Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: https://www.tandfonline.com/loi/tjor20

On the analysis of last-mile relief delivery on a tree network: Application to the 2015 Nepal earthquake

Abhinav Khare, Rajan Batta & Jee Eun Kang

To cite this article: Abhinav Khare, Rajan Batta & Jee Eun Kang (2020): On the analysis of last-mile relief delivery on a tree network: Application to the 2015 Nepal earthquake, Journal of the Operational Research Society, DOI: 10.1080/01605682.2019.1708824

To link to this article: https://doi.org/10.1080/01605682.2019.1708824

Published online: 10 Jan 2020.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2019.1708824

ORIGINAL ARTICLE

On the analysis of last-mile relief delivery on a tree network: Application to the 2015 Nepal earthquake

Abhinav Khare, Rajan Batta and Jee Eun Kang

University at Buffalo, Buffalo, NY, USA

ABSTRACT
The last mile delivery in humanitarian relief supply often happens on a tree or an almost-tree network allowing split deliveries. We present a relief delivery model incorporating a tree network for last mile delivery. We developed a mixed integer programming (MIP) formulation with the goal of minimizing the unsatisfied demand of the population. For better computational performance, we reformulated the MIP exploiting the tree network structure and found that this gave an order of magnitude reduction in computational time. To further improve computational efficiency, we developed a heuristic solution method based on a decomposition scheme applied to the tree network formulation. This led to the Capacitated Vehicle Routing Problem on trees with split deliveries, for which we derived a closed-form solution. This decomposition scheme resulted in a further order of magnitude reduction in computation time. To demonstrate the application of our approach we applied our model to the humanitarian logistics relief operation encountered in the 2015 Nepal earthquake.

ARTICLE HISTORY
Received 14 May 2019
Accepted 17 December 2019

KEYWORDS
Tree graphs; split delivery; last mile delivery; mixed-integer programming

1. Introduction and contribution

According to the World Disasters Report 2016 of the International Federation of Red Cross and Red Crescent Societies, 698,077 people died and damage worth 1399 billion dollars was caused by natural disasters around the world in the period 2006–2015 (International Red Cross Society, 2016). Supply of relief after a disaster is crucial for saving lives and enabling faster recovery. In practice, most humanitarian logistics operations are conducted in two stages, each over its own transportation network. In the first stage, a primary network (air, road, water) is used for transporting aid from different humanitarian organizations and countries to identified staging areas/depots in the vicinity of the affected region. In the second stage, a secondary network (road) is used for last mile delivery from the depots to the affected people. However, in most disasters, the secondary network present in affected areas is partially or fully compromised (Kovács & Spens, 2007; Luis, Dolinskaya, & Smilowitz, 2012). For instance, following a severe earthquake, a densely connected road network (Figure 1(a)) is often reduced to a sparse network lacking full connectivity (Figure 1(b)). To establish minimal connectivity, emergency repairs are usually performed which results in a tree or an almost-tree structure (Figure 1(c)). This is consistent with the literature which states that tree networks arise when the cost of road construction is much more than the routing cost (Kumar, Unnikrishnan, & Waller, 2012), which is true in the case of disaster relief. M3otivated by this observation, we focus our analysis in this paper under the assumption that the secondary network is a tree. We also assume that split deliveries are allowed in the our vehicle routing model. In rescue and relief operations on the ground post disasters, split delivery is observed in most scenarios. This feature is required because in a lot of disaster relief delivery scenarios the demand at a node in a period exceeds the capacity of one vehicle. Therefore, multiple vehicles are utilised to serve a node. We also found that these two assumptions to be true in multiple cases including the relief logistics operations for the Nepal Earthquake in 2015, the second stage network were almost-tree structured mountain trail and split delivery happened using porters and pack animals (World Food Program-3, 2016). Assuming the tree structure is not only realistic in cases of network damage and remote mountain transportation but also allows us to develop efficient solution algorithms for relief supply optimization, which we believe have wide applicability in disaster relief.

We develop a mixed integer program (MIP) that captures both stages of post-disaster relief in a single model. It is a multinetwork, multiperiod relief supply model, and its solution using a standard commercial software like CPLEX is not practical except for toy instances. To address this computational concern, we reformulate the model under the assumption that the second stage network is a tree.

CONTACT Rajan Batta batta@buffalo.edu Department of Industrial & Systems Engineering, University at Buffalo, Buffalo, NY 14260, USA

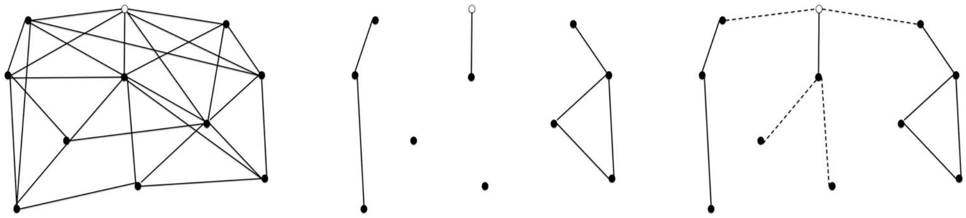

Figure 1. (a) A densely connected network (b) the network after a disaster (c) the network to ensure minimum connectivity from the depot.

Applying CPLEX to the tree formulation results in an order of magnitude improvement in computational efficiency but still leaves us at a point where realistic sized problems cannot be solved. For this reason, we develop a heuristic solution method for the tree network case based on decomposition principles. This results in a further order of magnitude reduction in computational effort. Our principle technical contribution is the development this efficient solution method for the case where the second stage network in post-disaster relief has a tree structure and split delivery is allowed. In this decomposition solution approach, we introduced the Capacitated Vehicle Routing Problem on trees with split deliveries (TCVRP-SD) and derived a closed-form solution for its optimal objective function value. Using this, we not only develop a computationally efficient solution method but also extend the literature on vehicle routing problem on trees (TVRPS) and vehicle routing model on split delivery (SDVRP).

Our next contribution is the application of our methodology on the 2015 Nepal Earthquake relief supply operations. In this earthquake, the second stage network consisted of mountain trails, many of which were compromised. The actual trails used for supplies closely resembled a tree network, and transportation over these trails was done by secondary vehicles and pack animals. The first stage network consisted of primary vehicles and staging areas, where the primary vehicles delivered relief supplies as well as the secondary vehicles. Our case study is based on the Nepal scenario. It is important to note that our method can also be used to generate heuristic solutions for cases where the second-stage network is an almost tree (i.e. a tree network with some additional arcs and a few cycles). This is why we believe that our findings have wide applicability especially in a wide range of natural disaster settings.

2. Literature review

In our methodology to solve the decomposition of our model, we extend the work on vehicle routing problem (SDVRP) with split deliveries to tree networks. Therefore, we divide our literature review into three parts. The first part focuses on vehicle routing problems on tree networks. We have used the material in these papers to develop the model with tree route formulation. Next, we discuss the literature on SDVRP. The third part focuses on last-mile delivery in humanitarian relief operations.

2.1. Vehicle routing on tree networks

Research on vehicle routing on tree shaped networks is scant. Labbé, Laporte, and Mercure (1991), published seminal work on the Capacitated Vehicle Routing on Trees in which they defined the Capacitated Vehicle Routing Problem on Trees (TCVRP) which defined the problem, showed it to be NP-hard and proivded the first solution. Basnet, Foulds, and Wilson (1999) developed two new heuristics to solve TCVRP and compared their computational performance to heuristics developed by Labbé, Hamaguchi and Katoh (1998) solved a version of TCVRP that seeks a set of tours of the vehicles with minimum total lengths. Mulsea presented five variants of TCVRP and five fast approximation algorithms to solve it (Muslea, 1997). Chandran and Raghavan presented a new paradigm in modeling TCVRP's (Chandran & Raghavan, 2008) in which they used the tree-like structure in modeling the constraints of the model and developed a bin-packing heuristic to solve their TCVRP. Kumar, Unnikrishnan, and Waller (2011) proposed the Capacitated Vehicle Routing on Trees with Backhaul (TCVRPB) in which line-haul customers require delivery from the depot, and back-haul customers have supply that needs to be delivered to the depot and also developed a 2-approximation algorithm to solve the problem.

2.2. Vehicle routing problem with split deliveries

Vehicle routing problems have the underlying assumption that each node is visited only by one vehicle. However, vehicle routing problem with split deliveries introduced by Dror and Trodeau in 1989–1990 (1989, 1990) relaxes this assumption and allows a node to be visited by multiple vehicles. They showed this leads to cost savings over VRPs. The literature on this topic is vast and there are three major surveys on this topic (Archetti & Speranza, 2008, 2012; Gulczynski, Golden, & Wasil, 2008) which discuss SDVRP, its proposed solution, its variants and its application in multiple scenarios. Archetti, Speranza, and Hertz (2006a) proved that if there is a feasible solution to the SDVRP, then, there always exist an optimal integer solution. It was also proven vehicle capacity equal to two SDVRP has a polynomial time algorithm to solve and for vehicle capacity greater than equal to three, SDVRP is NP-hard (Archetti, Mansini, & Speranza, 2005). Archetti, Savelsbergh, and Speranza (2006b) found the maximum possible savings with SDVRP and also showed how that depended on the size of the problem (Archetti & Speranza, 2008). To solve SDVRP exact approaches have been suggested that include the arc flow formulation, the Dror and Trudeau heuristic (Dror, Laporte, & Trudeau, 1994). Then, there have been the cutting plane approach by Belenguer, Martinez, and Mota (2000) and a two stage algorithm with valid inequalities (Jin, Liu, & Bowden, 2007) which provided good lower bounds followed by multiple column generation approaches (Archetti, Bianchessi, & Speranza, 2011a; Jin, Liu, & Eksioglu, 2008; Moreno, De AragãO, & Uchoa, 2010) that improved the lower bounds. Multiple heuristics have also been proposed that include a memetic algorithm (Boudia, Prins, and Reghioui, 2007), hybrid algorithms (Archetti, Speranza, & Savelsbergh, 2008; Chen, Golden, & Wasil, 2007; Khmelev & Kochetov, 2015), scatter search (Mota, Campos, & Corberán, 2007), an adaptive memory algorithm (Aleman, Zhang, & Hill, 2010), a tabu search with building vocabulary approach (Aleman & Hill, 2010), attribute-based hill climbing approach (Derigs, Li, & Vogel, 2010), an iterated local search approach (Silva, Subramanian, & Ochi, 2015), branch-and-cut approach (Archetti, Bianchessi, & Speranza, 2014a), a randomized granular tabu search (Berbotto, García, & Nogales, 2014), the novel priori splitting approach (Chen, Golden, Wang, & Wasil, 2017), and the latest maximum–minimum distance clustering method (Min, Jin, & Lu, 2019).

