



CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Two-Stage Solution Methods for Vehicle Routing in Disaster Area

Monia Rekik
Jacques Renaud
Djamel Berkoune

July 2012

CIRRELT-2012-31

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval,
sous le numéro FSA-2012-008.

Bureaux de Montréal :

Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :

Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Two-Stage Solution Methods for Vehicle Routing in Disaster Area

Monia Rekik^{1,*}, Jacques Renaud¹, Djamel Berkoune²

¹ Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325, de la Terrasse, Université Laval, Québec, Canada G1V 0A6

² Business Unit (BU) Equipements Aéronautiques et Transport Terrestre, Groupe Safran, 10, rue du Fort de Saint Cyr, 78180 Montigny le Bretonneux, France

Abstract. This paper addresses a vehicle routing problem in disaster area. The objective is to distribute humanitarian aid to affected zones from a set of opened humanitarian aid distribution centres using different types of vehicles. Given the particular context of emergency, distribution is planned to satisfy the demand of each affected zone for each type of humanitarian aid in the shortest possible time (this time includes both travelling and vehicles loading and unloading times). An exact and a heuristic solution method are presented, evaluated and compared through a large set of generated instances. We also investigate the impact of split delivery on delivery times.

Keywords. Emergency logistics, vehicle routing, split delivery, mathematical modelling, heuristic.

Acknowledgements. This work was financed by individual grants and by Collaborative Research and Development grants [CG 095123] from the Natural Sciences and Engineering Research Council of Canada (NSERC). This financial support is gratefully acknowledged. This research was carried out while Djamel Berkoune did his postdoctoral fellowship at the CIRRELT.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: Monia.Rekik@cirrelt.ca

1. Introduction

Emergency situations, whether caused by natural disasters (earthquake, flooding, tsunamis) or humans (chemical spills, wars, terrorist attacks), require large logistic deployments to assist victims. Several problems need to be solved at different levels of the supply chain (Sheu, 2007b). Emergency Management is generally divided into four main phases: mitigation, preparedness, response and recovery (Altay and Green III, 2006; Haddow et al., 2008)

The mitigation and preparedness phases are pre-crisis. They seek to define measures to reduce, mitigate or prevent the impacts of disasters and to develop action plans that will be implemented upon the occurrence of a disaster. When the crisis occurs, the phases of response and recovery are initiated. The response, or intervention, phase consists in mobilizing and deploying emergency services within the disaster area in order to protect people and reduce human and material damages. The recovery phase defines the measures leading to the return to normal, that is, to a standard of living of the same quality as it was before the disaster occurred.

In this paper, we focus on one of the most important problems of the response phase: the distribution of humanitarian aid within the disaster area. The objective is to define routes to deliver products and / or essential services to victims (clients) in short times. These products and services are assumed available in specialized locations deployed by crisis managers within the disaster area to serve as temporary depots, usually called, humanitarian aid distribution centres (HADC). Each of these HADC has a limited fleet of vehicles of different types, each type being characterized by a number of parameters such as loading capacity (in volume and weight), loading and unloading times, total work time, etc.

Given the particular context of emergency, our goal is to propose a deployment that responds to victims demand while minimizing the maximum delivery time. This delivery time takes into account both travelling time and

loading and unloading times of the vehicles used. Split delivery is assumed admissible enabling thus a client (a demand point) to be visited by more than one vehicle trip. This assumption becomes important in emergency situations since the primary objective is to assist victims with the product/service needed in the shortest possible time. Consequently, the distribution plan is expected to use the maximum of vehicles available at HADCs.

We will see in Section 3 that the distribution problem addressed in this paper reduces to a complex multi-commodity multi-depot vehicle routing problem with heterogeneous fleet and split delivery. The main contribution of this paper is to propose, evaluate and compare two solutions methods, one exact and one heuristic, for solving this problem. Both methods use a two-stage solution approach. The first stage generates a set of *empty* routes, i.e., routes in which only the sequence of clients visited from each HADC are determined. The second stage selects a subset of routes among those generated at the first stage and determines the quantity of products delivered to each client on each route. This is done by solving a mixed integer programming model with a classical branch-and-bound process.

Our experimental study shows that the exact method, although ensuring an optimal solution, may require large computing times. On the opposite, the heuristic method shows good performances in terms of computing times and results in relatively good solutions in general. We also investigate the impact of split delivery on delivery times. The results obtained prove that enabling an affected zone (a client) to be visited more than once considerably reduces the maximum access time.

The paper is organized as follows. Section 2 is a brief literature review on distribution problems in emergency contexts. Section 3 formally describes the problem addressed in this paper and introduces the terminology and notation used throughout. In section 4, we present the two-stage solution approach and describe in details the exact and the heuristic methods. Our experimental study is presented in Section 5. Section 6 summarizes the most

important contributions of the paper and presents our future research.

2. Literature review

There is a vast and rich literature on routing problems. The traveling salesman problem (TSP), which is central to almost all routing problems, has received considerable attention in the last decades (Laporte, 1992a, 2010; Applegate et al., 2006; Gutin and Punnen, 2007). Its generalization, the vehicle routing problem (VRP), has also been extensively studied (Golden and Assad, 1988; Laporte, 1992b; Cordeau et al., 2002; Laporte, 2009). The VRP was used as a testbed for the development of metaheuristics, and we can consider that the algorithms currently used are not only sophisticated, but also provide solutions that are very close to optimal (Laporte et al., 2000; Potvin, 2009).

In recent years, research has shifted to different and more complex VRP problems such as dynamic problems (Wen et al., 2010), delivery problems with time windows (Cornillier et al., 2009) and fleet composition (Pessoa et al., 2007). Routing problems in emergency situations are one of these new realistic and challenging problems (Sheu, 2007b). Altay and Green III (2006) report that most research has been done in the phases of prevention and preparedness of emergency management. According to their analysis, the response phase remains rarely addressed especially in terms of delivery planning.

Sheu (2007a, 2010) developed a decision support system for emergency management that includes three modules: a module for aggregating the affected areas into zones, a module for setting priorities on relief distribution for the aggregated zone and a module for distribution planning. Yi and Özdamar (2007) propose an integrated model for locating logistics support and evacuating victims to emergency units. An important part of their model deals with the distribution of medical entities between emergency units for emergency care of affected people. Yi and A. (2007) used the model of Yi

and Özdamar (2007) for relief distribution operations without considering the location component.

Tzeng et al. (2007) consider a three-objective model of relief distribution. The first objective minimizes the total cost, the second objective minimizes travel time and the third objective maximizes demand satisfaction of affected areas. The authors handle the dynamic data by considering a multi-period model in which most of parameters and variables are time-related. The goal of the model is to determine the transfer (i.e., distribution) centers to be opened and the quantities of products to be transported from collection points to transfer points and from transfer points to the final demand points. A fuzzy multi-objective programming method is used to solve the problem. The network considered in the experimental study includes four collection points, four transfer points and eight demand points.

