, and technological or man-made hazards. It provides definitions for each type of disaster while Green III and McGinnis [35] discuss a broad classification of these events by causation.

Altay and Green III [2] define also the emergency response as the global response to catastrophic events, without considering the daily responses of ambulance, police, or fire departments to routine emergency calls. Indeed, there exists a general agreement about a clear demarcation between what are termed routine emergencies [39], also called everyday emergencies [29], and more serious emergencies induced by disasters and catastrophes. Readers interested in research on daily emergencies are referred to Swersey [71] and Chaiken and Larson [22].

Emergency response operations consist in four phases: mitigation, preparedness, response and recovery [2]. Mitigation is the application of measures that will either prevent the onset of a disaster or reduce its impacts. Preparedness activities train the community to react when a disaster occurs. Response is the employment of resources and emergency procedures to preserve life, property, the environment, and the social, economic, and political structure of the community. Recovery involves the actions in order to stabilize the community and to restore some semblance of normalcy after the immediate impact of the disaster.

An important logistical issue after a disaster is to determine the transportation routes for first aids, supplies, rescue personnel and equipment between supply points and a large number of destination nodes, geographically scattered over the disaster region. The arrival time of relief supplies at the affected communities clearly impacts the survival rate of the citizens and the amount of suffering. According to Yi and

Özdamar [76], logistic support and evacuation are two major activities in disaster response.

The aim of this paper is to study a new version of the capacitated vehicle routing problem, raised by the deployment of helicopters in the response phase of relief operations: (a) the classical objective (total time or distance) becomes the sum of arrival times at affected sites and (b) vehicles are allowed to perform multiple trips. We call this problem the *multitrip cumulative capacitated vehicle routing problem* (mt-CCVRP). After disasters such as earthquakes, floods and hurricanes, roads and airfields can be damaged or blocked and only helicopters can be used in a first step to deliver relief aid. This research takes place in a wider program on disaster logistics funded by our university. Its first phase is to design algorithms for various vehicle routing problems identified in this domain, including the mt-CCVRP. The resulting optimization black boxes will be integrated in a software platform in a second phase.

The paper is structured as follows: Sect. 2 briefly reviews some problems related to our mt-CCVRP. In Sect. 3 the problem is formally defined and modelled as a mixed linear program. A heuristic approach derived from iterated local search is developed in Sect. 4. Computational results are presented in Sect. 5 while concluding remarks are given in Sect. 6.

2 Context and problems related to the mt-CCVRP

A recent trend is to apply operations research to facilitate logistic operations in disaster relief. Examples include inventory systems [11,12]), facility location [6], recovery of infrastructures [50] and vehicle routing [20,37,58]). Various studies focus on the four phases of disaster relief: mitigation [25,28], preparedness [51,74,75], response [8,9,58,69], and recovery [19,23,44,45,55]. Additional information about disaster relief can be found in the surveys of Altay and Green III [2], Caunhye et al. [21], and Galindo and Batta [32].

Özdamar et al. [58] describe a model that coordinates deliveries of supplies from different depots in the context of relief operations, while Barbarosoğlu et al. [9] propose a hierarchical decision support methodology to use helicopters efficiently. Pettit and Beresford [60] model the relationships between participating bodies including military and non-military organizations. De Angelis et al. [26] consider the airplane routing and scheduling problem to transport food after an emergency in Angola. Bakuli and Smith [5] propose a queuing network model for emergency evacuations.

As Campbell et al. [20] explain, it is critical that the deliveries to affected sites in a relief context be both fast and fair. These authors suggest that using service-based objective functions may better reflect the different priorities and strategic goals found in delivering humanitarian aid. Our approach considers such a service-based criterion, by minimizing the sum of arrival times.

The sum of arrival times has been already used in some traveling salesman problems (TSP). The minimum latency problem consists in finding a tour starting at a depot and visiting each other node only once, in such a way that the total latency is minimized [4,15]. The latency of a node is defined as the total distance or travel time to reach that node. This problem is also known as the delivery man problem [30] or the travelling repairman problem (TRP) [41,73] because of its possible application to maintenance operations. The multiple travelling repairman problem (k-TRP) is a generalization of the minimum latency problem where *k* tours must be determined [41]. The time-dependent travelling salesman problem (TDTSP) is a generalization of the standard TSP in which the traversal cost of an arc depends on its position in the tour [34,48,61]. When the objective of the TDTSP is to minimize the sum of distances travelled from the depot to each node, the problem is known as the TSP with cumulative cost or cumulative TSP (CTSP) [14].

Among applications requiring several capacitated vehicles, school bus routing problems (SBRP) frequently consider similar objective functions. Li and Fu [46] present a multi-objective algorithm for the SBRP with four objectives, one of them being the minimization of total student travel time. Bennett and Gazis [13] present a bi-objective approach minimizing both the total bus travel time and the total student travel time.

Kara et al. [42] tackle another vehicle routing problem that they call cumulative VRP. They explain that the cost of a route is not adequately represented by the distance between nodes. Some factors such as fuel consumption and vehicle load can be captured by a function of the flows on the arcs. The contribution to the cost proposed for each arc is the arc length multiplied by the flow traversing the arc (the vehicle load). The objective function is the sum of these products over the set of traversed arcs and can be considered as a generalization of the *k*-TRP. The problem is modelled as an integer linear program and tested on the highway network of Turkey to minimize the energy consumption of a fleet of vehicles.

In the context of disaster logistics, the last mile distribution problem, defined by Balcik et al. [7] as the final stage of the relief chain, is also related to the cumulative VRP. It refers to delivery of relief supplies from local and temporary distribution centers to the people in the affected areas (demand locations) and combines location and routing decisions. For instance, Özdamar [57] presents a planning system to coordinate helicopters in the last mile distribution problem after a disaster.

The cumulative capacitated vehicle routing problem (CCVRP), described by Ngueveu et al. [54], is a variant of the classical CVRP where the objective function becomes the sum of arrival times at demand nodes. The CCVRP can also be considered as the generalization of the minimum latency problem to several vehicles. These authors provide a mathematical model, several lower bounds and two memetic algorithms. Solutions are compared using classical CVRP instances [24], replacing the total length of the routes by the sum of arrival times, and on TRP instances [67]. The results confirm that the CCVRP and the classical CVRP can have quite different solutions on the same instance, as already observed by Campbell et al. [20]. In Euclidean versions for instance, CVRP solutions with edge-crossings are suboptimal, while the elimination of these crossings often brings no improvement for the cumulative objective [20].

Ribeiro and Laporte [65] present an adaptive large neighborhood search (ALNS) algorithm for the CCVRP, which is compared with the memetic algorithms in [54] using additional instances from Golden et al. [33]. Ke and Feng [43] propose a two-phase metaheuristic which applies exchange-based and cross-based operators to perturb the solutions in the first phase and a local search procedure in the second phase. This metaheuristic improves some best known solutions from previous works [54,65].

The open VRP (OVRP) is a another CVRP variant, where each route ends at its last customer, without returning to the depot [64,66]. Hence, like in the CCVRP, the distance to go back to the depot is ignored in the objective function. Nevertheless, Ngueveu et al. [54] give an example where the optimal OVRP and CCVRP solutions are very dissimilar.

A comparison between cost minimization, maximal arrival time minimization and average arrival time minimization for the TSP and CVRP is given by Campbell et al. [20]. Their paper presents lower bounds, an insertion heuristic and a local search procedure. Maximal arrival time minimization is also addressed by Applegate et al. [3] via a branch-and-cut algorithm, while Hemel et al. [38] solve a practical problem aiming at minimizing the maximal tour length, using the average arrival time to break ties. Dell et al. add a multi-period horizon and equity constraints [27].

In most studies, a fundamental assumption is that each vehicle performs a single trip. Clearly, in many cases this assumption does not hold. In the last decade, some publications have dealt with the case where vehicles are allowed to make multiple trips. The multitrip extension of the classical CVRP is known as the multitrip vehicle routing problem (mt-VRP), introduced by Fleischmann [31].

Taillard et al. [72] design a hybrid algorithm for the mt-VRP by combining tabu search, a population-based approach and a bin-packing heuristic. A penalty is incurred when a trip exceeds a given time limit. Brandão and Mercer [17] elaborate an approach for a real life problem with multiple trips, time windows, unloading times and access restrictions at customers, and a maximum length for each trip. The same authors compare in [18] a tabu search metaheuristic with the algorithm presented in Taillard et al. [72].

Petch and Salhi [59] develop an algorithm combining the approaches proposed by Brandão and Mercer [18] and by Taillard et al. [72] and evaluate the savings induced by multiples trips. Olivera and Viera [56] describe an adaptive memory procedure while Salhi and Petch [68] choose a genetic algorithm. Gribkovskaia et al. [36] present a real application of the mt-VRP. Prins [62] investigate the mt-VRP with heterogeneous vehicles and proposed constructive heuristics and a tabu search.

To the best of our knowledge, no published article has considered the multi-trip cumulative capacitated vehicle routing problem. The special objective already complicates the moves in local search procedures for the cumulative CVRP, although each vehicle is limited to one trip in this problem [54]. For instance, the cost of a sub-sequence of customers changes when it is inverted by a 2-opt move, contrary to the classical CVRP. As we shall see, the calculations are even more involved if multiple trips are allowed, since the cost of a multi-trip depends on the order of its trips.

3 Problem definition and mixed integer linear model

Compared to the CVRP, the mt-CCVRP uses the sum of arrival times at required nodes as objective function, like in the cumulative CVRP, but it allows more than one trip per vehicle. This flexibility is necessary when the total demand exceeds the total capacity of the fleet of vehicles. Moreover, while the fleet size is often a decision variable in

CVRP, it must be fixed for the CCVRP and the mt-CCVRP, otherwise the least-cost solution consists in doing a direct trip for each required node, as noted by Ngueveu et al. [54].

In the context of humanitarian logistics, the mt-CCVRP models the distribution of relief supplies to a set of sites affected by a disaster, using for instance helicopters which can do multiple sorties. We consider a planning horizon limited to one day, without time windows on sites nor maximum working times for vehicles. Indeed, time windows are imposed by customers in commercial logistics, while maximum working times result from regulations. After a disaster the victims are waiting for rescue and successive crews can be used to use each helicopter full time. The current version of our problem, which concerns mainly the preliminary transportation of experts, medical staff, radio equipment and first-aid drugs to the affected sites to make a first evaluation. Hence, there is a relatively small amount of goods and only one or two passengers per destination, which does not justify split deliveries.