2.3. Last-mile delivery in humanitarian logistics

The area of humanitarian logistics is rich with models and hence we mention a few important ones. Balcik and Beamon (2008) developed a two-phase multiperiod model for relief distribution (Gibbons & Samaddar, 2009). Berkoune, Renaud, Rekik, and Ruiz (2012), modeled a multicommodity and multidepot routing problem for minimizing the total duration of all trips for last-mile delivery. Using a genetic algorithm. Tzeng, Cheng, and Huang (2007) solved a multiperiod aid distribution problem using fuzzy multiobjective linear programming. Vitoriano, Ortuño, Tirado, and Montero (2011) also proposed a multiciteria problem described using a double-flow model and solved it using goal programming. Ozdamar (2011) developed a model for last-mile relief delivery using helicopters. More recent models include stochastic optimization models (Noyan, Balcik, & Atakan, 2016; Noyan & Kahvecioğlu, 2018), multicriteria model (Ferrer, Martín-Campo, Ortuño, Pedraza-Martínez, Tirado, & Vitoriano, 2018) and models utilizing drones for last mile delivery (Chowdhury, Emelogu, Marufuzzaman, Nurre, & Bian, 2017; Rabta, Wankmüller, & Reiner, 2018) These are a sample of papers from the vast literature on the last mile distribution in humanitarian relief.

In reviewing the humanitarian logistics literature for last mile delivery we found a rich body for modeling the last mile delivery problem. Literature on SDVRPS is also vast and we also found its application in last mile delivery post disasters like earthquake (Wang, Du, & Ma, 2014; Yi & Özdamar, 2007). These application show that splitting deliveries saves cost and also minimizes the vehicle use which are sometimes scarce in post disaster scenarios. While we clearly see almost-tree transportation networks due to damage post-disasters like in case of Nepal earthquake, we did not find any work that applied the model of vehicle routing on trees to last mile distribution problem. We fill this gap in the literature. We first utilize existing methodology Chandran and Raghavan (2008) to develop a reformulation of our model that exploits the underlying tree structure of the network. We further developed a decomposition of the reformulated model. To solve the reformulated, we extend the literature on both TCVRPs and SDCVRP by developing a variant, we call the vehicle routing on trees with split deliveries, SD-TCVRP. We also find an exact algorithm to solve it and a closed form solution for the optimal objective. Using this, we reduce the computational time 20 orders of magnitude as compared on toy datasets. Using this method real sized problems are also solved within 1 h.

The last mile delivery problems are fraught with data uncertainty and typically have many scenarios. In fact, a common way to solve these problems is to perform scenario analysis where solutions for hundreds of scenarios are developed and analyzed. Thus, there is a critical need for reducing the computational effort associated with developing the solution for a specific scenario. We show that by exploiting the tree network structure of the last mile delivery network we can reduce the computation time by orders of magnitude.

3. A two-stage relief delivery model

The following are the salient features of our model:

1. **Primary Depot:** This depot serves as a staging area for the relief brought in from humanitarian organizations and aiding countries for onward movement into the affected regions. Set of all nodes is denoted by $V = \{0, 1, 2, ...N - 1\}$. The node $\{0\}$ is the primary depot.
2. **Secondary Depots:** These depots are staging areas identified in the vicinity of the affected region and store relief supplies brought from the primary depot. The secondary depots also have demand and hence they are subset of demand node set $V - \{0\}$. S denotes the secondary depots.
3. **Primary Network:** The network to carry the relief supply from a primary depot to the secondary depots via primary vehicles. Primary network contains the primary depot $\{0\}$ and the secondary depots S.
4. **Secondary Network:** The network used for last mile delivery from the secondary depots to the demand nodes in the affected region. Node set $V - \{0\}$ are the demand nodes and define the secondary network. It must be noted that the secondary depots is a subset of demand nodes.
5. **Multiperiod:** The relief supply operations is done in multiple discrete time periods. T is the set of all time periods.
6. **Multicommodity:** Transport of multiple commodities like water, food, sanitation and medical supplies is possible. $C = \{0, 1, 2\}$ is the set of relief supplies including water, food, shelter. Index $c = 0$ represent a load for 1 kg of water, $c = 1$ represent load for 1 kg of food, $c = 2$ represent load for 1 Kg of shelter supplies.
7. **Split Deliveries:** More one vehicle can serve a demand node or a secondary depot in a period.
8. **Storage:** Storage of relief items (transported from the primary depot) is available at the secondary depots.

3.1. Decision variables

$y^t_{ijl} \in \{0, 1\}$ is 1 if primary vehicle l travels from node i to node j in period t, and 0 otherwise,

$x^t_{ijk} \in \{0, 1\}$ is 1 if secondary vehicle k travels from node i to node j in period t, and 0 otherwise,

$r^t_{ik} \in \{0, 1\}$ is 1 if secondary vehicle k is present at depot i in period t, and 0 otherwise,

$z^{tc}_{ijk} \in \{0, 1, 2..\}$ is the load of commodity c taken by secondary vehicles k from node i to node j in period t,

$v^{tc}_{ijl} \in \{0, 1, 2..\}$ is the load of commodity c taken by primary vehicle l from node i to node j in period t,

$d^{tc}_{i} \in \{0, 1, 2..\}$ is the total load of commodity c dropped by secondary vehicles at node i in period t,

$d^{tc}_{ik} \in \{0, 1, 2..\}$ is the load of commodity c dropped at node i by secondary vehicle k in period t,

$D^{tc}_{i} \in \{0, 1, 2..\}$ is the total load of commodity c dropped at node i by primary vehicle s in period t,

$S^{tc}_{i} \in \{0, 1, 2..\}$ is the total load of commodity c stored at node i at the end of period t, $i \neq j; i, j \in V = \{0, 2.......N - 1\}$, where V is the set of all N nodes including demand nodes, primary depot 0 and secondary depots,

$c \in C = \{0, 1, 2\}$, where C is the set of relief supplies including water, food, shelter. Index $c = 0$ represent a load of 1 kg of water, $c = 1$ represent load of 1 kg of food, $c = 2$ represent load of 1 Kg of shelter supplies.

3.2. Parameters

N, total number of nodes

S, set of all secondary depots

L, set of primary vehicles available at the primary depot

K, set of all secondary vehicles available at the secondary depots

T, set of all time periods

D, total number of items to be delivered

Up, parameter for upper bound used in equity constraint

Lo, parameter for lower bound used in equity constraint

a_{ij}, travel time for secondary vehicles from node i to node j

h_{ij}, travel time for primary vehicles from node i to node j

m^{tc}_{i}, demand of commodity c at node i in period t

q_s, capacity of a secondary vehicle k

q_l, capacity of a primary vehicle l

f_i, service time for a secondary vehicle at demand nodes i

g_i, service time for a primary vehicle at node i

r_k, maximum route time allowed for secondary vehicle k

r_h, maximum route time allowed for primary vehicle h

s_k, average speed of secondary vehicle k

s_h, average speed of primary vehicle h

B^{tc}, amount of item c available at primary depot for period t

3.3. General model

Maximize: $\sum_{i\in V}\sum_{t\in T}\sum_{c\in C} d_i^{tc} - \sum_{i\in V}\sum_{j\in V, i\neq j}\sum_{k\in K}\sum_{t\in T}\alpha a_{ij}x_{ijk}^t - \sum_{i\in S}\sum_{j\in S, j\neq i}\sum_{l\in L}\sum_{t\in T}\alpha h_{ij}y_{ijl}^t$

St:

Primary Vehicle Routing Constraints:

$$\sum_{i\in S\cup\{0\}, i\neq j} y_{ijl}^t = \sum_{i\in S\cup\{0\}, i\neq j} y_{jil}^t \qquad \forall j\in S\cup\{0\},\ \ l\in L,\ \ t\in T \tag{1}$$

$$\sum_{j\in S} y_{0jl}^t = 1 \qquad \forall l\in L, t\in T \tag{2}$$

$$\sum_{i\in W}\sum_{j\in W, j\neq i} y_{ijl}^t \le |W|-1 \qquad \forall W\subseteq S,\ \ |W|\ge 2,\ \ l\in L,\ \ t\in T \tag{3}$$

$$\sum_{j\in S}\sum_{c\in C} v_{0jl}^{tc} \le q_l \qquad \forall l\in L, t\in T \tag{4}$$

$$\sum_{i\in S\cup\{0\}}\sum_{j\in S, j\neq i} y_{ijl}^t(h_{ij}+g_i) \le r_h/s_h \qquad \forall l\in L, t\in T \tag{5}$$

Secondary Vehicles Routing Constraints:

$$\sum_{i\in V-\{0\}, i\neq j} x_{ijk}^t = \sum_{i\in V-\{0\}, i\neq j} x_{jik}^t \qquad \forall j\in V-\{0\},\ \ k\in K,\ \ t\in T \tag{6}$$

$$\sum_{j\in V-\{0\}} x_{ijk}^t = 1 \qquad \forall i\in S, k\in K, t\in T \tag{7}$$

$$\sum_{i\in U}\sum_{j\in U, j\neq i} x_{ijk}^t \le |U|-1 \qquad \forall U\subseteq (V-S)-\{0\}, |U|\ge 2, k\in K, t\in T \tag{8}$$

$$\sum_{j\in(V-S)-\{0\}}\sum_{c\in C} z_{ijk}^{tc} \le q_s \qquad \forall i=S, k\in K, t\in T \tag{9}$$

$$\sum_{i\in(V-S)-\{0\}}\sum_{j\in V-\{0\}, j\neq i} x_{ijk}^t(t_{ij}+f_i) \le r_k/s_k \qquad \forall k\in K, t\in T \tag{10}$$

$$\sum_{i\in V-\{0\}} x_{ijk}^t \le 1 \qquad j\in S, k\in K, t\in T \tag{11}$$

$$x_{ijk}^t = 0 \qquad \forall i\in S, j\in S\cup\{0\}, k\in K, t\in T \tag{12}$$

Loading and Delivery Constraints:

$$\sum_{c\in C} z_{ijk}^{tc} \le M x_{ijk}^t \qquad \forall i\in V-\{0\}, j\in V-\{0\}, k\in K, t\in T \tag{13}$$

$$\sum_{c\in C} v_{ijl}^{tc} \le M y_{ijl}^t \qquad \forall i\in S\cup\{0\}, j\in S\cup\{0\}, l\in L, t\in T \tag{14}$$

$$\sum_{j\in V-\{0\}}\sum_{k\in K} z_{ijk}^{tc} - \sum_{j\in V-\{0\}}\sum_{k\in K} z_{jik}^{tc} = d_i^{tc} \qquad \forall i\in (V-S)-\{0\}, t\in T, c\in C \tag{15}$$

$$\sum_{j\in S\cup\{0\}}\sum_{l\in L} v_{ijl}^{tc} - \sum_{j\in S\cup\{0\}}\sum_{l\in L} v_{jil}^{tc} = D_i^{tc} \qquad \forall i\in S, t\in T, c\in C \tag{16}$$

$$D_0^{tc} = 0 \qquad \forall t\in T, c\in C \tag{17}$$

$$d_i^{tc} \le m_i^{tc} \qquad \forall i\in V-\{0\}, t\in T, c\in C \tag{18}$$

$$\sum_{k\in K} d_{ik}^{tc} \le d_i^{tc} \qquad \forall i\in V-\{0\}, t\in T, c\in C \tag{19}$$

$$m_i^{tc} - d_i^{tc} \le Up/N \sum_{c\in C}\sum_{j\in V-\{0\}} (m_j^{tc} - d_j^{tc}) \qquad \forall i\in V-\{0\}, t\in T, c\in C \tag{20}$$

$$m_i^{tc} - d_i^{tc} \ge Lo/N \sum_{c\in C}\sum_{j\in V-\{0\}} (m_j^{tc} - d_j^{tc}) \qquad \forall i\in V-\{0\}, t\in T, c\in C \tag{21}$$

$$\sum_{c\in C} z_{ijk}^{tc} = 0 \qquad \forall i\in V-S, j\in S, k\in K, t\in T \tag{22}$$

$$\sum_{c\in C} v_{i0l}^{tc} = 0 \qquad \forall i\in S, k\in K, t\in T \tag{23}$$

$$\sum_{c\in C} d_{ik}^{tc} \ge r_{ik}^t \qquad \forall i\in V-\{0\}, k\in K, t\in T \tag{24}$$

$$\sum_{c\in C} d_{ik}^{tc} \le M r_{ik}^t \qquad \forall i\in V-\{0\}, k\in K, t\in T \tag{25}$$

$$\sum_{i\in S-\{0\}} r_{ik}^t = 0 \qquad \forall i\in V, k\in K, t\in T, c\in C \tag{26}$$