Balcik et al. (2008) consider the problem of humanitarian distribution in a network including a single depot and different vehicle types. The objective is to allocate the relief supplies available at the depot to the demand locations and determine the delivery schedules/routes for each vehicle throughout a multiple-day planning horizon. The authors propose a two-phase solution approach in which the first phase generates all possible delivery routes for each vehicle and the second phase determines the periods to visit each demand location, the delivery amounts, and the vehicle loads. The objective is to minimize the sum of transportation costs and penalty costs of unsatisfied and late-satisfied demand for different types of relief supplies. The main goal of the experimental study is to discuss the trade-off between resource allocation and routing decisions on a number of test problems.

Jotshi et al. (2009) developed a methodology for routing emergency vehicles in order to create better connections between the points where the victims are and hospitals. Recently, Berkoune et al. (2012) propose two heuristic methods: a greedy heuristic and a genetic algorithm to solve the humanitarian aid distribution in a context similar to the one considered in

this paper. The authors consider the problem where total distribution time is minimized with only back-and-forth routes.

The solution approach considered in this paper uses a two-stage procedure that presents a lot of similarities with that of Balcik et al. (2008) but remains different on some points. First, Balcik et al. (2008) consider only what we refer to as the exact method in the sense that all admissible routes are enumerated in the first phase. Second, the emergency context and the objectives of the distribution problem we address are different from those of Balcik et al. (2008) (multiple depots, maximum access time, etc.) making thus the problem much difficult to solve with such exact methods. Third, our objective in the second phase is to select routes and vehicle loads so as to minimize the maximum delivery time while satisfying the demand of each demand point. Such an objective is of a primary importance in emergency contexts since the time at which first relief is delivered has a significant impact on victims' life and health. Finally, and most importantly, one of our primary concerns was to propose, evaluate and compare the performance of the proposed methods for large problems. In Balcik et al. (2008), the emphasis is put on the impact of different input parameters (such as penalty costs of unsatisfied demands) on the solution structure. The largest instance considered include one depot, four demand points, two products and two vehicles. In our case, we consider instances including up to five depots and 25 demand points, prove the limit of the exact method for these instances and show how the proposed heuristic is efficient in this case.

Next section describes in more details the distribution problem we consider.

3. Problem description

In disaster situations, the humanitarian aid may consist of tangible products (e.g., food, health products, medicines, water, beds) or services (e.g., securing a bridge, restoring a power line). These products and services are

assumed available (in limited quantities) at what we called Humanitarian Aid Distribution Centres, HADC. The problem we address assumes that the number, the location and the supply of HADC for different products and services have been set in advance and will not change over the planning horizon (The planning horizon may extend over several hours or a full day). In the following, the set of HADC deployed within the affected area is denoted L .

When a disaster occurs, each particular house or building within the affected region could require relief or humanitarian aid, thus becoming a potential demand point. Generally, these demand points are geographically grouped into demand zones according to certain rules. This avoids handling a huge amount of information and resources for severe crisis affecting large areas. In the following, we employ the term “client” to designate an aggregated demand zone. The set of all clients within the affected region is denoted I . The nature and the quantity of demand vary from a client to another depending on the severity and the type of damage. They are however assumed fixed for a given client during the planning horizon. In the following, the set of required products is denoted J and the demand of each client i for each product j is denoted d_{ij} .

Humanitarian aid is deployed from the various HADC to clients using vehicles (in the case of products) or service providers (in the case of services). Given the diversity of product/services required in a crisis situation, the fleet of vehicles available at HADCs can be highly heterogeneous (light vehicles, trailers, semi-trailers, tanks). In crisis, since the emphasis is usually placed on the speed at which aid is distributed to victims, a common goal consists in minimizing the duration of distribution operations. Given the heterogeneity of the fleet of vehicles used, these durations should include not only travelling times but also vehicle loading and unloading times at HADCs and at delivery points. These times depend on the nature of the product handled, the type of vehicle used, more precisely the specific equipment of vehicles (cranes, door openings, refrigeration system), and the infrastructure available at both

HADC and delivery points. In the following, we denote by K_l the set of vehicles available at HADC l . Each vehicle $k \in K_l$ is characterized by: (1) the time, $\rho_{k,j,l,i}$, required for loading one unit (example a pallet) of product j from HADC l , and unloading it to client i ; (2) a capacity in terms of weight and volume, denoted respectively W_k , and V_k ; and (3) a maximum daily work time, denoted D_k , implying that the total vehicle trip duration cannot exceed D_k unit times. We assume that a vehicle departing from a HADC must necessarily end its trip at the same HADC. Furthermore, we denote by v_j the volume occupied by a unit of product j and by w_j its weight.

Hence, the Vehicle Routing Problem in Disaster Area (VRP-DA) addressed in this paper is defined on a graph $G = (L \cup I, A)$ where supply nodes, or depots, are the HADC, L , the demand nodes are the aggregated demand zones (the clients) within the disaster area, I , and A is the set of arcs. An arc exists between two nodes in this graph if a road connecting these nodes is still available after the disaster. To each arc $a \in A$ is associated a travel time t_a that takes into account the state of the road linking the origin and destination nodes of a .

The objective of the VRP-DA is to define delivery routes that satisfy the demand of all clients for all products, taking into account the characteristics of the vehicles available at each HADC, as well as the supply in products of each of these HADC. Unlike classical vehicle routing problems where the goal is usually to minimize the total travelling time, we consider here an objective specific to emergency contexts which aims to deliver aid to all demand points in the shortest possible time. Therefore, the VRP-DA aims at minimizing the maximum delivery time; this time considering both travel and loading and unloading times of vehicles. Indeed, in crisis situations, the primary objective is to supply victims with necessary relief in a relatively short time especially in the first hours following the disaster (at that time, one ignores the extent of damages and people's lives could be in danger). A maximum access time, denoted τ in the following, is generally pre-specified to ensure

that each client can be reached by at least one HADC within this maximum access time. We assume that HADCs are chosen in a way ensuring that all clients are reachable from at least one HADC within τ .

Finally, given the particular context of emergency, and unlike classical VRPs where the fixed costs of vehicles are taken into account in the distribution process, our objective here is to exploit as much vehicles as possible to reach our objective of minimizing the maximum delivery time. Hence, split delivery is tolerated to some extents, enabling thus a client to be visited by more than one route. Indeed, in practice, decision makers may want to limit the number of visits to a client to simplify delivery management. Furthermore, our experimental study shows that, for the instances considered, no significant gain is achieved beyond a given threshold of permitted splits.