The problem can be defined on an undirected complete graph $G = (V, E)$. The node-set $V = \{0, \dots, n\}$ includes a depot-node 0 and a subset $V' = V \setminus \{0\}$ of affected sites, also called *demand nodes* or *required nodes*. In the sequel, it is assumed that G is encoded as a symmetric directed graph, i.e., each edge $e = [i, j]$ is replaced by two opposite arcs (i, j) and (j, i), with a travel time $w_{ij} = w_{ji}$. A fleet of R identical vehicles of capacity Q is based at the depot and each node $i \in V'$ has a known demand q_i. It is assumed without loss of generality that $\sum_{i \in V'} q_i \geq R \times Q$, $n \geq R$, and $q_i \leq Q$, $\forall i \in V'$.

The objective is to identify a set of trips such that each site is visited exactly once and the sum of arrival times at sites is minimized. A trip is defined as a circuit, starting and ending at the depot, whose total demand fits vehicle capacity Q. Every trip must be assigned to exactly one vehicle and, if necessary, vehicles can perform more than one trip. Moreover, the trips assigned to each vehicle must be ordered because this affects the objective function. The set of successive trips performed by one vehicle is called a *multitrip* in the sequel.

We describe now a non-trivial 0–1 mixed integer linear program (MILP) for the mt-CCVRP. Contrary to the model proposed by Ngueveu et al. [54], variables are indexed only by arcs and no trip nor multitrip index is required. Before presenting the model equations we introduce the concepts of *replenishment arc* and *arc coefficient* that are central in our formulation.

The notion of *replenishment arc* has been used by Boland et al. [16] and Mak and Boland [49] for a multitrip traveling salesman problem in which one vehicle with limited capacity must replenish (reload) at the depot. Replenishment arcs constitute a nice trick to replace the successive trips of a multitrip by a single trip: when a trip with last customer i is followed by a trip with first customer j, the two arcs $(i, 0)$ and $(0, j)$ are replaced by a replenishment arc (i, j). We use the same technique for our mt-CCVRP: a multitrip is a sequence of arcs in which replenishment arcs delimit successive trips. The length of a replenishment arc (i, j) is $w'_{ij} = w_{i0} + w_{0j}$.

The concept of *arc coefficient* is defined by Ngueveu et al. for the CCVRP [54]. The sum of arrival times Z for a trip defined without loss of generality by a sequence of u required nodes $(1, 2, \dots, u)$ can be written as:

$$Z = \sum_{i=1}^{u} t_i = t_1 + t_2 + \cdots + t_u \tag{1}$$

where $t_i = \sum_{j=0}^{i-1} w_{j,j+1}$ is the arrival time to site i. Z can be rewritten as:

$$Z = u \cdot w_{01} + (u-1) \cdot w_{12} + \cdots + w_{u-1,u} = \sum_{j=0}^{u-1} (u-j) \cdot w_{j,j+1} \tag{2}$$

In other words, the traversal time of the first arc in the route is counted u times, the cost of the second arc $u-1$ times, and so on. The number of times the cost of arc (i, j) is counted is what we call the *coefficient* of (i, j). Note that this coefficient is also the number of demand nodes visited after arc (i, j) in the route, including node j. The concept can be extended to the replenishment arcs in our multitrip CCVRP.

Our model is based on five types of variables. Like in all flow-based vehicle routing models, variables F_{ij} define the flow on each arc (i, j), i.e., the load of the vehicle traversing this arc. The binary variables x_{ij} are equal to 1 if and only if arc (i, j) is traversed by a vehicle. The arc coefficients explained before are expressed by variables y_{ij} which are very useful to compute the objective function and prevent subtours. Due to the special nature of replenishment arcs, similar but separate variables x'_{ij} and y'_{ij} are used for these arcs.

To better understand the model, note that four basic moves are possible for a vehicle: (1) it leaves the depot to reach the first required node j of its multitrip ($x_{0j} = 1$); (2) it traverses a normal arc linking two required nodes i and j ($x_{ij} = 1$); (3) it traverses a replenishment arc connecting two required nodes i and j, i.e., it ends a trip at i, reloads at the depot and initiates a new trip with j ($x'_{ij} = 1$); (4) it returns to the depot after the last required node of its multitrip ($x_{j0} = 1$). Normal arcs $(0, j)$ leaving the depot are used only at the beginning of the first trip of a multitrip. The fourth move is not important because the return arcs $(j, 0)$ have no influence on the objective function considered. We assume that variables x_{j0} exist, but with a null cost.

The resulting 0–1 mixed integer linear program, a flow-based formulation, is given by equations (3) to (21). To make the equations lighter, we assume that $i \neq j$ for all variables. The objective function (3) represents the sum of arrival times to affected sites, rewritten like in equation (2) using the arc coefficient variables y_{ij}. The second sum with the y'_{ij} concern replenishment arcs (i, j), as their traversal time is in fact $w_{i0} + w_{0j}$. Constraints (4) mean that only R vehicles (R multitrips) can be used (recall that variables x_{0i} are used only at the beginning of the first trip of a vehicle).

Equations (5) and (6) respectively indicate that exactly one arc is traversed to arrive at site j and leave it. These equations are based on the three first moves explained before.

Constraints (7) to (9) concern flow variables. Equations (7) express the vehicle load variation after a visit to site i and ensure that each demand is satisfied. Using Equations (8), no flow can traverse an unused arc and each flow is limited by vehicle capacity. Equations (9) limit the maximum flow at the beginning of each trip to the capacity

Q of vehicles: $x_{0j} = 1$ concerns the flow for the first trip of a vehicle while $x'_{ij} = 1$ concerns its other trips, after a replenishment arc.

$$\min \ Z = \sum_{i \in V} \sum_{j \in V} w_{ij} \ y_{ij} + \sum_{i \in V'} \sum_{j \in V'} (w_{i0} + w_{0j}) \ y'_{ij} \tag{3}$$

$$\sum_{i \in V'} x_{0i} = R \tag{4}$$

$$\sum_{i \in V'} (x_{ij} + x'_{ij}) + x_{0j} = 1 \quad \forall \, j \in V' \tag{5}$$

$$\sum_{i \in V'} (x_{ji} + x'_{ji}) + x_{j0} = 1 \quad \forall \, j \in V' \tag{6}$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V} F_{ij} = q_i \quad \forall \, i \in V' \tag{7}$$

$$F_{ij} \leq Q \, x_{ij} \quad \forall \, i \in V', \ j \in V \tag{8}$$

$$F_{0j} \leq Q \left(x_{0j} + \sum_{i \in V'} x'_{ij} \right) \quad \forall \, j \in V' \tag{9}$$

$$y_{ij} \leq (n - R + 1) \, x_{ij} \quad \forall \, i, \ j \in V \tag{10}$$

$$y'_{ij} \leq (n - R) \, x'_{ij} \quad \forall \, i, \ j \in V' \tag{11}$$

$$\sum_{j \in V} (y_{ji} - y_{ij}) + \sum_{j \in V'} (y'_{ji} - y'_{ij}) = 1 \quad \forall \, i \in V' \tag{12}$$

$$y_{ij} \geq x_{ij} \quad \forall \, i \in V, \ j \in V' \tag{13}$$

$$y'_{ij} \geq x'_{ij} \quad \forall \, i, \ j \in V' \tag{14}$$

$$y_{ij} \geq 2 \cdot x_{ij} - x_{j0} \quad \forall \, i \in V, \ j \in V' \tag{15}$$

$$y'_{ij} \geq 2 \cdot x'_{ij} - x_{j0} \quad \forall \, i, \ j \in V' \tag{16}$$

$$x_{ij} \in \{0, 1\} \quad \forall \, i, \ j \in V, \ i \neq j \tag{17}$$

$$x'_{ij} \in \{0, 1\} \quad \forall \, i, \ j \in V', \ i \neq j \tag{18}$$

$$y_{ij} \geq 0 \quad \forall \, i, \ j \in V, \ i \neq j \tag{19}$$

$$y'_{ij} \geq 0 \quad \forall \, i, \ j \in V', \ i \neq j \tag{20}$$

$$F_{ij} \geq 0 \quad \forall \, i, \ j \in V, \ i \neq j \tag{21}$$

Constraints (10) to (12) concern arc coefficients. Equations (10) (resp. (11)) limit the coefficients y_{ij} (resp. y'_{ij}) to the maximum number of normal or replenishment arcs that can be used in a multitrip. Ngueveu et al. [54] showed that all the vehicles are used in an optimal CCVRP solution (each vehicle visits at least one demand node) and this is still true for the mt-CCVRP. This implies that no multitrip can visit more than $(n - R + 1)$ demand nodes, which is also the maximum arc coefficient. In constraints (11) the limit is reduced by one, because there is at least one required node before a replenishment arc.

Constraints (12) imply that the coefficients of successive arcs decrease along a trip. This property prevents subtours, like in the Miller-Tucker-Zemlin subtour elimination constraints for the TSP [52]. Note that it is not necessary to use integer variables for y_{ij} and y'_{ij}. Constraints (13) to (16) are valid inequalities, since they are not necessary to guarantee feasibility but allow to find better linear relaxations and reduce on average the time to get optimal solutions. Equations (13) (resp. (14)) assign an arc coefficient greater than or equal to one to a normal arc (resp. replenishment arc) when this arc is traversed. Similarly, equations (15) (resp. (16)) assign an arc coefficient greater than or equal to two to a normal arc (resp. replenishment arc), when this arc is traversed and the next arc in the route is not the last one ($x_{j0} = 0$).

Finally, constraints (17) to (21) define the five groups of variables. According to constraints (10), (11) and (12), note that the last arc traversed by each vehicle may have a rank equal to zero, which means that this arc is not considered in the objective function. The model has been tested on small instances and computational results are reported in Sect. 5.