Storage Constraints:

$$S_i^{tc} = S_i^{t-1c} + D_i^{tc} - \sum_{j\in V-S, j\neq i}\sum_{k\in K} z_{ijk}^{tc} - d_i^{tc} \qquad \forall i\in S-\{0\}, t\in T-\{0\}, c\in C \tag{27}$$

$$S_i^{0c} = D_i^{0c} - \sum_{j\in S-\{0\}, j\neq i}\sum_{l\in L} z_{ijk}^{0c} - d_i^{0c} \qquad \forall i\in S-\{0\}, c\in C \tag{28}$$

$$S_0^{tc} = S_0^{t-1c} - \sum_{j\in S-\{0\}, j\neq i}\sum_{l\in L} v_{0jl}^{tc} + B^{tc} \qquad \forall i\in S-\{0\}, t\in T, c\in C \tag{29}$$

$$S_0^{0c} = -\sum_{j\in S-\{0\}, j\neq i}\sum_{l\in L} v_{0jl}^{0c} + B^{0c} \qquad \forall c\in C \tag{30}$$

3.4. Model explanation

Maximize: $\sum_{i\in V}\sum_{t\in T}\sum_{c\in C} d_i^{tc} - \sum_{i\in V}\sum_{j\in V, i\neq j}\sum_{k\in K}\sum_{t\in T}\alpha a_{ij}x_{ijk}^t - \sum_{i\in S}\sum_{j\in S, j\neq i}\sum_{l\in L}\sum_{t\in T}\alpha\ h_{ij}\ y_{ijl}^t$ (*)

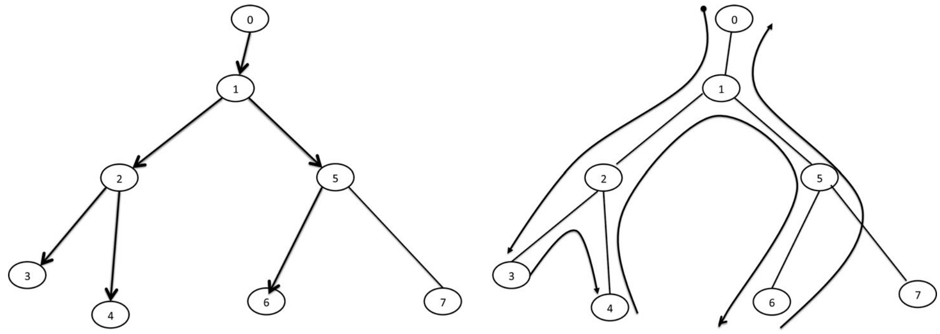

Figure 2. (a) Route tree of TCVRP (b) Depth first order of route.

Consider the objective function given by equation $(*)$. The term m_i^{tc} can be removed for optimization purposes as it is a constant. Minimizing unsatisfied demand summed over all commodities, all villages and all periods given by $\sum_{i\in V}\sum_{t\in T}\sum_{c\in C}(m_i^{tc} - d_i^{tc})$ is our primary objective. Minimizing travel cost is our secondary objective. We prioritize our primary objective over the secondary objective by multiplying the secondary objective term by a hyper-parameter α. We vary the values of α while testing the solution methods, to find an appropriate range of α for prioritizing the primary objective over secondary objective for a given dataset. This is discussed in detail in our case study.

3.4.1. Constraints

Constraints $(1)-(5)$ are primary vehicle routing constraints for routing primary vehicles on the primary network on the node set, $S \cup \{0\}$. Constraints $(1)-(2)$ ensure that the route for a primary vehicle is a cycle and that a primary vehicle starts and ends its tour at the primary depot. Constraint (3) is for subtour elimination for the primary vehicle. Constraint (4) is the primary vehicle capacity constraint ensuring that the primary vehicle never carries more than its capacity. Constraint (5) ensures that the primary vehicle does not cross its limit of travel in a period. Constraints $(6)-(12)$ are equivalent routing constraints to $(1)-(5)$ for the secondary vehicles on the secondary network on the node set, $V-\{0\}$.

Constraints $(13)-(26)$ govern the load carried and delivery made by primary and secondary vehicles. Constraint (13) ensure that a a load on an arc occurs if a secondary vehicle is traversing it. Constraint (14) do the same for primary vehicles. Constraints $(15)-(16)$ are the relationships between delivery and load variables for secondary vehicles and primary vehicles, respectively. Constraint (17) ensures that there is no delivery at a primary depot. Constraint (18) ensures that delivery of an item to a demand node in a period is less than its demand in that period. Constraint (19) ensures that the total delivery of an item in a demand node for a period is the sum of the items delivered to it by each of the visiting secondary vehicles. Constraints $(20)-(21)$ are equity constraints that ensure that the unsatisfied demand at each village is within a lower and upper bound of the mean unsatisfied demand. Constraints $(22)-(23)$ ensure that the load variables entering the depots are zero, i.e. the vehicles deliver all their load in each period. Constraints $(24)-(25)$ ensure that secondary vehicles deliver their load to the demand nodes that it visits. Constraint (26) ensures that each secondary vehicle serves its assigned secondary depot during each period.

The constraints $(27)-(30)$ are the storage-related constraints. The constraints $(27)-(28)$ establish the relationship between storage variables, delivery variables, and load variables at the secondary depots. Constraints $(29)-(30)$ do the same for the primary depot.

We found that using a commercial solver like CPLEX on this formulation does not yield an optimal or near-optimal solution in 24 h of computation time for realistic datasets like the one we have used in our case study. Exploiting the fact that many secondary networks associated with humanitarian relief efforts have a tree or almost-tree like structure, we now present a formulation of our model for the case where the secondary network is a tree, thereby alleviating the computational difficulty of the problem.

4. Model with tree formulation

In this section, we model the constraints of routing secondary vehicles on tree networks using the properties of vehicle routes on trees as described by Chandran and Raghavan (2008) illustrating the concept through an example. Given a set of nodes serviced by a vehicle in a tree network (2, 3, 4, and 6), the arcs travelled by the vehicle form a tree as shown by the directed arcs in Figure 2(a) and the tour is given by visiting the nodes in a depth first manner as shown in Figure 2(b) (Chandran & Raghavan, 2008). We use these properties to remodel the secondary vehicle routing constraints. Consider a node i in this tree network. It has a

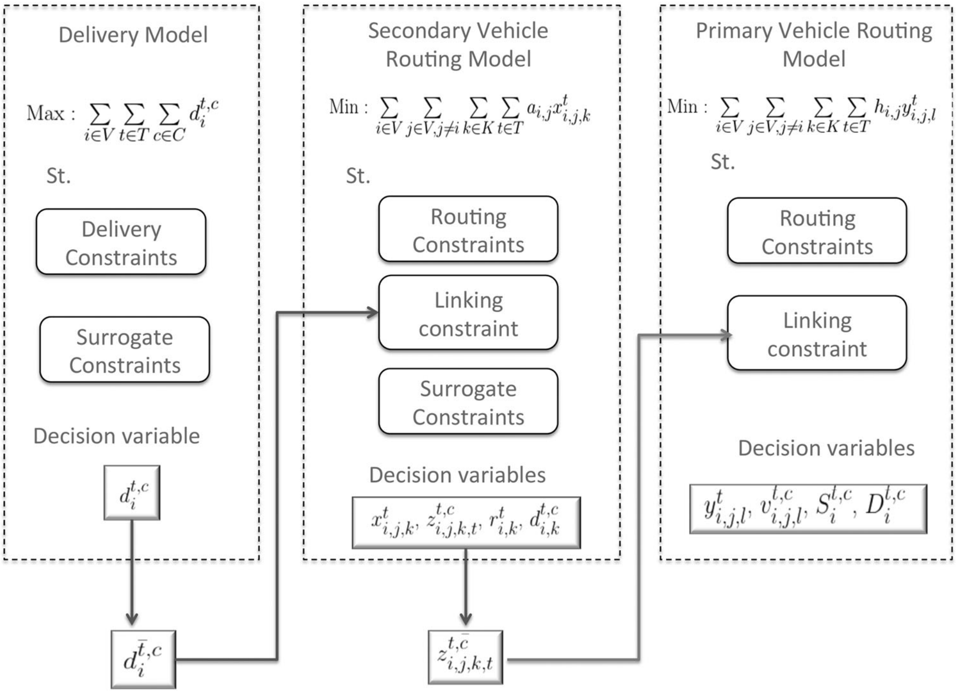

Figure 3. Schematic explaining the solution method for the decomposition.

unique parent $P(i)$ and a set of children $Ch(i)$. The route is constructed for each vehicle by "building upwards" from every node serviced by the vehicle towards the depot as shown in Figure 2(b). This generates a covering subtree for each secondary vehicle. Having generated the tree, i.e. the arcs the vehicles take, the route of the vehicles is the nodes visited in the depth first order. Using this, we replace four constraints $(6)-(8), (10)-(12)$ in our original model by three constraints $(31)-(33)$. Constraints $(15), (27), (28)$ are replaced by constraints $(34)-(36)$. Following are the reformulated constraints.

$$x^t_{P(i),ik} \geq x^t_{ijk} \quad \forall i \in (V-S)-\{0\}, j \in Ch(i), k \in K, t \in T \tag{31}$$

$$x^t_{P(i),ik} \geq r^t_{ik} \quad \forall i \in (V-S)-\{0\}, k \in K, t \in T \tag{32}$$

$$x^t_{ijk} = x^t_{jik} \quad \forall i \in V-\{0\}, j \in V-\{0\}, k \in K, t \in T \tag{33}$$

$$\sum_{j\in Ch(i)}\sum_{k\in K} z^{tc}_{ijk} - \sum_{j\in P(i)}\sum_{k\in K} z^{tc}_{jik} = d^{tc}_i \quad \forall i \in (V-S)-\{0\}, t \in T, c \in C \tag{34}$$

$$S^{tc}_i = S^{t-1c}_i + D^{tc}_i - \sum_{j\in Ch(i),}\sum_{k\in K} z^{tc}_{ijk} - d^{tc}_i \quad \forall i \in S, t \in T-\{0\}, c \in C \tag{35}$$

$$S^{tc}_i = D^{tc}_i - \sum_{j\in Ch(i)}\sum_{l\in L} z^{tc}_{ijk} - d^{tc}_i \quad \forall i \in S, t = 0, c \in C \tag{36}$$

The constraint (31) ensures that if a secondary vehicles k visits node i in period t, it travels on arc $(P(i), i)$. The constraint (32) populates the rest of the tree bottom up to the depot, using the fact that if a secondary vehicles uses arc (ij) it will have to use the arc $(P(i), i)$. The constraint (33) makes sure that arcs are travelled twice by a secondary vehicle, once while going down the tree and once while coming up. Constraints (34)–(36) remove summations with the load variables as they are automatically enforced in a tree network.