Figure 1 hereafter illustrates an example of the VRP-DA addressed in this paper. In this example, 11 demand points (represented by circles) are to be delivered from three HADC (represented by squares) with vehicles of different types available at each HADC. It is assumed that the road network defines a complete graph implying that all nodes (demand points and HADC) are connected by arcs (not shown on the graph to alleviate the presentation). The dotted circles surrounding each HADC represent the maximum access time prefixed by the crisis managers. For example, only clients 3, 4, 6 and 7 are reachable from HADC 2 in a time less than or equal to the maximum access time (by a direct forth and back trip). Therefore, client 6, for example, will not be delivered from HADC 2. However, client 3 is within the coverage region of both HADC 2 and 3 and could thus be delivered from one or both of them. Since split delivery is permitted, the demand of client 3 for a given product may be satisfied by two different trips, using different vehicles and possibly departing from different HADC (HADC 2 and/or 3).

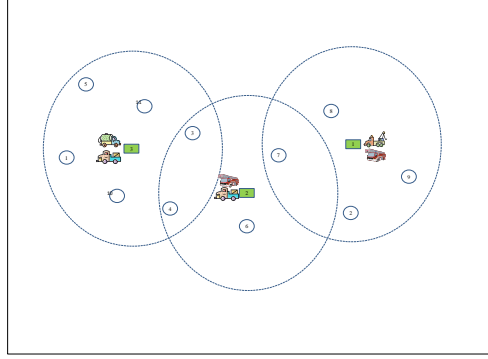


Figure 1: Illustrative example of VRP-DA

4. Solution Methods

The objective of the VRP-DA is to produce feasible routes from HADC to various delivery points to meet their demand for products and services in the shortest possible time. A route is feasible if it respects the time and capacity restrictions of the vehicle to which it is assigned. This depends on the type of vehicle used, the nature and the quantity of products delivered (time of loading and unloading), and the locations of the origin point (the HADC) and delivery points (clients) constituting the route.

Obviously, enumerating all feasible routes (with their loads) requires considerable time and would result in an exorbitant number of routes that cannot be efficiently handled by classical partitioning models. Notice that this number becomes larger in the context of split delivery treated here. Thereby, we consider a two-stage solution approach to address VRP-DA.

Two main concepts need to be introduced before describing the approach used: *empty routes* and *loaded routes*. An empty route r is defined by a three-tuple (l, k, S) , where l is the HADC from which it departs (and to which it returns), k is the vehicle to which the route is assigned (k is one of the vehicles available at HADC l , i.e., $k \in K_l$), and S is the sequence of clients visited. A loaded route is defined by a pair (r, Q) where $r = (l, k, S)$

is an empty route for which we specify, when selected, the quantities Q_{ij} of products j delivered to each client i in S .

The two-stage solution approach first determines a set of candidate empty routes with regard to some criteria. This set is then considered in the second stage to determine the empty routes to be selected and their loading. The second stage uses a Mixed Integer Programming model that is solved with the commercial solver CPLEX.

Two solution methods are presented: an exact method and a heuristic. These methods differ in the definition of candidate empty routes generated at the first stage. While the exact method considers all feasible empty routes that are able to generate an optimal solution, the heuristic considers a subset of empty routes. Next section describes the first stage for both the exact and the heuristic methods. Section 4.2 defines the MIP model of stage 2.

4.1. Stage 1: Generation of candidate empty routes

Recall that an empty route r is defined by a three-tuple (l, k, S) , where l is the HADC from which it departs (and to which it returns), k is the vehicle to which the route is assigned (k is one of the vehicles available at HADC l , i.e., $k \in K_l$), and S is the sequence of clients visited by the route. At this step, an empty route is feasible if it respects the time restrictions of the vehicle to which it is assigned as well as the maximum access time τ . Only travelling times are taken into account to determine routes admissibility (vehicles are empty). Two methods are considered for generating the set of feasible empty routes. The first method enumerates all possible empty routes that are able to yield an optimal solution. The second method considers only a subset of empty routes generated based on the sweep method.

4.1.1. Exhaustive enumeration

Algorithm 1 describes the procedure used for enumerating all promising empty routes. It considers the following notation:

- R^e : the set of empty routes generated by the exhaustive enumeration algorithm.
- I_l : the set of clients that are admissible for HADC l . A client i is admissible for HADC l if it can be reached from HADC l in a time less than or equal to the maximum access time τ with a direct trip, i.e. $I_l = \{i \in I : t_{li} \leq \tau\}$.
- I_l^u : a non-empty subset of I_l .
- $S_{l,k}^u$: the optimal feasible sequence for visiting clients in I_l^u (starting and ending at l) that minimizes the maximum access time of the last client visited in I_l^u (taking into account only travelling times). A sequence $S_{l,k}^u$ is feasible if: (1) it respects the maximum access time restriction, and (2) it respects the total vehicle trip duration D_k of vehicle k . To determine this sequence, we solve a particular TSP restricted to depot l , vehicle k and clients in I_l^u . In the following, this TSP is denoted $TSP(l, k, I_l^u)$.

The exhaustive enumeration algorithm is described as follows:

Algorithm 1 Exhaustive enumeration algorithm

0. $R^e = \emptyset$
 1. For each HADC $l \in L$
 2. Determine set I_l .
 3. Determine all subsets I_l^u of I_l .
 4. For each subset I_l^u
 5. For each vehicle $k \in K_l$
 6. Solve $TSP(l, k, I_l^u)$.
 7. If $TSP(l, k, I_l^u)$ is feasible then:
 8. Let $S_{l,k}^u$ its optimal solution.
 9. Add route $r = (l, k, S_{l,k}^u)$ to R^e .
 10. End If
 11. End For
 12. End For
 13. End For
-

Recall that problem $TSP(l, k, I_l^u)$ is different from the classical travelling salesman problem on three points. First, the objective function minimizes the maximum delivery time rather than the total travelling duration. Second, it considers additional constraints to model the maximum access time restriction (τ). Third, it considers additional constraints for the maximum tour duration (D_k) permitted for vehicles.

Depending on the size of subsets I_l^u generated in step 3 of Algorithm 1, problem $TSP(l, k, I_l^u)$ may be more or less easy to solve to optimality. In emergency contexts however, given the restrictions imposed by the maximum access time and the objective of minimizing the maximum delivery time, the number of clients visited on a tour is generally limited to a maximum pre-specified value (3 or 4 in our experimental study) yielding small subsets I_l^u . In this case, $TSP(l, k, I_l^u)$ is solved to optimality by a simple enumeration of all admissible sequences.