4 Metaheuristic approach

4.1 Principle and general structure

The proposed metaheuristic is a *multi-start iterated local search* (MS-ILS) calling a *variable neighborhood descent* (VND). Iterated local search has proved its efficiency on various optimization problems, see the tutorial of Lourenço et al. [47]. The MS-ILS variant, consisting in restarting an ILS from several initial solutions to diversify the search, has given excellent solutions on the CVRP [63]. VND is a kind of local search in which wider and wider neighborhoods are successively explored [53]. The resulting hybrid metaheuristic is sketched in Algorithm 1 while its internal components are described in the sequel.

MaxStart successive ILS are executed and update a global best solution bs. Each ILS calls a greedy randomized heuristic to get one initial solution, improves it using the VND, and performs *MaxIter* iterations. Each iteration of the incumbent ILS takes a copy s' of s, applies a perturbation procedure (*Perturb*) and calls the VND to improve the perturbed solution. In most implementations, s is replaced by s' in case of improvement. As solutions with equal costs are not rare, we decided to better diversify the search by replacing also s by s' when $Z(s) = Z(s')$ (line 11).

Note that each ILS generates a sequence of local optima with decreasing costs. The procedures *Precomputations* and *Update_Precomputations*, explained later, are used to speed up the VND.

4.2 Solution representation

A solution is encoded as R lists of nodes, one per vehicle or multitrip. Each list indicates the order in which the required nodes are visited by the corresponding multitrip, using the depot (node 0) as delimiter. Figure 1 gives an example for seven required nodes with respective demands {10, 20, 10, 20, 10, 10, 10} and two vehicles with capacity 35. The arcs used are plotted with their traversal times w_{ij}. Vehicle 1 performs one multitrip of two trips while vehicle 2 performs one trip only. Arc (2, 3) is an example of replenishment arc. The arrival times are written near each node. The corresponding arc

Algorithm 1 – MS-ILS

```
Z(bs) ← ∞
for Start ← 1 to MaxStart do
    Greedy_Randomized_Heuristic (s)
    Precomputations (s)
    VND (s)
    for Iter ← 1 to MaxIter do
        s' ← s
        Perturb (s')
        Update_Precomputations (s')
        VND (s')
        if Z(s') ≤ Z(s) then
            s ← s'
        end if
    end for
    if Z(s) < Z(bs) then
        bs ← s
    end if
end for
```

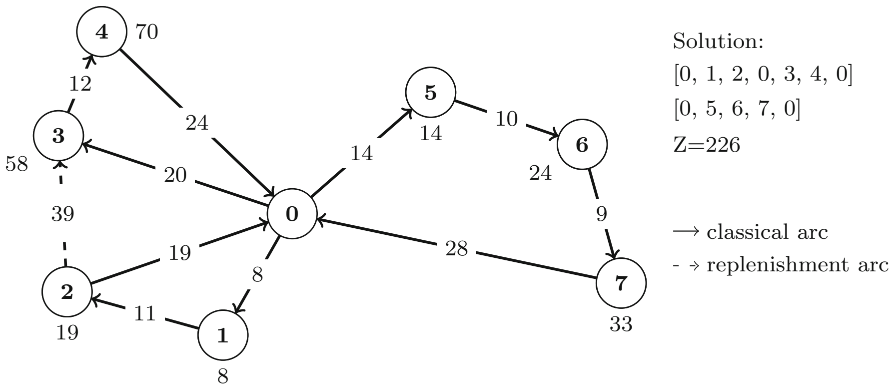

Fig. 1 Example of solution with $n = 7$, $R = 2$ and $Q = 35$

coefficients are: $y_{01} = 4$, $y_{12} = 3$, $y'_{23} = 2$, $y_{34} = 1$ and $y_{40} = 0$ for the first multitrip, and $y_{05} = 3$, $y_{56} = 2$, $y_{67} = 1$ and $y_{70} = 0$ for the second multitrip. The cost of this solution (sum of arrival times to required nodes) is $Z = \sum_{i \in V} \sum_{j \in V} w_{ij} \cdot y_{ij} = 226$.

4.3 Precomputations

Contrary to the CVRP, it is not obvious to evaluate a move in constant time in local search procedures for the mt-CCVRP, due to several subtle differences. Inserting or removing a node in a route shifts the arrival times at the following nodes. Several moves in our VND involve reversals of sequences of nodes: unlike the CVRP, the cost of a subsequence is modified after a reversal and, in particular, the cost of a trip changes when performed backwards. A move in a route changes also the costs of the following routes in the same multitrip. Finally, the cost of a multitrip depends on the ordering of its trips.

Table 1 Notations from Silva et al. [70] for a sequence S of nodes

Symbol	Definition
$C(S)$	Cost to perform S when leaving i at time 0, $C(S) = \sum_{k=i}^{j} t_k$
$D(S)$	Duration of S, i.e., $D(S) = \sum_{k=i}^{j-1} w_{k,k+1}$
$\|S\|$	Number of nodes in S

Ngueveu et al. [54] succeed in evaluating each move in $O(1)$ in their heuristics for the CCVRP, but using heavy cost variation formulas. Silva et al. [70] proposed for the cumulative TSP simpler equations, based on the fact that any move can be expressed by concatenations of node sequences. For instance, if node 5 is inserted in route $(1, 2, 3, 4)$ after node 2, the result can be written as $(1, 2) \oplus (5) \oplus (3, 4)$, $\oplus$ denoting the concatenation. Silva et al. define the attributes listed in Table 1 for a sequence of nodes $S = (S_i, S_{i+1}, \ldots, S_j)$, denoted in the sequel as $(i, i+1, \ldots, j)$ to save one level of indexing. Recall that t_k denotes the arrival time at node k.

For a sequence S reduced to a single node, we have of course $|S| = 1$ and $C(S) = L(S) = 0$. If sequence $S' = (i', i'+1, \ldots, j')$ is concatenated after S, the following equations can be used to deduce the attributes for $S \oplus S'$:

$$|S \oplus S'| = |S| + |S'| \tag{22}$$

$$D(S \oplus S') = D(S) + w_{j,i'} + D(S') \tag{23}$$

$$C(S \oplus S') = C(S) + |S'| \cdot (D(S) + w_{j,i'}) + C(S') \tag{24}$$

This formalism for the cumulative TSP can be extended to our mt-CCVRP, where a sequence S may span several trips in a multitrip and include copies of the depot. Three modifications are required. The first one is to check vehicle capacity by introducing the total demand $W(S)$ for a sequence S: a trip obtained by concatenating two sequences S and S' is feasible if $W(S) + W(S') \leq Q$.

The second modification is to ignore depot copies when computing $C(S)$ and $|S|$, i.e., k in the sum for $C(S)$ must be such that $S_k \neq 0$ and $|S|$ counts only the required nodes in S. However, $D(S)$ does not change and still includes the cost of arcs incident to depot copies. As several moves in our VND are based on sequence reversals, the third change consists in defining for a sequence S its reversal $\overleftarrow{S}$. Note that $|\overleftarrow{S}| = |S|$ and $D(\overleftarrow{S}) = D(S)$. The way of computing $C(\overleftarrow{S})$ is indicated at the end of this subsection.

As an example, imagine that in Fig. 1 a new demand node 8 is inserted after node 1 in the first multitrip, with $q_8 = 1$, $w_{18} = 7$ and $w_{82} = 8$. Vehicle capacity is respected but how to compute the cost of the resulting multitrip? If $S = (0, 1)$, $S' = (8)$ and $S'' = (2, 0, 3, 4, 0)$, the goal is to get $C(S \oplus S' \oplus S'')$. Using (24), we compute first $C(S \oplus S') = C(S) + |S'| \cdot (D(S) + w_{18}) + C(S') = w_{01} + 1 \cdot (w_{01} + w_{18}) + 0 = 2w_{01} + w_{18}$. Then, if $S \oplus S' = \overline{S}$, we calculate $C(S'') = t_2 + t_3 + t_4 = 0 + (w_{20} + w_{03}) + (w_{20} + w_{03} + w_{34}) = 2w_{20} + 2w_{03} + w_{34}$, and finally $C(\overline{S} \oplus S'') = C(\overline{S}) + |S''| \cdot (D(\overline{S}) + w_{82}) + C(S'') = (2w_{01} + w_{18}) + 3((w_{01} + w_{18}) + w_{82}) + (2w_{20} + 2w_{03} + w_{34})$. We obtain $5w_{01} + 4w_{18} + 3w_{82} + 2w_{20} + 2w_{03} + w_{34}$

$= 5 \times 8 + 4 \times 7 + 3 \times 8 + 2 \times 19 + 2 \times 20 + 12 = 182$. Note that $2(w_{20} + w_{03})$ corresponds to the contribution of replenishment arc (2, 3) in the objective function (4) of the mathematical model.

In the previous example, the computations can be done in constant time provided the attributes of involved sequences are already computed. This is the role of the *Precomputations* procedure in Algorithm 1, which calculates the attributes for each sequence $S = (i, i+1, \ldots, j)$ contained in the multitrips of the incumbent solution, and for its inverse $\overleftarrow{S}$. This procedure contains a main loop which scans each multitrip and each node i in this multitrip, while a nested loop inspects each node j from i onward in the same multitrip.

For $i = j$, we have $|S| = |\overleftarrow{S}| = 1$ and $D(S) = D(\overleftarrow{S}) = C(S) = C(\overleftarrow{S}) = 0$. When j is incremented, the attributes for the new sequence are derived in $O(1)$ from the ones for S, using equations (22)–(24) with $S' = (j+1)$. For instance, $C(S)$ is computed via equation (24): $C(S \oplus S') = C(S) + |S'| \cdot (D(S) + w_{j,j+1}) + C(S')$ $= C(S) + D(S) + w_{j,j+1}$. For the reverse sequence, we use the same formula but add $j+1$ at the beginning, i.e., $C(\overleftarrow{S \oplus S'}) = C(\overleftarrow{S'} \oplus \overleftarrow{S}) = C(\overleftarrow{S'}) + |\overleftarrow{S}| \cdot (D(\overleftarrow{S'}) + w_{j+1,i}) + C(\overleftarrow{S}) = |S| \cdot w_{j+1,i} + C(\overleftarrow{S})$.