The tree formulation can be used for situations where the secondary network is an almost-tree by removing arcs from the network to form a tree and by using the resultant tree as an approximation to the secondary network. We tried this approach and found that using a commercial solver like CPLEX on this new model provides an order of magnitude (10-fold) improvement in computational time but still is not suitable for use in realistic sized datasets that typically have to be run several times using different parameter settings. Therefore, we now focus on a decomposition-based method which is suitable for realistic datasets.

5. A decomposition-based solution method

We decompose the model into three parts: (1) delivery model, (2) capacitated secondary vehicle routing model and, (3) capacitated primary vehicle routing model. The model is separable into aforementioned three different parts because the decision variable for delivery (d_i^{tc}), the decision variables for secondary vehicle routing ($x^t_{ijk}, z^{tc}_{ijk}, r^t_{ik}, d^{tc}_{ik}$), and the decision variables for primary vehicle routing ($y^t_{ij,l}, v_i ijl^{tc}, S_i^{tc}$) in the original model are linked only via very few linking constraints. Constraint (34) is the linking constraint between delivery variables and secondary vehicle routing variables. Constraints (35)–(36) are the linking constraints between delivery variables, secondary routing and the primary variables. In a scenario where there are small number of linking constraints between the separable models, it is possible to closely approximate the optimal solution of the original model by adding surrogate constraints to the separable models and solving them individually. The surrogate constraints are added to the separable models to provide the upper bounds to the decision variables. In general, "a surrogate constraint is an inequality implied by the constraints of an integer program, and is designed to capture useful information that cannot be extracted from the parent constraints individually but is nevertheless is a consequence of their conjunction" (Glover, 1968). In our scenario, the useful information is the upper bounds to the decision variables of the decomposed models. These bounds are implicit in the original model, and are added explicitly as surrogate constraints in the decomposed models. The surrogate constraints in one model are obtained by conjunction of its linking constraints with constraints of other two models and provide an upper bound to the decision variables.

Subsections 5.1–5.3 describes the decomposed models individually and their solution methods. Figure 3 explains this process of combining their solutions using a schematic. We proceed by adding surrogate constraints to the delivery model and secondary vehicle routing models. Constraints (38)–(40) described in subsection 5.1 are the surrogate constraints that provide the upper bounds on decision variables d_i^{tc}, which are implicit in the original model. The constraints (56)–(57) described in subsection 5.2 are the surrogate constraints that provide the upper bounds to z_{ijk}^{tc}. These models are sequentially solved in that the solutions of one model becomes the input parameters of the following model. Hence, the optimal solution of the delivery variables $\bar{d}_i^{tc}$ becomes the input parameters for the capacitated secondary vehicle routing model, and so on. The linking constraints (34) are converted to constraint (49) in subsection 5.2, with $\bar{d}_i^{tc}$ as a parameter in the secondary vehicle routing model. Similarly, constraints (35)–(36) are converted to constraints (65)–(66) in the primary vehicle routing model, with $\bar{d}_i^{tc}$ and $\bar{z}_{ijk}^{tc}$ as parameters.

5.1. Delivery model

The delivery model (DM) determines the allocation of commodities to the demand nodes in each period. It uses the decision variables d_i^{tc}. The objective is given by $\sum_{i\in V}\sum_{t\in T}\sum_{c\in C} d_i^{tc}$. The model can be written as:

Maximize: $\sum_{t\in T}\sum_{i\in V}\sum_{cinC} d_i^{tc}$ **st:**

$$d_i^{tc} \le m_i^{tc} \qquad \forall i \in V-\{0\}, t\in T, c\in C \tag{37}$$

$$\sum_{i\in V-\{0\}}\sum_{c\in C} d_i^{tc} \le |K|q_s \qquad \forall t\in T \tag{38}$$

$$\sum_{i\in V-\{0\}}\sum_{Q\in\{1,2\ldots t\}}\sum_{c\in C} d_i^{Qc} \le |Q||L|q_l \qquad \forall t\in T \tag{39}$$

$$\sum_{i\in V-\{0\}}\sum_{Q\in\{1,2\ldots t\}} d_i^{Qc} \le \sum_{Q\in\{1,2\ldots t\}} B^{Qc} \qquad \forall t\in T, c\in C \tag{40}$$

$$m_i^{tc}-d_i^{tc} \le Up/N\sum_{c\in C}\sum_{j\in V-\{0\}}(m_j^{tc}-d_j^{tc}) \quad \forall i\in V-\{0\}, t\in T, c\in C \tag{41}$$

$$m_i^{tc}-d_i^{tc} \ge Lo/N\sum_{c\in C}\sum_{j\in V-\{0\}}(m_j^{tc}-d_j^{tc}) \quad \forall i\in V-\{0\}, t\in T, c\in C \tag{42}$$

Constraint (37) ensures that the amount of a commodity delivered to each node in each period, d_i^{tc}, does not exceed its demand. Constraint (38) ensures that the total primary vehicle capacity $|K|q_s$ is not exceeded in each period. Constraint (39) ensures that the cumulative capacity of primary vehicles $|Q||L|q_l$ (cumulative over time period) is not exceeded. Constraint (40) ensures that the primary depot does not violate its delivery capacity by requiring that the cumulative delivery to all nodes is less than or equal to the cumulative availability at the primary depot. Constraints (41)–(42) are the equity constraints used in the original and the tree model.

We note that the delivery problem is an extension of the Multiple Bounded Knapsack Problem, described by Detti (2009). We developed a special purpose algorithm, Algorithm 1, for solving the delivery model. The procedure and description is available in subsection Algorithm 1 of the Appendix.

5.2. Capacitated secondary vehicle routing model

The capacitated secondary vehicle routing model finds the minimum cost tours for the secondary vehicles given the output from the delivery model. The decision variables are $x_{ijk}^t, z_{ijk}^{tc}, r_{ik}^t, d_{ik}^{tc}$. The objective is minimizing the cost of transportation for the secondary vehicles. The constraints are the routing and loading constraints for the secondary vehicles from the model with tree formulation. $\bar{d}_i^{tc}$'s are the values of the decision variables d_i^{tc} as output from solution method of the delivery model and is input into the secondary vehicle routing problem as parameter.

Minimize: $\sum_{i\in V}\sum_{j\in V, j\neq i}\sum_{k\in K}\sum_{t\in T} a_{ij}x^t_{ijk}$

st:

$$x^t_{P(i),ik} \geq x^t_{ijk} \quad \forall i \in (V-S)-\{0\}, j \in Ch(i), k \in K, t \in T \tag{43}$$

$$x^t_{P(i),ik} \geq r^t_{ik} \quad \forall i \in (V-S)-\{0\}, k \in K, t \in T \tag{44}$$

$$x^t_{ijk} = x^t_{jik} \quad \forall i \in V-\{0\}, j \in V-\{0\}, k \in K, t \in T \tag{45}$$

$$\sum_{j\in V-S}\sum_{c\in C} z^{tc}_{0jk} \leq q_s \quad \forall k \in K, t \in T \tag{46}$$

$$\sum_{i\in(V-S)-\{0\}}\sum_{j\in V-\{0\}, j\neq i} x^t_{ijk}(t_{ij}+f_i) \leq r_k/s_k \quad \forall k \in K, t \in T \tag{47}$$

$$\sum_{c\in C} z^{tc}_{ijk} \leq Mx^t_{ijk} \quad \forall i \in V-\{0\}, j \in V-\{0\}, k \in K, t \in T \tag{48}$$

$$\sum_{j\in Ch(i)}\sum_{k\in K} z^{tc}_{ijk} - \sum_{j\in P(i)}\sum_{k\in K} z^{tc}_{jik} = \bar{d}^{tc}_i \quad \forall i \in (V-S)-\{0\}, t \in T, c \in C \tag{49}$$

$$\bar{d}^{tc}_i \leq m^{tc}_i \quad \forall i \in V-\{0\}, t \in T, c \in C \tag{50}$$

$$\sum_{k\in K} d^{tc}_{ik} \leq \bar{d}^{tc}_i \quad \forall i \in V-\{0\}, t \in T, c \in C \tag{51}$$

$$\sum_{c\in C} z^{tc}_{ijk} = 0 \quad \forall i \in (V-S)-\{0\}, j \in S, k \in K, t \in T \tag{52}$$

$$\sum_{c\in C} d^{tc}_{ik} \geq r^t_{ik} \quad \forall i \in V-\{0\}, k \in K, t \in T \tag{53}$$

$$\sum_{c\in C} d^{tc}_{ik} \leq Mr^t_{ik} \quad \forall i \in V-\{0\}, k \in K, t \in T \tag{54}$$

$$\sum_{i\in S-\{0\}} r^t_{ik} = 0 \quad \forall i \in V-\{0\}, k \in K, t \in T, c \in C \tag{55}$$

$$\sum_{i\in V-\{0\}}\sum_{Q\in\{1,2\ldots t\}}\sum_{c\in C} z^{Qc}_{ijk} \leq |Q||L|q_l \quad \forall t \in T \tag{56}$$

$$\sum_{i\in V-\{0\}}\sum_{Q\in\{1,2\ldots t\}} z^{Qc}_{ijk} \leq \sum_{Q\in\{1,2\ldots t\}} B^{Qc} \quad \forall t \in T, c \in C \tag{57}$$

The secondary vehicle routing problem is a special case of TCVRP. The underlying network is composed of multiple trees with depots at the root nodes and split deliveries allowed. We define a new problem as the capacitated vehicle routing problem on a tree with split deliveries (TCVRP-SD) in which vehicles are routed on a tree with split deliveries allowed. It is a special case of SDVRP in which the underlying network is a tree. Solving the secondary vehicle routing problem is equivalent to solving a TCVRP-SD for each tree in the secondary network for each time period. Theorem 1 provides a closed-form solution for the optimal objective of TCVRP-SD and a unique property of the structure of the routes. We use this to solve the secondary vehicle routing model. For stating Theorem 1, we have to define some notation. Consider a rooted tree T' with V' and E' as the set of nodes and arcs respectively. $P(i)$ is the parent of node i, $N(P(i), i)$ is the the number of vehicles that will traverse the arc $(P(i), i)$ each incurring a cost $Cost(P(i), i)$. Consider a subtree $S(i)$ rooted at i and $d(S(i))$ is the amount of the commodities delivered to the nodes within $S(i)$. We also define a Parent–Children Pair $(i, Ch(i))$. For instance, in Figure 4, there are 4 Parent–Children pair $(0,\{1\}), (1,\{2,5\}), (2,\{3,4\})$' and $(5,\{6,7\})$. $S(2)$ is the subtree rooted at node 2 and the $d(S(2))$ is equivalent to the sum of the deliveries made at node 2, 3, and 4. Since split deliveries are allowed in the model, the number of vehicles traveling the arc $(P(2), 2) = (1,\ 2)$ is equal to $\lceil d(S(2)/q_s\rceil$, where q_s is the capacity of each vehicle. It must be noted that minimum costs of an SDVRP are lower than that of VRP on the same network (Archetti & Speranza, 2012). Also, in an SDVRP there is a feasible solution with minimum number of vehicles to complete the deliveries. However the number of vehicles in a minimum cost solution might be more than the minimum number of vehicles possible. In TCVRP-SD, we show that the minimum cost route also utilizes the minimum number of vehicles (Archetti & Speranza, 2012). Now, we are ready to state Theorems 1 and 2.