4.1.2. Partial enumeration

As explained above, the exhaustive enumeration algorithm enumerates all the $(2^{|I_l|} - 1)$ possible subsets of set I_l . Obviously, when set I_l is too large, such an approach would yield intractable problems. We propose in the following, a method that enumerates a restricted number of subsets of I_l . This method is described by Algorithm 2 and is inspired by the classical sweep algorithm. It considers, in addition to the notation introduced in Algorithm 1, the following sets:

- R^p = the set of empty routes generated by the partial enumeration algorithm.
- I_l^{ord} = the set I_l ordered in ascending order with respect to clients polar coordinates.
- $I_l^{(u, ord)}$ = subset of I_l including m , ($m = 1, \dots, |I_l|$) consecutive elements of I_l^{ord} .

Algorithm 2 Partial enumeration algorithm

0. $R^p = \emptyset$
 1. For each HADC $l \in L$
 2. Determine set I_l and the corresponding order set I_l^{ord} .
 3. Determine all subsets $I_l^{(u,ord)}$ of I_l^{ord} .
 4. For each subset $I_l^{(u,ord)}$
 5. For each vehicle $k \in K_l$
 6. Solve $TSP(l, k, I_l^{(u,ord)})$.
 7. If $TSP(l, k, I_l^{(u,ord)})$ is feasible then:
 8. Let $S_{l,k}^u$ its optimal solution.
 9. Add route $r = (l, k, S_{l,k}^u)$ to R^p .
 10. End If
 11. End For
 12. End For
 13. End For
-

4.2. Stage 2: Definition and selection of loaded routes

The objective of Stage 2 is to select optimal empty routes, among a pre-specified set of candidate empty routes R (deriving either from the exhaustive or the partial enumeration), and determine their loadings (quantities of products $j \in J$ to deliver) so as to minimize the maximum delivery time. This is done by solving a particular MIP as described in the following.

4.2.1. Model parameters and decision variables

Let R denote the set of candidate empty routes considered by the MIP of stage 2. This model is referred to in the following as $(P(R))$. For each $r \in R$, we define three sets of constant parameters $\delta_{rl}, l \in L$, $\alpha_{rk}, k \in K$, and $\beta_{ri}, i \in I$ as follows:

- $\delta_{rl} = 1$, if route r is associated to HADC l ; 0, otherwise.
- $\alpha_{rk} = 1$, if route r is associated to vehicle k ; 0, otherwise.
- $\beta_{ri} = 1$, if route r visits client i ; 0, otherwise.

We also define for each route $r = (l, k, S) \in R$:

- T_r^d = the travelling time (excluding loading and unloading times) required to visit all the clients in S and return to depot l .

- \tilde{T}_r^d = the travelling time (excluding loading and unloading times) required to visit the last client in S (this time is different from T_r^d since it does not consider the travelling time required to return to depot l).

Model (P) uses the following decision variables:

- $x_r = 1$, if empty route $r \in R$ is selected; 0, otherwise. This variable is defined for each $r \in R$.
- y_{rji} = the quantity of product j delivered to client i by route r . This variable is defined for each $r = (l, k, S) \in R$, each $j \in J$ and each $i \in S$.
- L_{Max} = the maximum delivery time.

4.2.2. Mathematical model

Model $(P(R))$ is formulated as follows:

$$(P) : \min L_{max} \quad (1)$$

$$\sum_{r \in R} \beta_{ri} y_{rji} = d_{ij} \quad \forall i \in I, j \in J \quad (2)$$

$$\sum_{r \in R} \sum_{i \in I} \delta_{rl} \beta_{ri} y_{rji} \leq p_{jl} \quad \forall j \in J, l \in L \quad (3)$$

$$\sum_{r \in R} \sum_{i \in I} \sum_{j \in J} \beta_{ri} \alpha_{rk} w_j y_{rji} \leq W_k \quad \forall k \in K \quad (4)$$

$$\sum_{r \in R} \sum_{i \in I} \sum_{j \in J} \beta_{ri} \alpha_{rk} v_j y_{rji} \leq V_k \quad \forall k \in K \quad (5)$$

$$\sum_{j \in J} \sum_{i \in S} \rho_{kjl} y_{rji} + T_r^d \leq D_k x_r \quad \forall r = (l, k, S) \in R \quad (6)$$

$$\sum_{j \in J} \sum_{i \in S} \rho_{kjl} y_{rji} + \tilde{T}_r^d \leq L_{max} \quad \forall r = (l, k, S) \in R \quad (7)$$

$$\sum_{r \in R} \alpha_{rk} x_r \leq 1 \quad \forall k \in K \quad (8)$$

$$L_{max} \geq 0; x_r \in \{0, 1\}, y_{rji} \geq 0 \quad \forall r \in R, j \in J, i \in I. \quad (9)$$

Objective function (1) minimizes the maximum delivery time. Constraints (2) ensure that the demand of each client for each product is satisfied.

Constraints (3) guarantee that the total quantity of product j delivered from a HADC l does not exceed the amount of j available at l . Constraints (4), respectively (5), are the capacity constraints in terms of weight, respectively, volume, associated with each vehicle used. Constraints (6) ensure that the total duration of a route $r = (l, k, S)$ is less than or equal to the maximum duration permitted for the vehicle k performing this route. Constraints (6) also bind variables y_{rji} to take null values when route r is not selected (i.e., $x_r = 0$). Constraints (7) set the upper bound on the maximum delivery time to L_{max} . Constraints (8) ensure that each vehicle is assigned at most one route. Constraints (9) define the nature of the decision variables of the model.

As formulated, model $(P(R))$ enables a client to be delivered by multiple routes. However, in emergency situations, one may want to limit the number of times a client is delivered, in order, for example, to alleviate delivery operations management and/or to reduce the number of access to certain areas highly affected or risky. When such a restriction is considered, one has just to add the following set of constraints to model $(P(R))$:

$$\sum_{r \in R} \beta_{ri} x_r \leq N_i, \quad \forall i \in I, \quad (10)$$

where N_i is the maximum number of visits allowed for client i .

It is noteworthy that, when set R is generated by Algorithm 1 where TSPs are solved to optimality, solving model $(P(R))$ to optimality produces an optimal solution to VRP-DA. This combination corresponds indeed to what we refer to as the exact method. However, when set R is output by Algorithm 2, the optimal solution of model $(P(R))$ is not necessarily optimal for VRP-DA. This corresponds to the heuristic method.