In practice, we store the attributes in matrices, using the first node and last node of each sequence as subscripts. For instance, the value of $C(S)$ for $S = (i, i+1, \ldots, j)$ is stored in a matrix $\widetilde{C}$, in element $\widetilde{C}_{ij}$. As each attribute is calculated in $O(1)$, it is clear that all precomputations can be done in $O(n^2)$. After a call to *Precomputations*, each perturbation and each move in the VND can be evaluated in $O(1)$. As accepted perturbations and moves change the incumbent solution, the attributes must be updated. This task is carried out by the *Update_Precomputations* procedure, similar to *Precomputations* but restricted to the sequences contained in modified multitrips.

4.4 Dominance rule

The following dominance rule shows that it is not necessary to consider all possible trip orderings in a given multitrip.

Theorem *Define the* mean duration *of a trip k as its total duration $D(k)$ divided by its number of required nodes $|k|$. The cost of a multitrip is minimized by ordering its trips in non-decreasing order of mean trip duration.*

Proof Consider a multitrip h which does not follow this rule. It contains two consecutive trips k and k' such that the mean duration of k is larger than the one of k'. Let Z_k^h be the sum of arrival times of trip k in multitrip h, t_0^k its starting time, and t_i the arrival time at site i (if trip k starts at t_0^k). The two trip costs can be written as follows:

$$Z_k^h = \sum_{i \in k} (t_0^k + t_i) = |k| \cdot t_0^k + \sum_{i \in k} t_i \tag{25}$$

and

$$Z_{k'}^h = \sum_{i \in k'} (t_0^{k'} + t_i) = |k'| \cdot t_0^{k'} + \sum_{i \in k'} t_i \tag{26}$$

As k is ordered before k', we can take without loss of generality $t_0^k = 0$ and $t_0^{k'} = D(k)$. Then the total cost is:

$$Z_k^h + Z_{k'}^h = \sum_{i \in k} t_i + |k'| \cdot D(k) + \sum_{i \in k'} t_i \tag{27}$$

If k and k' are swapped to obtain a new multitrip h' which does not affect the trips before k or after k', then $t_0^{k'} = 0$, $t_0^k = D(k')$, and the sum of arrival times for the two trips becomes:

$$Z_k^{h'} + Z_{k'}^{h'} = |k| \cdot D(k') + \sum_{i \in k} t_i + \sum_{i \in k'} t_i \tag{28}$$

The cost difference is:

$$(Z_k^h + Z_{k'}^h) - (Z_k^{h'} + Z_{k'}^{h'}) = |k'| \cdot D(k) - |k| \cdot D(k') \tag{29}$$

But the initial assumption states that the mean duration of k' is less than the one of k:

$$\frac{D(k)}{|k|} > \frac{D(k')}{|k'|} \tag{30}$$

Therefore:

$$(Z_k^h + Z_{k'}^h) - (Z_k^{h'} + Z_{k'}^{h'}) > 0 \tag{31}$$

which shows that the total cost is reduced by swapping two consecutive trips that are not ordered in non-decreasing order of mean trip duration. □

Thanks to this dominance rule, the optimal cost for a multitrip with v given trips can be obtained by sorting the routes instead of evaluating the $v!$ possible orders. Preliminary tests have shown that it is still time-consuming to enforce the dominance rule at each step of our metaheuristic. In practice it is applied to the initial solution of each ILS and periodically in the VND, as explained for neighborhood N_4 in Sect. 4.6.

4.5 Initial solutions

Each ILS in our MS-ILS starts with an initial solution computed by a greedy randomized heuristic that builds one route at a time. Each new route is initialized with the farthest unserviced site. Then, all feasible insertions of remaining sites in the incumbent routes are evaluated. A restricted candidate list (RCL) gathers the sites satisfying Eq. (32), where V'' is the set of unserviced sites, $z(i)$ the insertion cost of site i in the emerging trip, and z_{max} and z_{min} the largest and smallest insertion costs. One site is randomly selected in the RCL for a real insertion. The incumbent trip is finalized when all demand nodes are serviced or when all possible insertions of remaining demand nodes violate vehicle capacity.

$$RCL = \{i \in V'' \mid z(i) \le z_{min} + 0.1 \times (z_{max} - z_{min})\} \tag{32}$$

Finally, R empty multitrips are prepared, the trips are sorted in non-decreasing order of mean duration $D(k)/|k|$ according to the dominance rule, and added one by one at the end of the multitrip with minimum duration.

4.6 Variable neighborhood descent

The improvement procedure used in our MS-ILS is a variable neighborhood descent (VND). VND is in itself a simple metaheuristic, based on p neighborhoods N_k, $k = 1, 2, \ldots, p$. Each neighborhood is implicitly defined by a type of move. Starting from $k = 1$ and one input solution s', the basic iteration of VND consists in exploring the neighborhood N_k of s'. We selected the first improvement mode: as soon as a better solution is discovered, it becomes the incumbent solution and k is reset to 1, otherwise k is incremented. The procedure stops when the exploration of N_p brings no improvement. The method is sketched in Algorithm 2.

Our VND for the mt-CCVRP is based upon $p = 5$ neighborhoods, each of them being implicitly defined by the union of several elementary moves. N_1 involves 2-opt moves and λ-interchanges, adapted to our objective function and applied to individual trips. N_2 considers the same moves, but applied to two trips in the same multitrip. In N_3 the same moves are attempted on two trips of different vehicles. N_4 and N_5 respectively consist in splitting a trip and reordering the trips of a multitrip. Only feasible solutions are considered, i.e., all moves must respect vehicle capacity and fleet size.

As we test also the reversal of each moved sequence in the reinsertions, each classical CVRP move like 2-opt is here declined into several cases. As it would take too much room to give the evaluation formula for each move and each of its cases, we will give the equation only for two moves in the sequel, knowing that the transposition to the other cases is easy using our pre-computed attributes. Without the pre-processing, the cost variation formulas would be very involved, especially for moves operating on two multitrips.

Algorithm 2 – VND for a solution s'

$k \leftarrow 1$
while $k \le p$ **do**
  search for the first solution $s'' \in N_k(s')$ such that $Z(s'') < Z(s')$
  if s'' exists **then**
    $s' \leftarrow s''$
    $k \leftarrow 1$
    Update_Precomputations (s')
  else
    $k \leftarrow k + 1$
  end if
end while

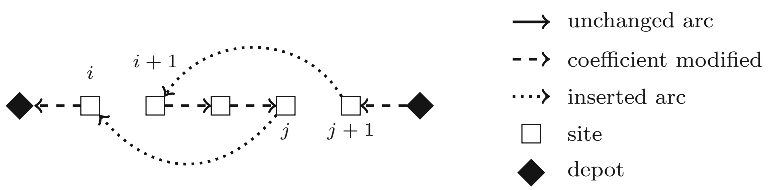

Fig. 2 2-opt move within a route

4.6.1 N_1 : 2-OPT moves and λ-interchanges on one trip

The 2-opt move on one trip was already used by Ngueveu et al. [54] for the CCVRP. In our case, two arcs $(i, i+1)$ and $(j, j+1)$ are deleted in one trip T of one multitrip M and the fragments are reconnected via two new arcs. Equivalently, it can be defined as the reversal of sequence $(i+1, \ldots, j)$. Let S_1 be the beginning of the multitrip M up to node i included, $S_2 = (i+1, \ldots, j)$, and S_3 the rest of M. The cost variation can be computed using Eq. (33):

$$\Delta Z = C(S_1 \oplus \overleftarrow{S_2} \oplus S_3) - C(S_1 \oplus S_2 \oplus S_3) \tag{33}$$

As the cost of a trip changes when inverted, we consider also the new variant of Fig. 2. Let S_1 be the sequence containing the trips before trip T, S_2 the beginning of T up to node i included, $S_3 = (i+1, \ldots, j)$, S_4 the end of T, and S_5 the rest of M. Contrary to the standard 2-opt move, S_2 and S_4 are reversed, but not S_3. The cost variation is the following:

$$\Delta Z = C(S_1 \oplus \overleftarrow{S_4} \oplus S_3 \oplus \overleftarrow{S_2} \oplus S_5) - C(S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5) \tag{34}$$

The other moves of N_1 are λ-interchanges. They consist in exchanging a sequence with one to λ consecutive nodes with another (non-overlapping) sequence containing zero to λ consecutive nodes. The two subsequences must belong to the same trip T but may have different lengths. Each subsequence with more than one node can be reversed in the reinsertion, giving four cases. We allow a null length for the second sequence, to include relocations of the first sequence as particular case.

Figure 3 depicts the case where no sequence is reversed. Let S_1 be the beginning of multitrip M up to node i, $S_2 = (i+1, \ldots, j)$, $S_3 = (j+1, \ldots, k)$, $S_4 = (k+1, l)$ and S_5 the rest of the multitrip. The cost variation if S_2 and S_4 are exchanged is the following:

$$\Delta Z = C(S_1 \oplus S_4 \oplus S_3 \oplus S_2 \oplus S_5) - C(S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5) \tag{35}$$

In all cases, the cost variation can be computed in constant time using the operator $\oplus$. It is easy to see that N_1 can be explored in $O(n^2/\tau)$, where τ is the number of trips in the incumbent solution.

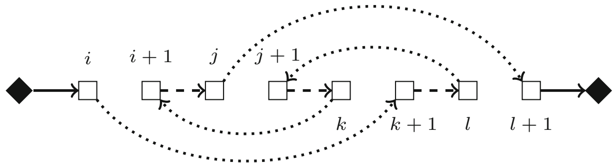

Fig. 3 λ-interchange within a route

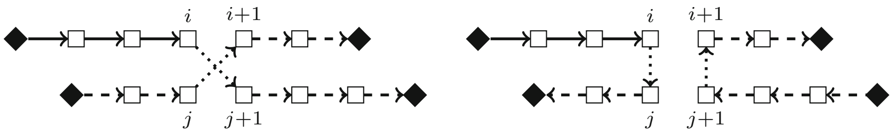

Fig. 4 2-opt move on two trips done by the same vehicle

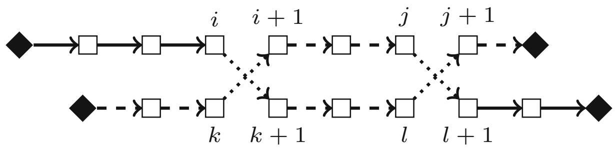

Fig. 5 λ-interchange on two trips done by the same vehicle

4.6.2 N_2 : 2-OPT moves and λ-interchanges involving two trips in a multitrip

Figure 4 depicts the two kinds of 2-OPT moves for two routes T and T' contained in the same multitrip. The λ-interchanges now involve one subsequence in each trip, like in Fig. 5. To avoid testing twice the same exchange, N_2 is searched with T before T'. This detail is important to interpret the two figures, as the line styles used (coefficients changed or not, see Fig. 2) depend on the relative order of the two trips. As we allow the reversal of T and T' in these transformations, eight cases must be evaluated for 2-OPT moves and four for λ-interchanges. The evaluation of each case is still possible in $O(1)$ using the pre-computations, and N_2 can be searched in $O(n^2/R)$.