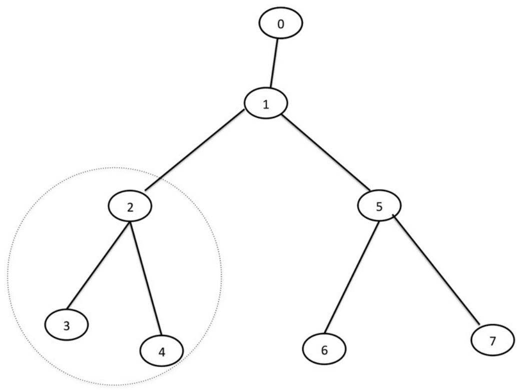

Figure 4. Example of a subtree of a tree.

Theorem 1. *Given a TCVRP-SD problem, the optimal objective of minimum cost is given by* $2\sum_{i\in V}\lceil d(S(i))/q_s\rceil Cost(P(i), i)$ *for each node i where* $d(s(i)$ *is a demand of a subtree rooted at i,* q_s *is the capacity of a vehicle and* $\lceil d(S(i))/q_s\rceil$ *is the minimum number of vehicle required to deliver in subtree S(i).*

Theorem 2. *In a TCVRP-SD the minimum cost solution also utilizes the minimum number of vehicles.*

The implication of Theorem 1 is that the values of the delivery variables $\bar{d}_i^{tc}$ can be used to calculate the number of vehicles traveling each arc $(P(i), i), N(P(i), i)$. This, in turn, determines the optimal objective $2 * \sum_{i \in V} N(P(i), i) * Cost(P(i), i)$. Also, the arcs traveled by vehicles within parent–children pairs is determined by $N(P(i), i)$'s, using which the route of a vehicle can be traced. We developed an algorithm, Algorithm 2 that uses these implications of Theorem 1. The description and procedure of the algorithm is available in the subsection $A\lambda\gamma o\rho\iota\tau\eta\mu$ 2 of the Appendix. In Steps 1–3, as given in the pseudo code of the algorithm, for each time period, the value of $N(P(i), i)$ is calculated using the parameters $\bar{d}_i^{tc}$ In Step 5, the optimal objective is calculated. Steps 6–31 are the vehicle routing steps. We determine the arcs travelled by the vehicles within parent–children pairs $(i, Ch(i))$ starting from the leaves of the tree going all the way upto the depot (root of the tree). For a leaf node within a parent–children pair, Steps 10–20 are used to determine the values of $\bar{x}_{ijk}^t$ and $\bar{d}_i^{tc}$. If the node is not a leaf node Steps 21–31 determine the values of $\bar{x}_{ijk}^t$ and $\bar{d}_i^{tc}$.

The implication of Theorem 2 is significant for real world applications like disasters response. The reason we observe split deliveries in real disaster scenarios is justified and warranted by Theorem 2. If the resources like vehicles for transportation and transportation network is sparse and almost-tree due to damage, split delivery of relief is the most cost/resource efficient strategy as it minimizes the vehicles in the minimizing cost solution.

5.3. Capacitated primary vehicle routing model

The capacitated primary vehicle routing models solves the problem of finding the minimum cost tours for the primary vehicles given the optimal deliveries from the delivery model and the optimal tours of the secondary vehicles from the secondary vehicle routing model. The decision variables are $y_{ijl}^t, v_{ijl}^{tc}, S_i^{tc}$, and D_i^{tc}. The objective is minimizing the cost of transportation of the primary vehicles. The constraints are primary vehicle routing constraints, loading constraints, and storage constraints from the model with tree formulation. $\overline{d_i^{tc}}$ is the optimal value of d_i^{tc} from the delivery model and $\overline{z_{ijk}^{tc}}$ is the optimal value of z_{ijk}^{tc} from the secondary vehicle routing model. They are input as parameters in the model. We found that capacitated primary vehicle routing models can be solved using CPLEX as computational times are reasonable for realistic datasets. The model is as follows:

Minimize: $\sum_{i \in S} \sum_{j \in S, j \neq i} \sum_{l \in L} \sum_{t \in T} h_{ij} y_{ijl}^t$**st:**

$$\sum_{i \in S \cup \{0\}, i \neq j} y_{ijl}^t = \sum_{i \in S \cup \{0\}, i \neq j} y_{jil}^t \qquad \forall j \in S \cup \{0\}, l \in L, t \in T \tag{58}$$

$$\sum_{j \in S} y_{0jl}^t = 1 \qquad \forall, l \in L, t \in T \tag{59}$$

$$\sum_{i \in W} \sum_{j \in W, j \neq i} y_{ijl}^t \leq |W| - 1 \qquad \forall W \subseteq S, |W| \geq 2, l \in L, t \in T \tag{60}$$

$$\sum_{j \in S} \sum_{c \in C} v_{0jl}^{tc} \leq q_l \qquad \forall, l \in L, t \in T \tag{61}$$

$$\sum_{i \in S \cup \{0\}} \sum_{j \in S, j \neq i} y_{ijl}^t (h_{ij} + g_i) \leq r_h / s_h \qquad \forall l \in L, t \in T \tag{62}$$

$$\sum_{c \in C} v_{ijl}^{tc} \leq M y_{ijl}^t \qquad \forall i \in S \cup \{0\}, j \in S \cup \{0\}, l \in L, t \in T \tag{63}$$

$$\sum_{c \in C} v_{ijl}^{tc} = 0 \qquad \forall i \in (V - S) - \{0\}, j \in S, k \in K, t \in T \tag{64}$$

$$S_i^{tc} = S_i^{t-1c} + D_i^{tc} - \sum_{j \in Ch(i),} \sum_{k \in K} \bar{z}_{ijk}^{tc} - \bar{d}_i^{tc} \qquad \forall i \in S, t \in T - \{0\}, c \in C \tag{65}$$

$$S_i^{tc} = D_i^{tc} - \sum_{j \in Ch(i)} \sum_{l \in L} \overline{z_{ijk}^{tc}} - \overline{d_i^{tc}} \qquad \forall i \in S, t = 0, c \in C \tag{66}$$

5.4. Computational complexity

The original model is NP-hard as it is an extension of CVRP. The reformulated model reduces the computational complexity of the problem by remodeling the secondary vehicle routing constraints that exploit the tree structure of the network. However, the reformulated model also does primary vehicle routing on a regular network, which is still an NP-hard problem.

As stated earlier, the decomposition approach solves three problems, the delivery problem, a secondary vehicle routing problem, and a primary vehicle routing problem. The delivery problem is an extension of the Multiple Bounded Knapsack Problem, for which we developed a polynomial time algorithm with a worst-case time complexity of $\mathcal{O}(|T|DN)$, because the algorithm has $|T|$ steps and each step has $\mathcal{O}(D * N)$ elementary calculations. The secondary vehicle routing model is a TCVRP-SD, a special case of a vehicle routing, for which we developed a polynomial time algorithm with a worst-case time complexity $\mathcal{O}(|T||K|N^2)$, because the algorithm has $|T|$ steps and each step has $\mathcal{O}(|K|N^2)$ elementary calculations. The primary vehicle routing model

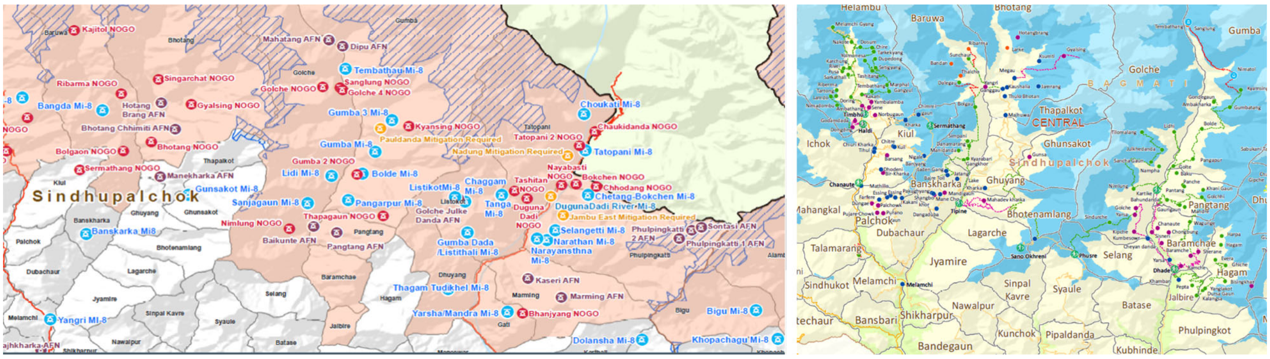

Figure 5. (a) Map of Identified Helicopter Landing Zones in Nepal (b) Map of Mountain Trail Network in Sindhupalchowk district, Nepal (World Food Program-2, 2016).

of the decomposition is a CVRP and is NP-hard. Since the primary network is small in size, the primary vehicle routing problem can be solved efficiently even though it is NP-hard, implying that the decomposition approach works well in practice.

6. Case study

In our case study, we modeled the relief supply operations that were conducted by the Logistics Cluster of the United Nations World Food Program for the very remote and mountainous regions of Nepal. They called this operation the Remote Access Operation (RAO) (World Food Program, 2016). In RAO, first, the relief supplies were transported from the district headquarters to identified base camps/helicopter landing zones (HLZ) via helicopters. Next, they were transported to the villages at higher altitudes and remote locations via porters and mules (other animal packs) on mountain trails. The helicopter network was the primary network while the porter network was the secondary network. Figure 5(a,b) show helicopter landing zones (primary network) and the mountain trails (secondary network). As evident, the secondary network of mountain trails has multiple almost-trees with very less cycles. The following section described the dataset in detail section includes the case study that we preformed.

The case study was preformed on two datasets: (1) a toy dataset created with miniature versions of the Nepal networks for comparing the computational performance of CPLEX and our solution method, and (2) real data from Sindhupachowk district of Nepal (part of RAO, Nepal Earthquake 2015) for testing our solution method's performance on realistic cases (CPLEX was found to be computationally impractical for these cases). CPLEX was used to solve the original and the reformulated model on the toy dataset while the decomposition method was tested using both the toy and real datasets. For solving the reformulated model, cycles were removed from the secondary networks using Krushkal's minimum spanning tree algorithm. The following sections contain detailed descriptions of the datasets.