5. Computational experiments

The objective of this section is twofold. First, we want to evaluate the computational performance of the two proposed solution methods in terms of solution time and solution quality. To this end, we randomly generate a set of instances with various size. Second, we want to assess the impact of split delivery on solution quality. To this end, we fix the value of the maximum number of visits allowed for clients to $\mathcal{N} = 1$ (implying that no split is permitted), $\mathcal{N} = 2$, $\mathcal{N} = 3$ and $\mathcal{N} = 4$ and compare the corresponding values of the maximum delivery time, L_{max} (output by the exact method).

Algorithms 1 and 2 are coded in Visual Basic and the procedure of branch-and-Bound CPLEX 12.0 (with its default settings) was used to solve MIP models in stage 2 on a PC Intel Core 2 Duo, 3.00 GHz and 4.00 GB RAM. A one-hour time limit was set for the resolution of MIP models with CPLEX for both methods.

5.1. Problem tests

A series of problems are randomly generated by varying the number of HADC ($|L|$) and the number of delivery points ($|I|$). For all generated instances, the vehicles available at each HADC are of two types with two vehicles of each type and consider two types of products (i.e., $|J| = 2$). 10 problem settings in total are generated. These problems are divided in two sets of 5 problems each. For the first set, the total number of all admissible empty routes (as generated by the exact method) is about 2120 on average. For the second set, this average number is about 6014. Each problem is solved under 8 contexts. These contexts are obtained by varying the split parameter ($\mathcal{N} = 1, 2, 3, 4$) and the maximum number of clients visited on a vehicle route ($Max_{Cli} = 3, 4$) for a total of 8 instances per problem setting. Notice that given these limits on the number of clients forming a route, TSP problems of stage 1 for both the heuristic and the exact methods are solved to optimality by enumerating and comparing all possible permutations.

5.2. Computational performance of the exact and the heuristic methods

Tables 1, 2, 3 and 4 hereafter display the results obtained with both the exact and the heuristic methods for the instances set 1 and 2, respectively. In the following, we will first compare the performance of the two solution methods in terms of solution quality (Tables 1 and 2). Solution times required by both methods are then compared in a second step (Tables 3 and 4).

5.2.1. Comparison of solution quality

Tables 1 and 2 report, for each instance and each solution method, the number of routes generated at the first stage ($|R|$), the objective value resulting from solving the second-stage model with CPLEX within a time limit of 3600s (Solution), and the corresponding integrality gap (computed by CPLEX as the distance in percentage between the best upper and lower bounds identified throughout the branch-and-bound process). To assess the quality of solutions obtained by the heuristic approach, we also report in the final column the distance, in percentage, between the solution output by the heuristic (B) and the exact method (A) (i.e., $(B - A)/A$).

The results of Tables 1 and 2 show that the exact method identifies an optimal solution for all instances where split is not permitted (value 1 under the column “Split”). Tables 1 and 2 also show that the exact approach solves the problem to optimality for 45 out of the 80 instances (56.25%) considered (within the time limit of one hour). For the remaining 35 instances, the gap as provided by CPLEX varies between 1.9% (instance 22) and 8.83% (instance 40) with an average of 5.01% for instances set 1 and between 5.74% (instance 51) and 54.8% (instance 79) for instances set 2 with an average of 21.82%.

The heuristic approach produces the same solutions as the exact method (within the time limit of one hour) for 35 instances over the 80 considered in this study (43.75%). For the remaining 45 instances, the exact method identified better solutions for 38 instances. For these instances, the difference (in percentage) between the solutions output by the heuristic and those produced by the exact method is on average equal to 2.45% ranging from

Pb	Inst.	Nb_HADC	Nb_Client	Max_Cli	Split	Exact method			Heuristic method			(B-A)/A
						R	Solution (A)	Gap	R	Solution (B)	Gap	
1	1	3	15	3	1	1640	62,15	0,00%	400	62,15	0,00%	0,00%
					2		59,01	3,77%		60,45	0,00%	2,45%
					3		59,01	4,58%		60,45	0,00%	2,45%
					4		59,01	5,51%		60,45	0,00%	2,45%
				4	1	2048	62,15	0,00%	436	62,15	0,00%	0,00%
					2		59,01	2,80%		60,45	0,00%	2,45%
					3		59,01	2,80%		60,45	0,00%	2,45%
					4		59,01	8,27%		60,45	0,00%	2,45%
2	2	3	15	3	1	2596	72,67	0,00%	444	72,67	0,00%	0,00%
					2		67,87	0,00%		67,87	0,00%	0,00%
					3		66,27	0,00%		67,87	0,00%	2,41%
					4		65,87	0,00%		67,87	0,00%	3,04%
				4	1	3992	72,67	0,00%	520	72,67	0,00%	0,00%
					2		67,87	0,00%		67,87	0,00%	0,00%
					3		66,27	0,00%		67,87	0,00%	2,41%
					4		65,87	0,00%		67,87	0,00%	3,04%
3	3	3	15	3	1	1320	48,50	0,00%	400	52,00	0,00%	7,21%
					2		46,32	3,58%		47,45	0,00%	2,43%
					3		46,29	7,46%		47,45	0,00%	2,50%
					4		46,29	5,47%		47,45	0,00%	2,50%
				4	1	1612	48,50	0,00%	444	52,00	0,00%	7,21%
					2		46,32	1,90%		47,45	0,00%	2,43%
					3		46,29	6,37%		47,45	0,00%	2,50%
					4		46,29	6,36%		47,45	0,00%	2,50%
4	4	4	15	3	1	2104	62,15	0,00%	524	62,15	0,00%	0,00%
					2		57,35	0,00%		57,35	0,00%	0,00%
					3		55,75	0,00%		55,75	0,00%	0,00%
					4		54,95	0,00%		55,08	0,00%	0,23%
				4	1	2620	62,15	0,00%	568	62,15	0,00%	0,00%
					2		57,35	0,00%		57,35	0,00%	0,00%
					3		55,75	0,00%		55,75	0,00%	0,00%
					4		54,95	0,00%		55,08	0,00%	0,23%
5	5	4	15	3	1	1484	47,26	0,00%	484	47,26	0,00%	0,00%
					2		43,03	2,40%		43,03	0,00%	0,00%
					3		41,95	5,48%		41,95	0,00%	0,00%
					4		41,27	5,71%		41,95	0,00%	1,65%
				4	1	1784	47,26	0,00%	528	47,26	0,00%	0,00%
					2		41,27	2,63%		41,27	0,00%	0,00%
					3		41,27	6,26%		41,27	0,00%	0,00%
					4		41,27	8,83%		41,27	0,00%	0,00%

Table 1: Solution quality of the exact and the heuristic methods for instances set 1

0.23% (instance 28) to 7.21% (instances 17 and 21). The heuristic outperforms the exact method 7 times (instances 62, 74, 75, 76, 78, 79, 80) with an average deviation of 6% reaching 12.27%. Six of these instances correspond to problem setting 10 which considers 5 HADC and 25 clients. Notice that problem setting 10 yields a number of admissible empty routes exceeding 14024 for the exact method compared to only 1272 for the heuristic.