4.6.3 N_3 : 2-OPT moves and λ-interchanges on two trips done by distinct vehicles

Moves affecting two trips pertaining to two different multitrips are similar to the ones browsed in N_1 and N_2 but require more complicated cost variation formulas since two multitrips are modified. The moves in N_3 bring deeper modifications to the incumbent solution, explaining why this neighborhood is searched after N_1 and N_2 by the VND. N_3 can be browsed in $O(n^2)$.

4.6.4 N_4 : Trip reordering

According to the dominance rule presented in Sect. 4.4, a solution can be improved by sorting the trips of each multitrip in non-decreasing order of mean duration. During preliminary tests we tried to evaluate the moves in N_1, N_2 and N_3 while reordering simultaneously the affected trips, but the metaheuristic became too time-consuming. In fact, the quality of final solutions is practically the same if the trips are reordered only when the three previous neighborhoods bring no improvement.

Although this reordering phase is called N_4 in the VND, no real neighborhood is searched since the trips are sorted systematically in each multitrip. This process can be executed in $O(Rm \log m)$, where R is the number of multitrips and $m = (\sum_{i=1}^{n} q_i)/(QR)$ is the average number of trips per multitrip.

4.6.5 N_5 : *Trip splitting*

This neighborhood is based on an adaptation of a route-splitting procedure called *meiosis* and designed by Petch and Salhi for the multitrip VRP [59]. This procedure determines the shortest multitrip p, inspects each multitrip $k \neq p$ and evaluates the cost variation when (a) the last trip of k is cut after the first, second, ..., or last but one customer, then (b) the rest of the trip is moved at the end of multitrip p. This process which can be implemented in $O(Rn)$ is interrupted as soon as a saving is obtained. In case of improvement, the VND goes back to neighborhood N_1, otherwise it terminates.

4.7 Perturbation

The perturbation applied before calling the VND consists in swapping three pairs of nodes randomly selected. The nodes in each pair may belong to the same trip, to two trips of the same multitrip, or to two multitrips, but vehicle capacity must be respected. It is interesting to observe that this kind of move is tested in the neighborhoods N_1, N_2 and N_3, so the effect of the perturbation could be theoretically cancelled in the VND that follows. In fact, the perturbation is sufficient to change the trajectory of the VND which uses a first-improvement implementation instead of a best improvement strategy. We tried to use in the perturbation different moves not used in the VND, like random 4-OPT moves, but without getting better results on average.

5 Computational experiments

5.1 Implementation and instances

The MS-ILS algorithm is implemented in Visual C++ and the mathematical model in CPLEX 12.4. Both have been tested on a 2.50 GHz Intel Core i5 computer with 4 GB of RAM and Windows 7 Professional. Three sets of experiments are reported. The first one compares the resolution of the 0–1 MILP via a commercial solver (CPLEX) with the hybrid metaheuristic. Obviously, such a comparison is only possible on small instances. The second set of experiments applies the metaheuristic to the CCVRP, the particular case of the mt-CCVRP with one trip per multitrip, because three published metaheuristics can be used for comparison. Finally, the third set consists in evaluating our MS-ILS on the mt-CCVRP.

The comparison with CPLEX involves 12 randomly generated instances with 15 demand nodes. For the CCVRP, we use the same instances as the competitors, derived from classical CVRP instances: the 14 CMT instances [24], with $n = 50$ to 199 demand nodes, and the 20 GWKC larger instances [33], with $n = 200$ to 483. To get CCVRP problems, the fleet size is fixed (otherwise the optimum is trivial, with one direct route to each demand node) and the maximum trip length defined for some instances is ignored. As the seven CMT instances with a trip length constraint share the same network as the seven unconstrained instances, $7+20 = 27$ CCVRP instances are finally obtained. In all instances the traveling time w_{ij} on arc (i, j) is equal to the Euclidean distance, computed as a double-precision real number.

We took the same 27 instances for the mt-CCVRP, but reducing the number of vehicles to force the metaheuristic to build several trips per vehicle. For the sake of simplicity, we still call the resulting data files CMT and GWKC instances in the sequel.

5.2 Parameter tuning

A big advantage of MS-ILS is to have only three parameters: the number of successive starts ($MaxStart$), the number of iterations per ILS ($MaxIter$), and the maximum number of consecutive sites in λ-interchange moves (λ). As the running time of MS-ILS is roughly proportional to the number of calls to the VND, we decided to allocate a fixed "computing budget" of 5,000 calls to avoid excessive execution times. Like Prins [63], who developped an MS-ILS for the CVRP, we observed that the partition of the calls among several restarts is critical, with two extreme options: using $MaxStart = 1$, the metaheuristic reduces to a single ILS with 5,000 iterations, while for $MaxIter = 1$ it becomes a GRASP with 5,000 independent iterations. For our problem the best results on average are obtained using $MaxStart = 5$ and $MaxIter = 1{,}000$.

The reported MS-ILS solution values for each instance are the average and best solution cost over five runs, while the running time is always the average time per run. For the CCVRP, our metaheuristic lasts longer than the competitors because it is designed for a more general problem: to limit the running time on these instances, we have preferred to reduce the number of successive ILS to two, while keeping five runs.

5.3 Results on small mt-CCVRP instances

Table 2 presents the results for the 0-1 mixed integer linear program and the MS-ILS. The first four columns display the instance name, the number of required nodes, the number of vehicles, and the average number of trips per vehicle $\sum_{i=1}^{n} q_i/(QR)$. For CPLEX we present two kind of results, first without valid inequalities (13) to (16) (0–1 MILP[1]) and then with valid inequalities (0–1 MILP[2]). For both CPLEX results we provide the cost of the linear relaxation of the mathematical model (LR), the solution value and the running time in seconds, limited to one hour. When CPLEX must be interrupted, the best lower bound and the cost of the best integer solution found are reported. The best solution value over five runs and the average running time per run in seconds are indicated for MS-ILS. The last column gives the percentage deviation

Table 2 Tests on small mt-CCVRP instances

File	n	R	$\frac{\sum q_i}{QR}$	0–1 MILP[1]			0–1 MILP[2]			MS-ILS (5 runs)		Gap
				LR	Cost	Time	LR	Cost	Time	Best cost	Time	
$Test_{01}$	15	4	1.00	588.75	687.29	15.97	619.04	687.29	8.66	687.29	1.95	0.00
$Test_{02}$	15	4	1.25	605.08	741.91	77.63	633.47	741.91	33.99	741.91	2.76	0.00
$Test_{03}$	15	4	1.67	661.11	855.91	260.60	686.32	855.91	38.38	855.91	1.96	0.00
$Test_{04}$	15	4	2.50	836.57	1,090.67	264.80	851.99	1,090.67	24.20	1,090.67	1.41	0.00
$Test_{05}$	15	3	1.11	632.83	817.22	138.95	697.26	817.22	16.13	817.22	4.78	0.00
$Test_{06}$	15	3	1.33	643.13	942.45	2,943.82	708.68	942.45	414.64	942.45	4.28	0.00
$Test_{07}$	15	3	1.67	674.25	(941.99/1,008.03)	–	733.24	1,008.03	560.17	1,008.03	2.32	0.00
$Test_{08}$	15	3	2.22	749.57	(1,035.92/1,111.44)	–	788.88	1,111.04	124.54	1,111.04	2.53	0.00
$Test_{09}$	15	2	1.25	687.48	(1,070.01/1,201.30)	–	802.20	(1,116.97/1,182.66)	–	1,182.66	5.45	5.88
$Test_{10}$	15	2	1.67	702.20	(1,059.41/1,310.17)	–	814.44	(1,100.45/1,327.76)	–	1,310.17	3.69	19.06
$Test_{11}$	15	2	2.00	720.16	(1,101.58/1,403.74)	–	830.58	(1,199.17/1,391.60)	–	1,391.60	3.49	16.05
$Test_{12}$	15	2	2.50	761.58	(1,164.97/1,513.06)	–	862.40	(1,296.97/1,513.06)	–	1,513.06	1.57	16.66
Mean						2,108.48			1,301.73		3.02	4.80

of the best MS-ILS cost $Z(MS\text{-}ILS)$ to the optimum $Z(MILP)$ of CPLEX (or to the best lower bound when the solver must be stopped), computed as $(Z(MS\text{-}ILS)/Z(MILP) - 1) \times 100$.

The results indicate that the relaxed LP yields acceptable lower bounds, a good sign for a future study on valid inequalities. For the eight first instances, CPLEX finds an optimal solution, retrieved in all cases by MS-ILS. The model with valid inequalities outperforms the model without them: all linear relaxations are improved, two more optimal solutions are found in less than one hour, the running times are decreased when optimal solutions are found by both models, and the gap between the best integer solution and the best lower bound is reduced when time limit is reached. The instances look harder when the number of vehicles decreases and the average number of trips per vehicle increases: the running time of CPLEX augments quickly and the four last instances cannot be solved in one hour. However, our metaheuristic always returns a solution in at most 5.5 s (3 on average). As the gaps are computed in these cases using the lower bound found by CPLEX when its time limit is reached, they are probably overestimated: some MS-ILS solutions for instances test_{09} to test_{12} are perhaps optimal but we cannot prove it.

5.4 Results on the CCVRP

Four metaheuristics have been published for the CCVRP: the memetic algorithms MA1 and MA2 of Ngueveu et al. [54], the adaptive large neighborhood search (ALNS) of Ribeiro and Laporte [65] and the two-phase metaheuristic T-PM of Ke and Feng [43]. All these authors solved the CMT and GWKC instances using 5 runs, except Ngueveu et al. who tackled only the CMT problems, and with a single run. However, for a fair comparison with ALNS, Ribeiro and Laporte asked the codes of MA1 and MA2 to Ngueveu et al. to solve five times the CMT and GWKC instances on their computer. We use these recomputed costs in our comparison.