6.1. Sindhupalchowk dataset

The dataset contains one hundred sixteen villages as part of the porter network, seven helicopter landing zones/trail heads (in seven of the above villages), and one helicopter depot in the Sindhupalchok district of Nepal. Table 1 is a sample description of our dataset. Figure 5(b) is the map of Sindhupalchok district that contains our dataset with the villages, depot at Chautara, and trail heads/HLZ's at Chanaute, Haldi, Timbhu, Sermathang, Tipine, Dhade, and Kartike. The cost matrix for the porter network which consists of the time taken by the porters to travel between villages was constructed using the Naismith's rule. The original Naismith's rule from the year 1892 says, that men should allow 1 h per 3 miles on the map and an additional 1 h per 2000 feet of ascent. Naismith's 1 h/3 min + 1 h/2000 ft can be converted to other conventions. We use the convention 12 min/1 km + 10 min/100 m.

Table 2 has the population of the municipalities of the villages from the population census of Nepal in 2011. Since, we did not have village-level population data, we distributed the population of each municipality into its villages such that the population of the villages followed a normal distribution and summed up to the total population of the municipality. Using this population, we estimated the demand for food and metal roofs in each period at each village. The average number of people in a Nepali family is 4.62 (Government of Nepal, 2016). We assumed a requirement of one 26 gauge metal roof per family which weighs approximately 0.45 Kg. The daily food requirement of a person is nearly 600 g providing 2100 Kcal of energy, 65 g and 40 g of proteins and fat, respectively (in the form of cereals, pulses, oils, salt, sugar etc.) (WHO & UNICEF, 2004). It must be noted that in the case of the RAO, food and shelter related items like tin roofs were given priority. Water was available in the villages via local hand pumps that use the ground water table. There are 1000 porters/animal packs and 8 Mi-8 helicopters as primary and secondary vehicles. The weight carrying capacity of a porter is 30 kg while that of a helicopter is 4 metric ton or 4000 Kg. The limit to travel for each porter is 4 days (96 h) which includes 4-h service and rest time at each of the nodes served. For a helicopter, the limit on travel in one period is 8 h which includes 0.5 h of service time at each HLZ. The average speed of a helicopter is assumed to be 100 km/h. The amount of each item available for distribution at the helicopter depot is 10,000 kg or 10 metric ton for each period. The operation was carried out for three time periods each equivalent to a period of 4 days. We vary the value of parameter α from 0.001 to 2 and show the difference in results.

Table 1. Sample of Villages and Municipalities Served in Sindhupalchowk district served in the model.

Node	Village	Municipality	Depot/Trail Head/Village
0	Chautara	Chautara	Depot
1	Chanaute	Palchok	Trail Head
2	Mathillo	Palchok	Village
3	Dungae	Palchok	Village
4	Farfere	Palchok	Village
5	Pal chowk	Palchok	Village
6	Essing	Palchok	Village
7	Kakani	Palchok	Village
8	Dhodeni	Kiul	Village
9	Bir-Kharka	Kiul	Village
10	Barsang	Kiul	Village

Table 2. Number of households and population of municipalities of Sindhupalchowk.

Municipalities	Households	Total population
Baramchae	705	3248
Baruwa	487	1831
Bhanskhara	588	2259
Bhotang	624	2582
Bhotenamlag	792	3551
Chautara	1618	5952
Ghunsakot	449	1902
Golche	731	3611
Gumba	674	3431
Hagam	818	3847
Helmabu	656	2564
Jalbire	611	2540
Palchok	489	1927
Pantang	487	2481

6.2. Toy dataset

The toy dataset contains nine villages as part of the porter network, two helicopter landing zones, two mountain trails, and one helicopter depot. There are 8 porters/animal packs and two helicopters for transpiration. The distances in the cost matrices and demands in each village is also reduced to smaller numbers. Rest of the parameters are the same as in the Sindhupachowk dataset.

6.3. Results and discussion

In the experiment with the toy dataset, we had the following three schemes:

1. Solve the Original model using CPLEX.
2. Solve the Reformulated model using CPLEX.
3. Solve the Reformulated model using the Decomposition Method.

We followed the above schemes for different values of α. Table 3 compares the results of these three schemes. It contains the values of the objective function, the primary objective, Obj 1 $= \sum_{i\in V}\sum_{t\in T}\sum_{c\in C} d_i^{tc}$, the secondary objective, Obj 2 $= \sum_{i\in V}\sum_{j\in V, i\neq j}\sum_{k\in K}\sum_{t\in T} a_{ij}x_{ijk}^t + \sum_{i\in S}\sum_{j\in S, i\neq j}\sum_{l\in L}\sum_{t\in T} h_{ij}y_{ijl}^t$, and the computational time for the three schemes. We now use the results from Table 3 to (a) find the appropriate range of α, and (b)

Table 3. Number of households and population of municipalities of Sindhupalchowk.

Model	Solution method	α	Objective	Obj 1	Obj 2	Comp time (s)
Original	CPLEX	0.0001	479.9006	480	994	3.09
Reformulated	CPLEX	0.0001	479.8923	480	1077	1.24
Reformulated	Decomposition	0.0001	479.8868	480	1132	0.32
Original	CPLEX	0.01	471.02	480	898	3143.14
Reformulated	CPLEX	0.01	469.32	480	1068	102.1
Reformulated	Decomposition	0.01	468.68	480	1132	0.26
Original	CPLEX	0.1	390.2	480	898	4013.05
Reformulated	CPLEX	0.1	373.2	480	1068	136.17
Reformulated	Decomposition	0.1	366.8	480	1132	0.27
Original	CPLEX	0.5	31	480	898	3468.46
Reformulated	CPLEX	0.5	−54	480	1068	28.34
Reformulated	Decomposition	0.5	−86	480	1132	0.27
Original	CPLEX	1	−418	471	898	4452.18
Reformulated	CPLEX	1	−525	471	996	26.89
Reformulated	Decomposition	1	−661	471	1132	0.13
Original	CPLEX	1.5	−866	475	894	3965.33
Reformulated	CPLEX	1.5	−1011	399	940	8.22
Reformulated	Decomposition	1.5	−1218	480	1132	0.13
Original	CPLEX	2	−1294	410	852	202.78
Reformulated	CPLEX	2	−1505	329	917	3.97
Reformulated	Decomposition	2	−1784	480	1132	0.3

Table 4. Variation in values of objectives with variation in parameters.

Parameter	Original value	Changed value	Obj 1	Obj 2	Objective	Comp time (s)
None	N/A	N/A	90,000	15,312	28,468.8	3376.12
q_s	30	20	60,000	13,326	18,667.4	3469.71
q_l	4000	2000	48,000	14,971	14,502.9	3793.33
L	8	4	48,000	14,203	14,579.7	3642.56
K	1000	500	45,000	13,117	13,688.3	3725.13
B^{tc}	50,000	9000	81,000	14,159	25,584.1	871.45
s_h	100	50	90,000	22,583	18,708.5	27,741.7

compare the computational performance of the three schemes.

A smaller value of α leads to a smaller contribution of the secondary objective. Therefore, it is expected that for each dataset, there exists a value of $\alpha = \alpha' \in (0, \infty)$ below which the contribution of primary objective in the objective function is more than the secondary objective. In our toy dataset, this was observed at $\alpha' < 0.5$. The primary objective, Obj 1, is at its peak value 480 for α ranging from 0.0001 to 0.5 (in both the original and reformulated model) as shown in Table 3. It decreases if α is further increased for both the models. The best range of α for which the primary objective is maximized depends on the dataset and its parameter values. In subsection $\Pi\rho oo\phi$ $o\phi$ $T\eta\varepsilon o\rho\varepsilon\mu$ 1 of the Appendix we establish the existence of α for different datasets and show how it is related to the values of the primary and secondary objective. Given this value, we must further find a range of value of α for which the secondary objective achieves a minimum given that the primary has achieved its optimal value. This can be found by varying the $\alpha \in (0, \alpha')$. In our case study, the secondary objective achieves its minimum value given an optimal primary objective (for both the models) at $\alpha \in (0.01, 0.5)$ as shown in Table 3. Hence, we conclude that for the toy dataset the desirable range is $\alpha \in (0.01, 0.5)$. Another interesting observation is that the decomposition-based solution method always maximizes the priority objective to 480 and is unaffected by the value of α. This is because the solution method breaks down the model into delivery and routing models. It always solves the delivery model prior to the routing model, maximizing the primary objective before minimizing the secondary objective.

We observe that the value of the optimal objective for both the original and the reformulated model are nearly equal for $\alpha \in (0.0001, 0.5)$. This can be explained by the fact that the secondary network in the original model is a 3-almost tree graph with very small number of cycles. Hence, in the reformulated model, when we apply the Krushkal's Minimum Spanning tree to the original network (to obtain a a tree), at most 3 arcs are eliminated from the biconnected components of the graphs (that contain cycles). Consequently, the minimum spanning tree of the network very closely approximates the original network. Hence, the change in cost of routing secondary vehicles as reflected in the secondary objective is very small. Since the secondary objective contributes in the overall objective by the factor of α, the change in the overall objective function is negligible. It can also be observed that the value of the objective from the decomposition-based solution method approximates the value for both the original and the reformulated model very well. As expected, the value of primary objective for both the models solved using CPLEX is equal (for all α's). This is because the delivery variables are unaffected by reformulation of constraints and the structure of the underlying network which is a tree in the reformulated model.

It is observed that for most values of α on the toy dataset, the computational time for CPLEX on the original model is the highest. We observe that (for α ranging from 0.1 to 1) there is a 10-fold reduction in compuational times when CPLEX is used to solve the reformulated model instead of the original model. The reduction in computational time in the reformulated model is due to the secondary vehicle routing constraints that assume the underlying network is a tree. The original model is computationally inefficient due to the inefficient representation of the vehicle routing constraints in the secondary network. The reformulated model, on the other hand, is computationally efficient as the vehicle routing constraints exploit the tree network structure. The decomposition-based solution method shows exceptional computational efficiency as compared to CPLEX, with a 100-fold reduction in computational time, solving each case in the toy dataset in less than a second. This is because the complexity of the individual models in the decomposition method is much less than the original and reformulated methods. The delivery and secondary vehicle routing models have polynomial time solutions using Algorithms 1 and 2, respectively, as detailed in Section 5.4. Even though the primary vehicle routing model is NP-hard, the size of the primary network is small and consequently the computational time to solve it using CPLEX is small.

We observed that the usage of CPLEX became computationally burdensome as we increased the toy dataset size by adding 10 more demands nodes. Hence, for solving the real world dataset from

Sindhupalchow district we used the decomposition method. For the Sindhupachowk dataset we fixed α at 0.1. The method solved the model in a little more than one hour of computational effort. It yielded a primary objective equal to 30,000, secondary objective equal to 15,312 and an overall objective equal to 22,344. To test the sensitivity of the model to different parameters, we varied the parameters and report results in Table 4. The primary objective is bounded by total load carrying capacity, which is equal to $|T||K|q_s, |T||L|q_l$ and $\sum_{t\in T}\sum_{c\in C}B^{tc}$. Therefore, changes in the parameters q_s, q_l, $|L|$ $|K|$ and B^{tc} are reflected in the objectives. For instance, on changing the load carrying capacity of a porter, q_s, from 30 Kg to 20 Kg, the primary objective also reduced from 9000 to 6000. Similar observations can be made for other parameters.