It is also clear from Tables 1 and 2 that the MIP of the second stage is much easier to solve for the heuristic than for the exact method. Indeed, the MIP resulting from the partial enumeration (the heuristic approach) was solved to optimality for all the instances of set 1 and for 33 instances of set 2 (that is 91, 25% of the instances in total). Notice that this information is straightforwardly deduced from the "Gap" column of the heuristic approach. The average gap for the 7 MIP non-solved to optimality is 7.98% with a value

Pb	Inst.	Nb_HADC	Nb_Client	Max_Cli	Split	Exact method			Heuristic method			(B-A)/A
						R	Solution (A)	Gap	R	Solution (B)	Gap	
6	4847464544434241	4	15	3	1	2428	61,34	0,00%	540	61,34	0,00%	0,00%
					2		56,54	0,00%		56,54	0,00%	0,00%
					3		54,94	0,00%		55,41	0,00%	0,86%
					4		54,94	0,00%		55,41	0,00%	0,86%
				4	1	3352	61,34	0,00%	588	61,34	0,00%	0,00%
					2		56,54	0,00%		56,54	0,00%	0,00%
					3		54,94	0,00%		55,41	0,00%	0,86%
					4		54,14	0,00%		55,41	0,00%	2,35%
7	56555453525150494847464544434241	4	20	3	1	4824	62,15	0,00%	760	62,34	0,00%	0,30%
					2		57,35	10,37%		58,83	3,08%	2,57%
					3		55,75	5,74%		58,83	7,54%	5,52%
					4		55,75	9,87%		58,83	7,10%	5,52%
				4	1	7424	62,15	0,00%	868	62,34	0,00%	0,30%
					2		55,35	8,37%		58,83	3,95%	6,28%
					3		55,75	9,34%		58,83	10,51%	5,52%
					4		57,35	10,32%		58,83	2,68%	2,58%
8	646362616059585756555453525150494847464544434241	4	20	3	1	5012	61,34	0,00%	708	61,34	0,00%	0,00%
					2		56,54	0,00%		57,23	0,00%	1,23%
					3		56,54	9,91%		57,23	0,00%	1,23%
					4		56,54	8,18%		57,23	0,00%	1,23%
				4	1	8268	61,34	0,00%	808	61,34	0,00%	0,00%
					2		59,06	11,18%		57,23	0,00%	-3,09%
					3		56,54	10,80%		57,23	0,00%	1,23%
					4		56,54	11,36%		57,23	0,00%	1,23%
9	80797877767574737271706968676665646362616059585756555453525150494847464544434241	5	20	3	1	3392	51,40	0,00%	668	51,40	0,00%	0,00%
					2		48,75	0,00%		48,75	0,00%	0,00%
					3		48,75	0,00%		48,75	0,00%	0,00%
					4		48,75	0,00%		48,75	0,00%	0,00%
				4	1	4628	51,40	0,00%	752	51,40	0,00%	0,00%
					2		48,75	0,00%		48,75	0,00%	0,00%
					3		48,75	0,00%		48,75	0,00%	0,00%
					4		48,75	0,00%		48,75	0,00%	0,00%
10	80797877767574737271706968676665646362616059585756555453525150494847464544434241	5	25	3	1	8560	55,57	0,00%	1064	55,57	0,00%	0,00%
					2		55,53	47,4%		52,55	0,00%	-5,37%
					3		55,43	25,4%		52,33	0,00%	-5,60%
					4		53,21	47,7%		52,33	21,00%	-1,66%
				4	1	14024	55,57	0,00%	1272	55,57	0,00%	0,00%
					2		57,62	45,7%		52,55	0,00%	-8,80%
					3		59,65	54,8%		52,33	0,00%	-12,27%
					4		55,11	44,5%		52,22	0,00%	-5,24%

Table 2: Solution quality of the exact and the heuristic methods for instances set 2

reaching 21% for instance 76. One should notice that for these instances, the MIP corresponding to the exact method is also very difficult to solve to optimality (the gap exceeds 54% for instance 79).

It is noteworthy that although the heuristic approach results in a MIP that is relatively easy to solve for almost all the instances, the deviation from the optimal solution of the VRP-DA can be relatively large. This is due to the fact that this approach does not consider the set of all feasible empty routes as is the case for the exact approach. For example, the MIPs of both methods are solved to optimality for instance 17. However, the difference between the exact and the heuristic solutions reaches 7.21% implying thus that the solution produced by the heuristic in this case is 7.21% sub-optimal. In fact, for the 45 solved to optimality with the exact method within the time limit of one hour, the heuristic method was able to identify the optimal

solution for 30 instances. The solutions output by the heuristic for the 15 remaining instances deviate 2.17% on average from the optimal solution.

The results of Tables 1 and 2 also shows that, for the majority of instances, the difficulty of solving the MIP of stage 2 increases with the number of visits allowed for a client. This was pretty expected since enabling a client to be visited 4 times offers more alternative deployments than restricting the number of visits to only 1.

5.2.2. Comparison of solution time

Tables 3 and 4 compare the two approaches in terms of the solution time required. They display for each method: (1) the time (in seconds) needed to generate admissible empty routes at stage 1 (under the column "Stage 1"); (2) the time (in seconds) for solving the MIP of the second stage (column "Stage 2"); and (3) the total solution time (under the column "1 + 2"). Notice that a dash '-' in columns "Stage 2" indicates that no optimal solution for the second-stage model was identified within the time limit of one hour.

The results of Tables 3 and 4 show that the time required for generating admissible routes for the heuristic method (i.e., the time required by Algorithm 2) is relatively small for all the instances. This time is on average equal to 7.38 seconds for problems of set 1 and 11.94 seconds for problems in set 2 and does not exceed 24.51 seconds for all the instances.

However, the exact method requires much more time at the first stage; this time exceeding 103939 seconds for problem setting 10 in which 14024 routes are enumerated. When considering all the problem settings, the average time required by Algorithm 1 to generate admissible empty routes equals 1473.11 seconds. One can notice also that this time considerably increases when the maximum number of clients visited on a route goes from 3 to 4.