MS-ILS must be modified to use at most one trip per vehicle and to stop using the trip splitting moves of neighborhood N_5. On CCVRP instances, the initial constructive heuristic of Sect. 4.5 builds in general no more than R trips and the final step that packs these trips into multitrips is no longer used. The following simple procedure is applied when R trips are not sufficient: (a) the node u not yet serviced with largest demand is determined; (b) if u cannot be feasibly inserted, demand nodes are removed at random from the R trips until feasible insertions appear; (c) the best possible insertion of u in the R trips is performed. This process, repeated until all sites are routed, never fails on the instances tested.

Apart from the suppression of N_5 and the modification of the initial heuristic, the code of MS-ILS remains identical when applied to the CCVRP: for instance, we did not try to simplify the complex moves of the VND.

Table 3 compares the four published algorithms with MS-ILS on the seven CMT instances and the 20 GWKC larger instances. The four first column headers correspond to the instance name, the number of demand nodes n, the number of vehicles used R, and the best known solution (BKS) found using all heuristics. For each algorithm are indicated the best and average deviation to the best known solution over 5 runs

Table 3 Results for CMT instances [24] and GWKC instances [33] for the CCVRP

Instance	n	R	BKS	MA1			MA2			ALNS			T-PM			MS-ILS		
				BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time
CMT_{01}	50	5	2,230.35	**0.00**	0.00	10.63	**0.00**	0.69	3.70	**0.00**	0.22	30.29	**0.00**	0.75	17.64	**0.00**	0.34	19.29
CMT_{02}	75	10	2,391.63	1.27	2.15	27.78	1.57	6.88	2.04	**0.00**	0.42	60.77	**0.00**	1.62	22.48	**0.00**	1.00	25.22
CMT_{03}	100	8	4,045.41	0.68	0.68	97.91	0.68	0.85	40.46	**0.00**	0.46	172.45	**0.00**	0.46	60.96	**0.00**	0.44	141.17
CMT_{04}	150	12	4,987.51	**0.00**	0.67	449.44	**0.00**	0.19	188.41	**0.00**	0.15	235.12	**0.00**	0.28	92.68	**0.00**	0.59	244.09
CMT_{05}	199	17	5,809.59	0.01	0.56	1,035.45	0.01	0.54	629.27	0.49	0.83	277.37	**0.00**	0.81	135.36	0.38	0.71	311.88
CMT_{11}	120	7	7,314.55	0.05	1.11	160.64	0.45	1.11	68.83	0.02	0.37	202.07	**0.00**	0.27	71.64	**0.00**	0.27	136.15
CMT_{12}	100	10	3,558.92	**0.00**	0.01	38.20	0.01	0.01	23.66	**0.00**	0.20	152.74	**0.00**	0.06	53.72	**0.00**	0.08	128.50
Average				0.29	0.74	260.01	0.39	1.47	136.62	0.07	0.38	161.54	0.00	0.61	64.93	0.05	0.49	143.76
$GWKC_{01}$	240	9	54786.92	0.24	0.36	1,593.60	0.26	0.43	866.80	0.19	0.31	1,038.27	**0.00**	0.13	347.63	0.40	0.57	1,188.64
$GWKC_{02}$	320	10	100,621.30	0.17	0.25	4,549.05	0.14	0.26	2,054.68	**0.00**	0.27	1,484.74	**0.00**	0.12	848.55	0.12	0.14	5,491.50
$GWKC_{03}$	400	10	171,027.05	0.14	0.21	9,295.64	0.16	0.28	3,336.09	0.34	0.70	2,061.75	**0.00**	0.17	1,466.81	**0.00**	0.09	11,394.18
$GWKC_{04}$	480	10	262,049.31	0.17	0.26	12,810.31	0.19	0.33	6,557.67	0.49	1.17	2,626.89	**0.00**	0.17	1,946.57	0.08	0.09	10,181.15
$GWKC_{05}$	200	5	114,163.52	**0.00**	0.06	471.45	**0.00**	0.15	246.05	0.29	0.60	1,200.67	**0.00**	0.01	349.82	0.03	0.05	2,004.68
$GWKC_{06}$	280	7	140,429.95	**0.00**	0.02	2,358.40	0.02	0.09	788.71	0.27	0.36	1,547.43	**0.00**	0.07	548.18	0.04	0.09	4,534.21
$GWKC_{07}$	360	8	180,481.56	1.55	3.45	1,537.36	5.48	6.70	258.45	**0.00**	0.63	1,926.19	1.68	2.10	429.32	3.28	4.77	1,703.93
$GWKC_{08}$	440	10	194,100.64	0.24	0.34	9,598.42	0.22	0.31	5,165.83	0.58	0.68	2,330.34	**0.00**	0.16	1,493.28	0.38	0.43	5,376.19
$GWKC_{09}$	255	14	4,725.58	0.15	0.35	2,453.61	0.29	0.43	1,582.86	0.04	0.09	864.89	**0.00**	0.08	310.34	0.57	0.67	1,002.82
$GWKC_{10}$	323	16	6,713.92	0.27	0.49	7,652.00	0.11	0.43	4,332.24	**0.00**	0.06	1,092.34	0.02	0.08	484.25	0.29	0.38	2,052.32
$GWKC_{11}$	399	18	9,214.07	0.31	0.49	8,835.74	0.29	0.47	7,429.60	**0.00**	0.03	1,356.99	0.04	0.12	709.16	0.37	0.40	3,663.71
$GWKC_{12}$	483	19	12,526.17	0.82	0.98	9,881.24	1.06	1.46	3,449.06	**0.00**	0.13	1,540.32	0.19	0.35	829.51	1.20	1.57	2,836.79
$GWKC_{13}$	252	26	3,628.30	0.68	0.90	2,012.17	1.61	9.31	74.31	**0.00**	0.28	632.72	0.20	0.77	195.68	1.20	1.34	300.69
$GWKC_{14}$	320	29	5,216.80	10.60	15.88	2,263.84	11.69	23.07	9.59	**0.00**	0.79	682.19	1.40	1.78	158.83	2.64	3.00	254.69

Table 3 continued

Instance	n	R	BKS	MA1			MA2			ALNS			T-PM			MS-ILS		
				BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time
$GWKC_{15}$	396	33	7,010.41	0.96	1.85	5,597.98	22.34	22.34	0.00	**0.00**	0.18	855.25	0.24	0.93	798.44	1.48	1.83	627.96
$GWKC_{16}$	480	37	9,250.98	0.54	0.96	16,204.52	1.04	1.83	5,307.29	**0.00**	0.19	1,104.53	0.22	0.46	1,218.37	1.31	1.45	949.39
$GWKC_{17}$	240	22	3,065.46	0.83	1.29	1,631.86	1.01	1.74	431.31	0.03	0.12	618.70	**0.00**	0.47	345.22	1.11	1.25	235.96
$GWKC_{18}$	300	27	4,221.14	7.27	8.56	2,410.31	20.43	21.72	0.10	**0.00**	0.56	630.47	2.13	3.10	405.28	3.39	4.28	240.15
$GWKC_{19}$	360	33	5,523.38	1.03	1.37	5,535.25	1.57	2.14	763.56	0.18	0.33	853.43	**0.00**	0.37	630.62	0.61	0.93	432.90
$GWKC_{20}$	420	38	7,223.08	2.64	3.47	6,264.59	17.04	17.04	0.00	**0.00**	0.25	881.92	0.20	1.11	725.80	1.62	1.79	449.05
Average				1.43	2.08	5,647.86	4.25	5.53	2,132.71	0.12	0.39	1,266.50	0.32	0.63	712.08	1.01	1.26	2,746.05

Null deviations in boldface indicates best-known solutions achieved by the algorithms
The costs and times reported for MA1 and MA2 are the ones recomputed in [65], see text
Best and average deviations given for 5 runs. Average times per run for a 2 GHz PC except for MS-ILS, tested on a 2.5 GHz PC

(BestD and AvgD), and the average computational time in seconds (Time). Best known solutions have a null deviation, displayed in boldface.

On the seven CMT instances, our approach retrieves six best known solutions. The running time increases quickly with instance size, but remains comparable with the durations of the other heuristics. The only instance where we fail to find the BKS is the largest one (199 demand nodes). Considering performance indicators averaged on the seven instances, the execution time is less than two and a half minutes (variation between 19 s and 5 min) and the best and average deviations are quite small (0.05 and 0.49 %). Moreover, our best deviation is outperformed by T-PM only and MS-ILS yields the second best average deviation after ALNS.

On the 20 larger GWKC instances, MS-ILS finds only one best known solution. TP-M and ALNS are better in terms of solution quality and running time, but our metaheuristic is still better than MA1 and MA2 for these two criteria. Concerning the results averaged over the 20 instances, the mean execution time is around 45 min, varying between 4 and 189 min, the best deviation is 1.01 % and the mean deviation 1.26 %. These results are honorable, keeping in mind that MS-ILS is designed for a more general problem (the mt-CCVRP) and tested here without simplifying its code.

5.5 Results on mt-CCVRP

Our results for mt-CCVRP are presented in Table 4 for CMT instances and in Table 5 for GWKC instances, using the same columns: instance name, number of nodes n, number of vehicles R, average number of trips per vehicle $m = \sum_{i=1}^{n} q_i/(QR)$, best solution found in 5 runs (BS), average deviation of the 5 solutions to BS in percent (D_{avg}), average duration per run in seconds (*Time*) and maximum deviation to BS in percent (D_{max}). We recall that the CMT and GWKC instances for the mt-CCVRP are identical to the CCVRP versions, except a reduced fleet size to get several trips per vehicle.