The primary objective worsened as we reduced the values of the parameters while the secondary objective improved. However, the secondary objective is less sensitive to change in parameters than primary objective (magnitude of change is less). A direct implication of these results is that in a scenario like the Nepal earthquake, parameters like number of porters $|K|$, number of helicopters $|L|$, and amount of total supply at the depot $\sum_{t\in T}\sum_{c\in C}B^{tc}$ that can be controlled should be increased without worrying too much about the cost of transportation. Not only is the cost a secondary objective but also it is less sensitive to parameter values.

The major observations from the case study are:

- **Range of α:** The appropriate range is $\alpha \in (0.1, 0.5)$.
- **Computational time savings:** There is 10-fold reduction in computational time using CPLEX to solve the reformulated as compared to the original model (on the toy dataset). There is a further 100-fold reduction in computational time using the decomposition method.
- **Sensitivity analysis:** The secondary objective of cost of transportation is less sensitive to values of the following two parameters, number of porters $|K|$ and number of helicopters $|L|$.

In conclusion, we observed that the last mile delivery in disaster relief transportation often happens on almost-tree and utilises split deliveries. Motivated by this, we developed a two stage multi-period vehicle routing model with split deliveries, in which the network for last mile deliveries is a tree or an almost-tree. We developed a reformulation of the model which uses the properties of routing on trees to improve computational efficiency by an order of magnitude. We also developed a decomposition of the model that further improves computational performance by another order of magnitude. To solve the routing problem on a tree network in the decomposed model, we defined the Capacitated Vehicle Routing Problem on a Tree with Split Deliveries (TCVRP-SD) and developed an exact algorithm for its solution. We proved certain mathematical properties of the TCVRP-SD and found a closed-form solution to the optimal objective function value of TCVRP-SD. Our theoretical findings show that using split delivery in a sparse and almost-tree damaged network is the most cost/resource efficient strategy as the minimum cost solution also minimises the number of vehicles.

To validate the model and our solution method, we presented a case study using a dataset from relief delivery operations in the 2015 Nepal earthquake. We showed that our method improved the computational efficiency over tradition methods like CPLEX by 20-folds on small datasets. CPLEX became intractable beyond 10–12 nodes. On real-sized data of the case study our method could solve the problem within 1 h. Using sensitivity analysis we found that the secondary objective of minimizing cost was more sensitive to the parameters of number of secondary vehicles, the number of primary vehicles and the amount supplied than the primary objective of minimizing unmet demand.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Aleman, R. E., & Hill, R. R. (2010). A tabu search with vocabulary building approach for the vehicle routing problem with split demands. *International Journal of Metaheuristics*, *1*(1), 55–80.

Aleman, R. E., Zhang, X., & Hill, R. R. (2010). An adaptive memory algorithm for the split delivery vehicle routing problem. *Journal of Heuristics*, *16*, 441–473.

Archetti, C., Bianchessi, N., & Speranza, M. G. (2011). A column generation approach for the split delivery vehicle routing problem. *Networks*, *58*, 241–254.

Archetti, C., Bianchessi, N., & Speranza, M. G. (2014). Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*, *238*, 685–698.

Archetti, C., Mansini, R., & Speranza, M. G. (2005). Complexity and reducibility of the skip delivery problem. *Transportation Science*, *39*, 182–187.

Archetti, C., Savelsbergh, M. W. P., & Speranza, M. G. (2006). Worst-case analysis for split delivery vehicle routing problems. *Transportation Science*, *40*, 226–234.

Archetti, C., & Speranza, M. G. (2008). The split delivery vehicle routing problem: A survey. In B. Golden, S. Raghavan & E. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges* (pp. 103–122). Boston, MA: Springer.

Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, *19*(1–2), 3–22.

Archetti, C., Speranza, M. G., & Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, *40*(1), 64–73.

Archetti, C., Speranza, M. G., & Savelsbergh, M. W. (2008). An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, *42*(1), 22–31.

Balcik, B., & Beamon, B. M. (2008). Facility location in humanitarian relief. *International Journal of Logistics Research and Applications*, *11*, 101–121.

Basnet, C., Foulds, L. R., & Wilson, J. M. (1999). Heuristics for vehicle routing on tree-like networks. *Journal of the Operational Research Society*, *50*, 627–635.

Belenguer, J.-M., Martinez, M., & Mota, E. (2000). A lower bound for the split delivery vehicle routing problem. *Operations Research*, *48*, 801–810.

Berbotto, L., García, S., & Nogales, F. J. (2014). A randomized granular tabu search heuristic for the split delivery vehicle routing problem. *Annals of Operations Research*, *222*, 153–173.

Berkoune, D., Renaud, J., Rekik, M., & Ruiz, A. (2012). Transportation in disaster response operations. *Socio-Economic Planning Sciences*, *46*(1), 23–32.

Boudia, M., Prins, C., & Reghioui, M. (2007). An effective memetic algorithm with population management for the split delivery vehicle routing problem. In T. Bartz-Beielstein et al. (Eds.), *International Workshop on Hybrid Metaheuristics* (pp. 16–30). Berlin, Heidelberg: Springer.

Chandran, B., & Raghavan, S. (2008). Modeling and solving the capacitated vehicle routing problem on trees. In B. Golden, S. Raghavan & E. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges* (pp. 239–261). Boston, MA: Springer.

Chen, P., Golden, B., Wang, X., & Wasil, E. (2017). A novel approach to solve the split delivery vehicle routing problem. *International Transactions in Operational Research*, *24*(1–2), 27–41

Chen, S., Golden, B., & Wasil, E. (2007). The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, *49*, 318–329.

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S. G., & Bian, L. (2017). Drones for disaster response and relief operations: A continuous approximation model. *International Journal of Production Economics*, *188*, 167–184.

Derigs, U., Li, B., & Vogel, U. (2010). Local search-based metaheuristics for the split delivery vehicle routing problem. *Journal of the Operational Research Society*, *61*, 1356–1364.

Detti, P. (2009). A polynomial algorithm for the multiple knapsack problem with divisible item sizes. *Information Processing Letters*, *109*, 582–584.

Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics*, *50*, 239–254.

Dror, M., & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, *23*, 141–145.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics*, *37*, 383–402.

Ferrer, J. M., Martín-Campo, F. J., Ortuño, M. T., Pedraza-Martínez, A. J., Tirado, G., & Vitoriano, B. (2018). Multi-criteria optimization for last mile distribution of disaster relief aid: Test cases and applications. *European Journal of Operational Research*, *269*, 501–515.

Gibbons, D. E., & Samaddar, S. (2009). Designing referral network structures and decision rules to streamline provision of urgent health and human services. *Decision Sciences*, *40*, 351–371.

Glover, F. (1968). Surrogate constraints. *Operations Research*, *16*, 741–749.

Government of Nepal. (2016). Nepal Household Survey. Retrieved from https://reliefweb.int/report/nepal/nepal-annual-household-survey-201516

Gulczynski, D. J., Golden, B., & Wasil, E. (2008). Recent developments in modeling and solving the split delivery vehicle routing problem. In *State-of-the-art decision-making tools in the information-intensive age* (pp. 170–180). INFORMS.

Hamaguchi, S., & Katoh, N. (1998). A capacitated vehicle routing problem on a tree. In K. Y. Chwa & O. H. Ibarra (Eds.), *International Symposium on Algorithms and Computation* (pp. 399–407). Berlin, Heidelberg: Springer.

International Red Cross Society. (2016). World disaster report. Retrieved from https://www.ifrc.org/Global/Documents/Secretariat/201610/WDR\%202016-FINAL_web.pdf

Jin, M., Liu, K., & Bowden, R. O. (2007). A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. *International Journal of Production Economics*, *105*(1), 228–242.

Jin, M., Liu, K., & Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, *36*, 265–270.

Khmelev, A., & Kochetov, Y. (2015). A hybrid local search for the split delivery vehicle routing problem. *International Journal of Artificial Intelligence*, *13*(1), 147–164.

Kovács, G., & Spens, K. M. (2007). Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management*, *37*, 99–114.

Kumar, R., Unnikrishnan, A., & Waller, S. T. (2011). Capacitated-vehicle routing problem with backhauls on trees: Model, properties, formulation, and algorithm. *Transportation Research Record: Journal of the Transportation Research Board*, *2263*(1), 92–102.

Kumar, R., Unnikrishnan, A., & Waller, S. T. (2012). Heuristic scheme for heterogeneous vehicle routing problem on trees based on generalized assignment and bin-packing upper bounds. *Transportation Research Record: Journal of the Transportation Research Board*, *2283*(1), 1–11.

Labbé, M., Laporte, G., & Mercure, H. (1991). Capacitated vehicle routing on trees. *Operations Research*, *39*, 616–622.

Luis, E., Dolinskaya, I. S., & Smilowitz, K. R. (2012). Disaster relief routing: Integrating research and practice. *Socio-Economic Planning Sciences*, *46*(1), 88–97.

Min, J., Jin, C., & Lu, L. (2019). Maximum-minimum distance clustering method for split-delivery vehicle-routing problem: Case studies and performance comparisons. *Advances in Production Engineering & Management*, *14*(1), 125–135.

Moreno, L., De AragãO, M. P., & Uchoa, E. (2010). Improved lower bounds for the split delivery vehicle routing problem. *Operations Research Letters*, *38*, 302–306.

Mota, E., Campos, V., & Corberán, Á. (2007). A new metaheuristic for the vehicle routing problem with split demands. In C. Cotta & J. van Hemert (Eds.), *European Conference on Evolutionary Computation in*

Combinatorial Optimization (pp. 121–129). Berlin, Heidelberg: Springer.
Muslea, I. (1997). The very offline k-vehicle routing problem in trees. XVII International Conference of the Chilean Computer Science Society, 1997 Proceedings (pp. 155–163). Valparaiso, Chile: IEEE.
Noyan, N., Balcik, B., & Atakan, S. (2016). A stochastic optimization model for designing last mile relief networks. *Transportation Science*, *50*, 1092–1113.
Noyan, N., & Kahvecioğlu, G. (2018). Stochastic last mile relief network design with resource reallocation. *OR Spectrum*, *40*(1), 187–231.
Ozdamar, L. (2011). Planning helicopter logistics in disaster relief. *OR Spectrum*, *33*, 655–672.
Rabta, B., Wankmüller, C., & Reiner, G. (2018). A drone fleet model for last-mile distribution in disaster relief operations. *International Journal of Disaster Risk Reduction*, *28*, 107–112.
Silva, M. M., Subramanian, A., & Ochi, L. S. (2015). An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research*, *53*, 234–249.
Tzeng, G.-H., Cheng, H.-J., & Huang, T. D. (2007). Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E: Logistics and Transportation Review*, *43*, 673–686.
Vitoriano, B., Ortuño, M. T., Tirado, G., & Montero, J. (2011). A multi-criteria optimization model for humanitarian aid distribution. *Journal of Global Optimization*, *51*, 189–208.
Wang, H., Du, L., & Ma, S. (2014). Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake. *Transportation Research Part E: Logistics and Transportation Review*, *69*, 160–179.
WHO & UNICEF. (2004). Food and nutrition needs in emergencies. Retrieved from https://apps.who.int/iris/bitstream/handle/10665/68660/a83743.pdf
World Food Program. (2016). Nepal lessons learned report. Retrieved from https://logcluster.org/sites/default/files/logistics_cluster_nepal_lessons_learned_report_160121.pdf
World Food Program-2. (2016). Nepal documents. Retrieved from https://logcluster.org/countries/NPL
World Food Program-3. (2016). Remote acess operations. Retrieved from https://logcluster.org/document/remote-access-operation-snapshot
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.