When considering stage 2, the MIPs are solved to optimality for 45 instances over the 80 for the exact method requiring an average solution time of 252.08 seconds ranging from 2.77 seconds for instance 25 to 2165.36 seconds for instance 58. On another hand, the MIPs considered in the heuristic

Pb	Inst.	Nb_HADC	Nb_Client	Max_Cli	Split	Exact Method			Heuristic method		
						Stage 1	Stage 2	1+2	Stage 1	Stage 2	1+2
1	1 2 3 4 5 6 7 8	3	15	3	1	68,41	31,46	99,87	3,46	6,88	10,34
					2		-	-		613,85	617,31
					3		-	-		415,21	418,67
					4		-	-		572,46	575,92
				4	1	900,94	109,33	1010,27	10,14	3,78	13,92
					2		-	-		719,16	729,30
					3		-	-		2190,55	2200,69
					4		-	-		1125,38	1135,52
2	9 10 11 12 13 14 15 16	3	15	3	1	66,28	19,86	86,14	3,63	4,24	7,87
					2		252,78	319,06		1509,44	1513,07
					3		391,80	458,08		1290,00	1293,63
					4		615,70	681,98		-	-
				4	1	895,93	34,98	930,91	10,34	8,98	19,32
					2		828,33	1724,26		1578,56	1588,90
					3		343,73	343,73		1939,20	1949,54
					4		1512,58	1512,58		-	-
3	17 18 19 20 21 22 23 24	3	15	3	1	64,68	131,50	196,18	3,43	12,82	16,25
					2		-	-		595,32	598,75
					3		-	-		582,52	585,95
					4		-	-		345,79	349,22
				4	1	824,5	225,48	1049,98	9,32	10,59	19,91
					2		-	-		246,36	255,68
					3		-	-		166,31	175,63
					4		-	-		1792,31	1801,63
4	25 26 27 28 29 30 31 32	4	15	3	1	91,48	2,77	94,25	4,83	1,50	6,33
					2		34,16	125,64		12,87	17,70
					3		40,08	40,08		12,09	16,92
					4		56,82	56,82		179,60	184,43
				4	1	1201,43	26,35	1227,78	13,15	2,18	15,33
					2		20,20	20,20		19,63	32,78
					3		50,56	50,56		8,22	21,37
					4		56,04	56,04		217,14	230,29
5	33 34 35 36 37 38 39 40	4	15	3	1	73,18	5,59	78,77	4,18	2,75	6,93
					2		-	-		78,27	82,45
					3		-	-		123,99	128,17
					4		-	-		245,86	250,04
				4	1	900,86	62,59	963,45	11,36	2,73	14,09
					2		-	-		14,06	25,42
					3		-	-		12,51	23,87
					4		-	-		16,91	28,27

Table 3: Solution time of the exact and the heuristic methods for instances set 1

method are solved to optimality for 73 instances. The average computing time does not exceed 316.84 for these instances ranging from 1.40 seconds (instance 42) to 2190.55 seconds (instance 7). For the instances (30 in total) that are solved to optimality by both methods, the time required by CPLEX to solve the MIP of the exact method is on average equal to 116.40 seconds compared to 115.59 seconds for the heuristic.

When considering the total solution time (stages 1 and 2), the exact approach requires 1388.53 seconds on average to solve the 45 instances to optimality. For the 30 instances solved to optimality by both methods, the heuristic needs 122.45 seconds on average where as the exact method requires 1516.59 seconds.

In conclusion, our experimental study reveals that the exact method shows good computational performance for small instances. However, for

Pb	Inst.	Nb_HADC	Nb_Client	Max_Cli	Split	Exact Method			Heuristic method		
						Stage 1	Stage 2	1+2	Stage 1	Stage 2	1+2
6	80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41	4	15	3	1	86,46	4,85	91,31	4,58	2,37	6,95
					2		34,02	120,48		1,40	1,40
					3		54,80	141,26		5,10	5,10
					4		417,43	503,89		4,66	4,66
				4	1	1121,77	22,62	1144,39	13,12	3,79	16,91
					2		46,72	1168,49		17,38	17,38
					3		22,99	1144,76		7,58	7,58
					4		1457,14	2578,91		12,93	12,93
7	80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41	4	20	3	1	231,04	161,80	392,84	6,66	23,91	30,57
					2		-	-		-	-
					3		-	-		-	-
					4		-	-		-	-
				4	1	3579,24	245,94	3825,18	18,99	34,52	53,51
					2		-	-		-	-
					3		-	-		-	-
					4		-	-		-	-
8	80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41	4	20	3	1	206,8	109,08	315,88	6,05	12,93	18,98
					2		2165,36	2372,16		495,62	495,62
					3		-	-		575,78	575,78
					4		-	-		516,99	516,99
				4	1	3688,12	215,96	3904,08	16,41	14,76	31,17
					2		-	-		395,73	395,73
					3		-	-		562,09	562,09
					4		-	-		1040,28	1040,28
9	80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41	5	20	3	1	252,7	10,68	263,38	5,9	3,67	9,57
					2		227,40	480,10		11,76	11,76
					3		303,32	556,02		14,16	14,16
					4		45,01	297,71		9,02	9,02
				4	1	4343,25	25,16	4368,41	14,4	2,42	16,82
					2		78,29	4421,54		15,27	15,27
					3		130,98	4474,23		12,75	12,75
					4		93,35	4436,60		29,50	29,50
10	80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41	5	25	3	1	471,6	148,22	619,82	8,75	21,61	30,36
					2		-	-		347,98	347,98
					3		-	-		178,87	178,87
					4		-	-		-	-
				4	1	10393,42	469,90	10863,32	24,51	18,52	43,03
					2		-	-		656,25	656,25
					3		-	-		405,43	405,43
					4		-	-		970,78	970,78

Table 4: Solution time of the exact and the heuristic methods for instances set 2

large instances, the number of candidate empty routes considerably grows resulting in large computing times for the first stage and intractable MIP models at the second stage. The heuristic method appears as a good alternative in this case since it requires considerably shorter computing time. Although the solution output by this heuristic may be poor in some cases, the deviation with regard to optimal solutions remains relatively small on average.

5.3. Impact of split delivery on solution quality

The objective of this section is to study the impact of split delivery on solution quality. To this end, we consider problem settings 2, 4, 6 and 8 with a value of $Max_{Cli} = 4$ for which an optimal solution was identified for all the values of the split parameter (\mathcal{N}). Recall that a value of 1 means that no

split is permitted, a value of 2, respectively, 3 and 4, assumes that a client can be visited by 2, respectively, 3 and 4, vehicle trips.