On the CMT instances, the average computational time is less then 6 min, varying between 1.14 and 12.51 min. A comparison with Table 3 (CCVRP) shows that the running time is roughly multiplied by 2.5 when multitrips must be handled. Over the

Table 4 Results for the mt-CCVRP using the 7 CMT instances [24]

Instance	n	R	m	BS	D_{avg}	Time	D_{max}
CMT_{01}	50	3	1.62	3,856.39	0.16	78.26	0.35
CMT_{02}	75	3	3.25	8,300.15	0.04	68.37	0.21
CMT_{03}	100	3	2.43	10,957.00	0.42	238.07	1.04
CMT_{04}	150	3	3.73	20,599.00	0.25	479.34	0.48
CMT_{05}	199	3	5.31	34,044.80	0.48	750.80	1.94
CMT_{11}	120	3	2.29	15,797.40	0.17	470.41	0.36
CMT_{12}	100	3	3.02	10,658.70	0.00	329.87	0.00
Average					0.22	345.02	0.63

seven instances, the average and maximum solution costs for five runs are very close to the best cost (0.22 and 0.63 %), indicating that MS-ILS is fairly robust.

The average and worst deviations increase moderatly on the GWKC instances, with 0.36 and 0.85 %, respectively. The average computational time is 78 min, ranging from 13 to 265 min. The proposed metaheuristic is still relatively stable in terms of solution quality but its running time becomes important on the largest instances with 480 demand nodes. Moreover, the execution times vary a lot among instances of the same size: for example, instances $GWKC_{01}$ and $GWKC_{17}$ contain 240 demand nodes but require respectively 2,443 and 791 s of processing time.

This variation is mainly due to the neighborhood N_1 in which in the VND spends most of its time. Indeed, we saw in Sect. 4.6 that $|N_1| = O(n^2/T)$, where the number of trips T can be computed as $T = Rm$ from Table 5: $GWKC_{01}$ and $GWKC_{17}$ respectively require 8.72 and 21.6 trips.

Figure 6 shows the evolution of the average deviation to the best known solution during the iterations of MS-ILS, on the CMT instances, using a logarithmic scale for the deviation axis. For each number of iterations, the deviation reported is the average of 35 values (7 instances and 5 runs per instance). The initial deviation is 7.78 % for the first local optimum, obtained by the greedy randomized heuristic followed by

Table 5 Results for the mt-CCVRP using the 20 GWKC instances [33]

Instance	n	R	m	BS	D_{avg}	Time	D_{max}
$GWKC_{01}$	240	4	2.18	123,386.93	0.26	2,443.14	0.83
$GWKC_{02}$	320	4	2.29	254,585.38	0.23	5,330.64	0.43
$GWKC_{03}$	400	4	2.22	435,821.34	0.25	10,123.85	0.62
$GWKC_{04}$	480	4	2.40	676,703.13	0.44	15,887.69	0.78
$GWKC_{05}$	200	3	1.48	193,339.55	0.15	3,018.47	0.58
$GWKC_{06}$	280	3	2.07	335,034.88	0.22	5,846.47	0.62
$GWKC_{07}$	360	3	2.67	493,826.53	0.34	9,144.21	0.88
$GWKC_{08}$	440	5	1.96	399,995.84	0.42	13,280.66	0.88
$GWKC_{09}$	255	5	2.69	13,784.33	0.32	1,433.82	1.26
$GWKC_{10}$	323	6	2.53	18,443.35	0.65	2,374.63	1.04
$GWKC_{11}$	399	7	2.43	24,271.05	0.33	3,920.85	1.26
$GWKC_{12}$	483	8	2.34	31,144.78	0.68	5,831.21	1.41
$GWKC_{13}$	252	10	2.51	10,081.61	0.26	913.49	0.78
$GWKC_{14}$	320	12	2.39	13,457.03	0.24	1,687.19	0.43
$GWKC_{15}$	396	15	2.15	16,408.36	0.16	2,263.48	0.31
$GWKC_{16}$	480	15	2.39	24,188.15	0.39	3,737.54	0.79
$GWKC_{17}$	240	10	2.16	6,618.18	0.63	791.04	1.20
$GWKC_{18}$	300	12	2.25	9,385.02	0.43	1,265.00	1.01
$GWKC_{19}$	360	12	2.70	14,894.56	0.33	2,032.59	0.86
$GWKC_{20}$	420	15	2.52	17,939.90	0.49	2,554.13	1.06
Average					0.36	4694.01	0.85

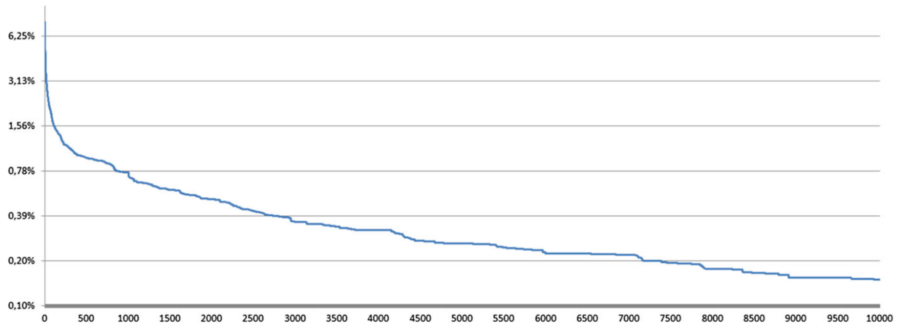

Fig. 6 Evolution of average deviation to best known solution on CMT instances

Table 6 Improvement of initial solution (average of 5 runs)

Instance	Initial gap to BS in %	After 1st call to VND	After 1000 calls	After 5000 calls
CMT_{01}	19.61	4.12	0.35	0.16
CMT_{02}	36.08	8.20	0.36	0.04
CMT_{03}	35.13	6.41	1.02	0.42
CMT_{04}	43.38	8.93	0.55	0.25
CMT_{05}	39.12	9.02	1.80	0.48
CMT_{11}	63.30	7.40	0.42	0.17
CMT_{12}	53.52	10.39	0.88	0.00
Average	41.45	7.78	0.77	0.22

one call to the VND. A major part of the descent is accomplished using the allocated budget of 5,000 iterations: the deviation reduces to 0.22 %. Additional iterations bring only a minor improvement, at the expense of larger running times.

Table 6 details the deviations for each instance and adds the deviation of the initial heuristic solution. As it can be observed, the deviation is below 1 (0.77 %) after 1,000 iterations versus 0.22 % after 5,000. Hence, a first way to reduce the large running times on GWKC instances (up to 15,000 s) is to use 1,000 iterations only : in that case no running time would exceed 50 min. A much faster option is to run the initial constructive heuristic and improve it using a single call to the VND, which gives an average deviation of 7.78 % in at most 3 s per run, on all instances tested.

6 Conclusions

This paper introduces a new problem, the multi-trip cumulative capacitated vehicle routing problem (mt-CCVRP). This problem constitutes a good way to model the delivery of relief supplies via copters after a humanitarian disaster, when the number of copters is limited and the time to reach affected areas is critical. A non-trivial mathematical model without vehicle nor multitrip indexes is proposed and tested on small instances. The article develops also the first metaheuristic to solve the mt-CCVRP, by hybridizing a multi-start ILS with a VND which evaluates each move in constant time.

On small instances, the resulting algorithm MS-ILS finds the same results as the mathematical model when the latter can be solved to optimality. MS-ILS competes with published methods for the case without multitrips, the CCVRP. Although the running time becomes important on the largest instances close to 500 demand nodes, we have shown that the fast descent can be truncated to get good solutions in a reduced duration.

More work is still needed to get closer to the real constraints encountered in disaster logistics. One promising extension is the Generalized CCVRP, in which the relief supplies must be delivered to one airport to be selected among the ones that are still operational in each region. Another interesting variant is a bi-objective problem over a multiperiod horizon, with equity constraints to try to satisfy at the same level the needs of affected sites. As the quantities shipped can be important in these two extensions, split deliveries should be allowed for a better use of vehicles. Such deliveries raise subtle complications. For instance, the arrival time for the first delivery at a site, the arrival time for the last delivery or the average arrival time of all deliveries produce different results. Moreover, a local search able to modify the number of deliveries and the amounts delivered will be certainly more involved.

References

1. About disasters.: International Federation of Red Cross and Red Crescent Societies (IFRC), http://www.ifrc.org/en/what-we-do/disaster-management/about-disasters/ (2011)
2. Altay, N., Green III, W.G.: OR/MS research in disaster operations management. Eur. J. Oper. Res. **175**(1), 475–493 (2006)
3. Applegate, D., Cook, W., Dash, S., Rohe, A.: Solution of a min-max vehicle routing problem. INFORMS J. Comput. **14**(2), 132–143 (2002)
4. Archer, A., Williamson, D.P.: Faster approximation algorithms for the minimum latency problem. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (2003)
5. Bakuli, D.L., Smith, J.M.: Resource allocation in state-dependent emergency evacuation networks. Eur. J. Oper. Res. **89**(3), 543–555 (1996)
6. Balcik, B., Beamon, B.M.: Facility location in humanitarian relief. Int. J. Logist. Res. Appl. **11**(2), 101–121 (2008)
7. Balcik, B., Beamon, B.M., Smilowitz, K.: Last mile distribution in humanitarian relief. J. Intell. Transp. Syst. **12**, 51–63 (2008)
8. Barbarosoğlu, G., Arda, Y.: A two-stage stochastic programming framework for transportation planning in disaster response. J. Oper. Res. Soc. **55**(1), 43–53 (2004)
9. Barbarosoğlu, G., Özdamar, L., Çevik, A.: An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. Eur. J. Oper. Res. **140**(1), 118–133 (2002)
10. Beamon, B.M.: Humanitarian relief chains: issues and challenges. In: Proceedings of the 34th International Conference on Computers and Industrial Engineering (2004)
11. Beamon, B.M., Kotleba, S.A.: Inventory management support systems for emergency humanitarian relief operations in South Sudan. Int. J. Logist. Manag. **17**(2), 187–212 (2006)
12. Beamon, B.M., Kotleba, S.A.: Inventory modeling for complex emergencies in humanitarian relief operations. Int. J. Logist. Res. Appl. **9**(1), 1–18 (2006)
13. Bennett, B.T., Gazis, D.C.: School bus routing by computer. Transp. Res. **6**(4), 317–325 (1972)
14. Bianco, L., Mingozzi, A., Ricciardelli, S.: The traveling salesman problem with cumulative costs. Networks **23**(2), 81–91 (1993)

15. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing. STOC '94, pp. 163–171. ACM, New York (1994)
16. Boland, N., Clarke, L., Nemhauser, G.: The asymmetric traveling salesman problem with replenishment arcs. Eur. J. Oper. Res. **123**(2), 408–427 (2000)
17. Brandão, J., Mercer, A.: A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. Eur. J. Oper. Res. **100**(1), 180–191 (1997)
18. Brandão, J., Mercer, A.: The multi-trip vehicle routing problem. J. Oper. Res. Soc. **49**(8), 799–805 (1998)
19. Bryson, K.M.N., Millar, H., Joseph, A., Mobolurin, A.: Using formal MS/OR modeling to support disaster recovery planning. Eur. J. Oper. Res. **141**(3), 679–688 (2002)
20. Campbell, A.M., Vandenbussche, D., Hermann, W.: Routing for relief efforts. Transp. Sci. **42**(2), 127–145 (2008)
21. Caunhye, A.M., Nie, X., Pokharel, S.: Optimization models in emergency logistics: a literature review. Socio Econ. Plan. Sci. **46**(1), 4–13 (2012). Special Issue: Disaster Planning and Logistics: Part 1
22. Chaiken, J.M., Larson, R.C.: Methods for allocating urban emergency units: a survey. Manag. Sci. **19**(4), 110–130 (1972)
23. Chang, S.E., Nojima, N.: Measuring post-disaster transportation system performance: the 1995 Kobe earthquake in comparative perspective. Transp. Res. (Part A) **35**(6), 475–494 (2001)
24. Christofides, N., Mingozzi, A., Toth, P.: The Vehicle Routing Problem. Wiley, Chichester (1979)
25. Davidson, R.A., Zhao, H., Kumar, V.: Quantitative model to forecast changes in hurricane vulnerability of regional building inventory. J. Infrastruct. Syst. **9**(2), 55–64 (2003)
26. De Angelis, V., Mecoli, M., Nikoi, C., Storchi, G.: Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola. Comput. Oper. Res. **34**(6), 1601–1615 (2007)
27. Dell, R.F., Batta, R., Karwan, M.H.: The multiple vehicle TSP with time windows and equity constraints over a multiple day horizon. Transp. Sci. **30**(2), 120–133 (1996)
28. Farahani, R.Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M.: Covering problems in facility location: a review. Comput. Ind. Eng. **62**(1), 368–407 (2012)
29. Fischer III, H.: A Proposed Disaster Scale. Technical. Report, Millersville University of Pennsylvania (2003)
30. Fischetti, M., Laporte, G., Martello, S.: The delivery man problem and cumulative matroids. Oper. Res. **41**(6), 1055–1064 (1993)
31. Fleischmann, B.: The vehicle routing problem with multiple use of vehicles. Working paper, Fachbereich Wirtschaftswissenschafte Universitat Hamburg (1990)
32. Galindo, G., Batta, R.: Review of recent developments in OR/MS research in disaster operations management. Eur. J. Oper. Res. **230**(2), 201–211 (2013)
33. Golden, B., Wasil, E., Kelly, J., Chao, I.: The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets and computational results. In: Crainic, T., Laporte, G. (eds.) Fleet Management and Logistics, pp. 33–56. Kluwer, Dordrecht (1998)
34. Gouveia, L., Voss, S.: A classification of formulations for the (time-dependent) traveling salesman problem. Eur. J. Oper. Res. **83**(1), 69–82 (1995)
35. Green III, W.G., McGinnis, S.R.: Thoughts on the higher order taxonomy of disasters. Notes Sci. Extreme Situat. **7**, 1–6 (2002)
36. Gribkovskaia, I., Gullberg, B.O., Hovden, K.J., Wallace, S.W.: Optimization model for a livestock collection problem. Int. J. Phys. Distrib. Logist. Manag. **36**(2), 136–152 (2006)
37. Haghani, A., Oh, S.C.: Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. Transp. Res. (Part A) **30**(3), 231–250 (1996)
38. Hemel, T., van Erk, S., Jenniskens, P.: The Manhattan Project (1996). http://www.win.tue.nl/whizzkids/1996/tsp.html
39. Hoetmer, G.: Introduction. In: Hoetmer, G.J., Drabek, T.E. (eds.) Emergency Management: Principles and Practice for Local Government, pp. xvii–xxiv. International City Management Association (1991)
40. IFRC: World Disasters Report 2013. Focus on technology and the future of humanitarian action. IFRC (International Federation of Red Cross and Red Crescent Societies) (2013)
41. Jothi, R., Raghavachari, B.: Approximating the k-traveling repairman problem with repairtimes. J. Discret. Algorithms **5**(2), 293–303 (2007)
42. Kara, I., Kara, B.Y., Yetis, M.K.: Cumulative vehicle routing problems. In: Caric, T., Gold, H. (eds.) Vehicle Routing Problem, pp. 85–98. Intech, (2008)

43. Ke, L., Feng, Z.: A two-phase metaheuristic for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. **40**, 633–638 (2013)
44. Kim, S.K., Dshalalow, J.: Stochastic disaster recovery systems with external resources. Math. Comput. Model. **36**(1113), 1235–1257 (2002)
45. Lambert, J.H., Patterson, C.E.: Prioritization of schedule dependencies in hurricane recovery of transportation agency. J. Infrastruct. Syst. **8**(3), 103–111 (2002)
46. Li, L.Y.O., Fu, Z.: The school bus routing problem: a case study. J. Oper. Res. Soc. **53**(5), 552–558 (2002)
47. Lourenço, H., Martin, O., Stützle, T.: Iterated local search: framework and applications. In: Gendreau, M., Potvin, J. (eds.) Handbook of Metaheuristics, pp. 363–397. Springer, Berlin (2010)
48. Lucena, A.: Time-dependent traveling salesman problem: the deliveryman case. Networks **20**, 753–763 (1990)
49. Mak, V., Boland, N.: Heuristic approaches to the asymmetric travelling salesman problem with replenishment arcs. Int. Trans. Oper. Res. **7**(45), 431–447 (2000)
50. Maya, P., Sörensen, K.: A GRASP metaheuristic to improve accessibility after a disaster. OR Spectr. **33**, 525–542 (2011)
51. Mete, H.O., Zabinsky, Z.B.: Stochastic optimization of medical supply location and distribution in disaster management. Int. J. Prod. Econ. **126**(1), 76–84 (2010)
52. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. J. ACM **7**(4), 326–329 (1960)
53. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)
54. Ngueveu, S.U., Prins, C., Calvo, R.W.: An effective memetic algorithm for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. **37**(11), 1877–1885 (2010)
55. Nikolopoulos, C.V., Tzanetis, D.E.: A model for housing allocation of a homeless population due to a natural disaster. Nonlinear Anal. Real World Appl. **4**(4), 561–579 (2003)
56. Olivera, A., Viera, O.: Adaptive memory programming for the vehicle routing problem with multiple trips. Comput. Oper. Res. **34**(1), 28–47 (2007)
57. Özdamar, L.: Planning helicopter logistics in disaster relief. OR Spectr. **33**, 655–672 (2011)
58. Özdamar, L., Ekinci, E., Küçükyazici, B.: Emergency logistics planning in natural disasters. Ann. Oper. Res. **129**, 217–245 (2004)
59. Petch, R.J., Salhi, S.: A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. Discret. Appl. Math. **133**, 69–92 (2004)
60. Pettit, S., Beresford, A.: Emergency relief logistics: an evaluation of military, non-military and composite response models. Int. J. Logist. **8**(4), 313–331 (2005)
61. Picard, J.C., Queyranne, M.: The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. Oper. Res. **26**(1), 86–110 (1978)
62. Prins, C.: Efficient heuristic for the heterogeneous fleet multitrip VRP with application to a large-scale real case. J. Math. Model. Algorithms **1**, 135–150 (2002)
63. Prins, C.: A GRASP× evolutionary local search hybrid for the vehicle routing problem. In: Pereira, F.B., Tavares, J. (eds.) Bio-Inspired Algorithms for the Vehicle Routing Problem. Springer, Berlin (2009)
64. Repoussis, P.P., Tarantilis, C.D., Bräysy, O., Ioannou, G.: A hybrid evolution strategy for the open vehicle routing problem. Comput. Oper. Res. **37**(3), 443–455 (2010)
65. Ribeiro, G.M., Laporte, G.: An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. **39**(3), 728–735 (2012)
66. Salari, M., Toth, P., Tramontani, A.: An ILP improvement procedure for the open vehicle routing problem. Comput. Oper. Res. **37**(12), 2106–2120 (2010)
67. Salehipour, A., Sörensen, K., Goos, P., Bräysy, O.: An efficient GRASP + VND metaheuristic for the traveling repairman problem. Working paper, University of Antwerp, Faculty of Applied Economics (2008)
68. Salhi, S., Petch, R.: A GA-based heuristic for the vehicle routing problem with multiple trips. J. Math. Model. Algorithms **6**, 591–613 (2007)
69. Shim, K.C., Fontane, D.G., Labadie, J.W.: Spatial decision support system for integrated river basin flood control. J. Water Resour. Plann. Manag. **128**(3), 190–201 (2002)
70. Silva, M.M., Subramanian, A., Vidal, T., Ochi, L.S.: A simple and effective metaheuristic for the minimum latency problem. Eur. J. Oper. Res. **221**(3), 513–520 (2012)

71. Swersey, A.J.: The deployment of police, fire, and emergency medical units. In: Pollock, M.R.S.M., Barnett, A. (eds.) Operations Research and the Public Sector, Handbooks in Operations Research and Management Science, vol. 6, pp. 151–200. Elsevier, Amsterdam (1994)
72. Taillard, E.D., Laporte, G., Gendreau, M.: Vehicle routeing with multiple use of vehicles. J. Oper. Res. Soc. **47**(8), 1065–1070 (1996)
73. Tsitsiklis, J.N.: Special cases of traveling salesman and repairman problems with time windows. Networks **22**, 263–282 (1992)
74. Viswanath, K., Peeta, S.: Multicommodity maximal covering network design problem for planning critical routes for earthquake response. Transp. Res. Record **1857**, 1–10 (2003)
75. Wei, Y.M., Xu, W.X., Fan, Y., Tasi, H.T.: Artificial neural network based predictive method for flood disaster. Comput. Ind. Eng. **42**, 383–390 (2002)
76. Yi, W., Özdamar, L.: A dynamic logistics coordination model for evacuation and support in disaster response activities. Eur. J. Oper. Res. **179**(3), 1177–1193 (2007)