APPENDIX

Algorithm 1

We need to define some notation for describing the algorithm. For each period t, there is a maximum delivery P_t given by $\sum_{i \in V} \sum_{c \in C} d_i^{tc}$. In each period t, there is an unsatisfied demand at each demand node $(m_i^{tc} - d_i^{tc})$. Let the mean of this unsatisfied demand for all demand nodes in each period be denoted by M. Let the upper bound and lower bound on M (ensured by equity constraints (40)−(41) be U' and L' respectively. The algorithm returns $\overline{d_i^{tc}}$, which is input into the capacitated appondary vehicle routing problem as a parameter.

In step 1, as given in the pseudo code of the algorithm, $P_t = \sum_{i \in V} \sum_{c \in C} d_i^{tc}$ is calculated such that the bounds on P due to constraints (38)−(41) are satisfied. In steps 1–9, the algorithm distributes the calculated P into individual $\bar{d}_i^{tc}$'s, such that the constraint (40) is satisfied and constraint (41) is tight. In Steps 11–16, the remaining delivery R is distributed to the nodes such that the value of $N(P(i), i) = \lceil \sum_c \bar{d}_i^{tc} / q_s \rceil$ (described in the next section) does not change. If remaining delivery R is still positive, in steps 18–20, R is distributed to the nodes in ascending order of their distance from the depot (ensuring the upper and lower bound on unsatisfied demand imposed by constraints (40)−(41) at each node). In Step 21, the objective is calculated.

Algorithm 1. Algorithm to solve the delivery model

1: **for** $t \in T$ **do** ▷ for each of the time period
2: $P_t \leftarrow Min(\ |K|q_s, | \ L|q_l, \sum_{i \in V-} \ \{0\} \sum_{c \in C} \ m_i^{tc},$
$\sum_{Q \in \{1,2 \ldots t\}} \sum_{c \in C} B^{Qc} - \sum_{Q \in \{1,2 \ldots t-1\}} \sum_{c \in C} B \ Qc)$
3: $M \leftarrow (1/(N-1))(\sum_{i \in V-\{0\}} \sum_{t \in T} \sum_{c \in C} m_i^{tc}) - P_t$
4: $U' \leftarrow U_p * M$
5: $L' \leftarrow L_o * M$
6: **if** $]P_t[\geq L' or [P_t] \leq U'$ **then**
7: **return** *Model Infeasible*
8: **else**
9: $\bar{d}_i^{tc} \leftarrow m_i^{tc} -]U'[$
10: $R \leftarrow P_t - \sum_{i \in V} \sum_{c \in C} \bar{d}_i^{tc}$
11: $i = 1$
12: **while** $R \neq 0$ **and** $i \in V - \{0\}$ **do**
13: **if** $(] \sum_c \bar{d}_i^{tc} + 1[)/q_s =] \sum_c \bar{d}_i^{tc} [/q_s$ **then**
14: $\bar{d}_i^{tc} = \bar{d}_i^{tc} + 1$
15: $R = R - 1$
16: $i = i + 1$
17: **while** $R \neq 0$ **do**
18: **for** $i \in V_a$ **do**
19: $\bar{d}_i^{tc} = \bar{d}_i^{tc} + 1$
20: $R = R - 1$
21: **Objective** $= \sum_{t \in T} \sum_{i \in V} \sum_{c \in C} d_i^{tc} = \sum_{t \in T} P_t$

Algorithm 2

In Steps 1–3 of the algorithm, as given in the pseudo code of the algorithm, for each time period, the value of $N(P(i), i)$ is calculated using the parameters $\bar{d}_i^{tc}$. In Step 5, the optimal objective is calculated. Steps 6–31 are the vehicle routing steps. We determine the arcs travelled by the vehicles within parent children pairs $(i, Ch(i))$ starting from the leaves of the tree going all the way upto the depot (root of the tree). For a leaf node within a parent–children pair, Steps 10–20 are used to determine the values of $\bar{x}_{ijk}^t$ and $\bar{d}_i^{tc}$. If the node is not a leaf node Steps 21–31 determine the values of $\bar{x}_{ijk}^t$ and $\overline{d_i^{tc}}$.

Algorithm 2. Algorithm to solve the Capacitated Secondary Vehicle Routing Problem

1: **for** each time period $t \in T$ **do**
2: **for** each node $i \in V - \{0\}$ **do**
3: $N(P(i), i) =]d(S(i))/q_s[$ ▷ Using $\bar{d}_i^{tc}$'s

4: **Initialize** $V_i = \{0\}$ ▷ Empty set for vehicles at each node j
5: **Objective**$=2 * \sum_{i\in V} N(P(i), i) * Cost(P(i), i)$
6: **for** each tree T' in the secondary network **do** ▷ vehicle routing steps begin here
7: **for** each parent–children pair $(i, Ch(i))$ **do** ▷ from leaf nodes going up to the depot
8: **for** each child $j \in Ch(i)$ **do** ▷ each child in the parent–children pair
9: **if** j is a leaf node **then**
10: **Append** $N(P(j), j)$ vehicles to the set V_j from vehicle set K
11: **for** each vehicle k at the node j **do**
12: $q_k = q_s$
13: **for** each commodity c required at the node j **do**
14: $d_j^c = \bar{d}_j^{tc}$
15: $\bar{d}_{j,k}^{tc} = min(d_j^c, q_k)$
16: $q_k = q_k - \bar{d}_{j,k}^{tc}$
17: $d_j^c = d_j^c - \bar{d}_{j,k}^{tc}$
18: **Append** vehicle k to vehicle set $V_P(j)$ of node $P(j)$
19: $x_{P(j),j,k}^t = 1$ ▷ arc traveled once while going down
20: $x_{j,P(j),k}^t = 1$ ▷ arc traveled once while going up
21: **else if** j is not a leaf node **then**
22: $k_d = N(P(j), j) - |V_j|$
23: **if** $k_d > 0$ **then**
24: **Transfer** load of vehicles in set V_j to $N(P(j), j)$
25: **Remove** extra vehicles from set V_j and **append** to the set K
26: **Repeat** Steps 10–20
27: **else if** $k_d = 0$ **then**
28: **Repeat** Steps 10–20
29: **else if** $k_d < 0$ **then**
30: **Remove** vehicles from set K and **append** to the set V_j
31: **Repeat** Steps 10–20
32: —**Calculate** $\bar{z}_{ijk}^{tc}$ using the values of $\bar{d}_{jk}^{tc}$'s

---

Theorem 1. *Given a TCVRP-SD problem, the optimal objective of minimum cost is given by $2\sum_{i\in V}\lceil d(S(i))/q_s\rceil Cost(P(i), i)$ for each node i where $d(s(i)$ is a demand of a subtree rooted at i, q_s is the capacity of a vehicle and $\lceil d(S(i))/q_s\rceil$ is the minimum number of vehicle required to deliver in subtree S(i).*

Proof of Theorem 1

Proof. Consider a node i in the tree. Let $N(P(i), i)$ be the number of vehicles that will travel the arc $(P(i), i)$ in a minimum cost solution. Given, the cost of traveling each arc and the fact that each arc is travelled twice, once while going down from the depot and once while coming up towards the depot, the minimum cost is equivalent to the solution is given by $2\sum_{i\in V} N(P(i), i) * Cost(P(i), i)$. Let Δi be the set of nodes in the subtree $S(i)$. Then, the total amount of delivery made in the subtree is given by $d(S(i)) = \sum_{r\in\Delta i} d_r$. Since split deliveries are permissible, the minimum number of vehicles that are required to meet the demands of $S(i)$ is $\lceil d(s(i))/q_s\rceil$ (least integer greater than $d(s(2))/q_s$). Since the costs are to be minimized and the total costs are given by $2 * \sum_{i\in V} N(P(i), i) * Cost(P(i), i)$, minimum number of vehicles that are required to serve $S(i)$ will travel the arc, $(P(i), i)$. This implies the $N(P(i), i) = \lceil d(s(i))/q_s\rceil$. For a leaf node i, $d(s(i)) = d_i$ d. Therefore, the minimum cost is $2\sum_{i\in V}\lceil d(S(i))/q_s\rceil Cost(P(i), i)$. This completes the proof.

Proof of Theorem 2

Theorem 2. *In a TCVRP-SD the minimum cost solution also utilizes the minimum number of vehicles.*

Proof. Using Theorem 1, the number of vehicles traveling an arc $(P(i), i)$ for a minimum cost solution of SDVRP is the minimum number of vehicles required to deliver in subtree $S(i)$ and is given by $N(P(i), i) = \lceil d(s(i))/q_s\rceil$. Therefore, the number of vehicles traveling from the depot to subtrees rooted at the depot is also the minimum vehicles required to serve the subtrees. Hence, the total number vehicles in minimum cost solution is also the minimum number of vehicles required to do a split delivery.

Appropriate range of values for α

Given our objective function $\sum_{i\in V}\sum_c d_i^{tc} - \sum_{i=1}^N\sum_{j=1}^N\sum_{k=1,i\neq j}^{K*H}\alpha a_{ij}x_{ijk}^t - \sum_{iS}\sum_{jS,j\neq i}\sum_{l=1}^L \alpha h_{ij}y_{ijl}^t$ there exists a range of value of parameter α for which the optimum value of objective is obtained when our priority objective term $\sum_{i\in V}\sum_c d_i^{tc}$ reaches its optimum value.

Proof. The objective function can be written in the formmax$(A - \alpha.B)$, where:

$$A = \sum_{i\in V}\sum_c d_i^{tc}$$

$$B = \sum_{i=1}^N\sum_{j=1}^N\sum_{k=1,i\neq j}^{K*H} a_{ij}x_{ijk}^t - \sum_{iS}\sum_{jS,j\neq i}\sum_{l=1}^L h_{ij}y_{ijl}^t$$

$$\alpha \in (-\infty, \infty)$$

Let Set of all feasible solution be S_f. Then, $S_f \in ((A_1, B_{A_i}), (A_2, B_{A_2}) \ldots (A_{\max}, B_{A_{\max}} \ldots, (A_n, B_{A_n}))$ where n is the number of feasible solutions A_i is the value of A for the ith feasible solution B_{A_i} is the set of values of B when $A = A_i$ $A_{\max}$ is the max value of A in the feasible solution set $B_{A_{\max}}$ is the set values of B when $A = A_{\max}$

Let there be a value of α for which $\max(A - \alpha.B) = A_{\max} - \alpha\max(B_{\underset{A}{\max}})$. For this α, $A_{\max} - \alpha\max(B_{A_{\max}})$

$> A_j - \alpha\max(B_{A_j})$ and

$$A_j - \alpha\max(B_{A_j}) > A_i - \alpha(B_{A_i}) i \in (1, 2, \ldots n) - j, \max$$

Thus, we have:

$$\alpha < \frac{A_{\max} - A_j}{\max(B_{A_{\max}}) - \max(B_{A_j})}$$