To assess the impact of the split parameter on the solution quality, we report in Table 5 the gain (in percentage) on the maximum delivery time resulting from considering a split of value $\mathcal{N} \geq 2$ versus no split. For example, for problem setting 2, enabling a client to be visited twice yields a gain of 7.07% ($= \frac{|67.87-72.67|}{67.87}$) when compared to the case where all the demand of the client must be delivered by a single vehicle trip. This gain reaches 9.66% when $\mathcal{N} = 3$ and 10.33% when $\mathcal{N} = 4$.

pb	Split		
	2	3	4
2	7,07%	9,66%	10,33%
4	8,37%	11,48%	13,10%
6	8,49%	11,65%	13,30%
9	5,45%	5,45%	5,45%

Table 5: Impact of split delivery on solution quality

The results of Table 5 show that a considerable gain (reaching 13.3%) is obtained when permitting split delivery. This gain increases rapidly as the value of \mathcal{N} moves from 1 (no split) to 2 and less rapidly for larger values. When considering the 4 problem settings, the average gain reaches 7.34% for $\mathcal{N} = 2$, 9.56% for $\mathcal{N} = 3$ and 10.54% for $\mathcal{N} = 4$.

However, as noticed in Section 5.2.2, solution times required by both the exact and the heuristic methods also grow when split is permitted and it becomes difficult to identify an optimal solution for large values of \mathcal{N} . Indeed, the exact method was able to locate an optimal solution for all the problem settings when no split is permitted (20 instances in total). However, when $\mathcal{N} = 2$, an optimal solution was identified for 9 instances only. Hence, there is a trade-off that need to be managed in this case to decide on the

suitable value to assign to \mathcal{N} to ensure, on one hand, that solution times are not considerably increased and exploit, on another hand, the gain in distribution time yielded by offering more distribution alternatives.

6. Conclusion

This paper addresses a Vehicle Routing Problem in Disaster Area (VRP-DA). The objective is to distribute humanitarian aid to affected zones from a set of opened humanitarian aid distribution centres using different types of vehicles. Given the particular context of emergency, distribution is planned to satisfy the demand of each affected zone for each type of humanitarian aid in the shortest possible time (this time includes both travelling and vehicles loading and unloading times). We prove that VRP-DA reduces to a classical multi-depot, multi-commodity vehicle routing problem with heterogeneous fleet and split delivery but with some characteristics that are proper to emergency contexts (maximum access time, maximum number of clients visited with a trip, restrict split, etc.)

The paper presents two solution methods for solving VRP-DA: an exact method and a heuristic method. Both methods use a two-stage approach. In the first stage, admissible and promising empty routes are generated so as to minimize the maximum delivery time. These routes are then passed to a MIP in the second stage to select optimal ones and propose an optimal loading of them. The two methods differ on the way empty routes are generated at the first stage. While the exact method generates all admissible and promising routes, the heuristic method restricts the set of candidate routes based on a sweep algorithm.

Experimental results show that the exact method yields optimal solutions in relatively short computing times for small instances. For larger problems, both the first and the second stages require large computing times. On the opposite, the heuristic method remains interesting for large instances requiring relatively short computing times for the majority of instances.

We also investigate the impact of split delivery on solution quality and solution time. Our experiments prove that important gains in delivery times can be achieved when split delivery is permitted. However, when the number of deliveries for a client increases, the problem is harder to solve to optimality. A good alternative could be to permit a restricted split (2 visits maximum per a client for our experiments) to take advantage from split delivery without considerably increasing computing times.

Obviously, working on a subset of candidate empty routes (as is the case for the proposed heuristic) certainly results in more tractable second-stage models, but may also yield poor-quality solutions for VRP-DA. In our future work, we plan to define improvement heuristics to iteratively enrich the set of candidate empty routes by extracting relevant information from solving the MIPs of the second stage.

Acknowledgments

This work was financed by individual grants and by Collaborative Research Development Grants [CG 095123] from the Canadian Natural Sciences and Engineering Research Council (NSERC). This financial support is gratefully acknowledged.

References

- Altay, N., Green III, W. G., 2006. OR/MS research in disaster operations management. *European Journal of Operational Research* 175, 475–493.
- Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W., 2006. The traveling salesman problem. A computational study. *Princeton Series in applied mathematics*.
- Balcik, B., Beamon, B. M., Smilowitz, K., 2008. Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems* 12, 51–63.

- Berkoune, D., Renaud, J., Rekik, M., Ruiz, A., 2012. Transportation in Disaster Response Operations. *Socio-Economic Planning Sciences: Special Issue: Disaster Planning and Logistics: Part 1* 46 (1), 23–32.
- Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F., 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53, 512–522.
- Cornillier, F., Laporte, G., Boctor, F., Renaud, J., 2009. The petrol station replenishment problem with time windows. *Computers and Operations Research* 36, 919–935.
- Golden, B. L., Assad, A. A., 1988. *Vehicle routing: Methods and studies*. Studies in management science and systems.
- Gutin, G., Punnen, A. P., 2007. *The traveling salesman problem and its variations*. Springer.
- Haddow, G. D., Bullock, J. A., Coppola, D. P., 2008. *Introduction to Emergency Management*. Butterworth-Heinemann homeland security series, Elsevier Inc.
- Jotshi, A., Gong, Q., Batta, R., 2009. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Science* 43 (1), 1–24.
- Laporte, G., 1992a. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 231–247.
- Laporte, G., 1992b. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345–358.

- Laporte, G., 2009. Fifty years of vehicle routing. *Transportation Science* 43, 408–416.
- Laporte, G., 2010. A concise guide to the Traveling Salesman Problem. *Journal of the Operational Research Society* 61, 35–40.
- Laporte, G., M., G., J-Y, P., F., S., 2000. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7, 285–300.
- Pessoa, A., Uchoa, E., de Aragao, M. P., 2007. A robust branch-and-cut algorithm for the heterogeneous fleet vehicle routing problem. *Lecture notes in Computer Science* 4525, 150–160.
- Potvin, J.-Y., 2009. Evolutionary algorithms for vehicle routing. *INFORMS Journal on Computing* 21, 518–548.
- Sheu, J.-B., 2007a. An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E* 43, 687–709.
- Sheu, J.-B., 2007b. Challenges of emergency logistics management. *Transportation Research Part E* 43, 655–659.
- Sheu, J.-B., 2010. Dynamic relief-demand management for emergency logistics operations under large-scale disasters. *Transportation Research Part E* 46, 1–17.
- Tzeng, G.-H., Cheng, H.-J., Huang, T. D., 2007. Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E* 43, 673–686.
- Wen, M., Cordeau, J.-F., Laporte, G., Larsen, J., 2010. The dynamic multi-period vehicle routing problem. *Computers and Operations Research* 37, 1615–1623.

- Yi, W., A., K., 2007. Ant colony optimization for disaster relief operations. *Transportation Research Part E* 43, 660–672.
- Yi, W., Özdamar, L., 2007. A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research* 179, 1177–1193.