# A Column Generation Algorithm for Vehicle Scheduling and Routing Problems

**Tasnim Ibn Faiz[1], Chrysafis Vogiatzis[2] and Md. Noor-E-Alam[1*]**

[1]Department of Mechanical and Industrial Engineering,
Northeastern University
360 Huntington Ave, Boston, MA 02115, USA

[2]Department of Industrial and Systems Engineering
North Carolina A&T State University
1601 E. Market St, Greensboro, NC, 27411, USA

*Corresponding author email: mnalam@neu.edu

**ABSTRACT**

During natural or anthropogenic disasters, humanitarian organizations (HO) are faced with the time sensitive task of sending critical resources from multiple depots to the affected areas that can be scattered across a region. This responsibility includes the quick acquisition of vehicles from the local market and the preparation of pickup and delivery schedules and vehicle routes. During crises, the supply of vehicle is often limited, their acquisition cost is steep, and acquired vehicles are restricted by rental periods. Moreover, the affected areas suffer from a dire need for aid materials and they must be reached within due time. Therefore, it is imperative that the decisions of acquiring, scheduling, and routing of vehicles are made optimally and quickly. In this paper, we consider a variant of a truckload open vehicle routing problem with time windows which exhibits vehicle routing operations characteristics in a humanitarian crisis. We present two integer linear programming models to formulate the problem. The first model is an arc-based mixed integer linear programming model. The second model is a path-based integer linear programming model; we design two fast path generation algorithms to formulate this model. The first model is solved exactly using the commercial solver, while we design a column generation algorithm to solve the second model. We compare the results obtained from the two models and show that the path-based model with column generation algorithm outperforms the arc-based model when solved exactly in terms of solution time without sacrificing the solution quality.

**Key words:** Humanitarian logistics, Truckload open vehicle routing problem with time windows, Column generation algorithm, Path generation algorithm.

## 1. Introduction

The motivation for our work stems from planning vehicle routing operations under a humanitarian logistics perspective. During a humanitarian crisis arising from either natural (flood, earthquakes, hurricanes, etc.) or anthropogenic (e.g., wars) causes, international and local humanitarian organizations (HOs) work towards a fast response to the needs of the affected populations. These agencies often are the only means of providing immediate response to an urgent need for commodities, such as dry food, water, first aid and other medicine, power source, clothes. To avoid a duplication of efforts, organizations like The Logistics Cluster sometimes act as a liaison for information gathering (data collection), operational decision-making, and coordination of the available resources for improving the timeliness and efficiency of the humanitarian operations taking place.

Due to the unpredictable nature of disaster occurrences, HOs are not able to maintain a dedicated fleet of vehicles at all locations and route them as needed after a disaster hits. Instead, when a disaster strikes a specific region, HOs make decisions on procuring the necessary transportation means in addition to the sourcing decisions for food and other resources. In land-based transportation, these decisions include vehicle acquisition and their retention period and vehicle routing for picking up resources from local resource depots (sources) and delivering them to the affected sites. Moreover, the unpredictable disaster occurrence nature makes vehicle availability limited and their acquisition costly. This is why it is crucial for humanitarian logistics coordinator (e.g., The Logistics Cluster) to make prompt and optimal decisions regarding vehicle acquisition, scheduling, and routing to deliver all necessary sustenance to the affected people.

Moreover, this problem setting satisfies our assumptions (see Section 2.1). For our assumptions where vehicles are tasked to deal with one relief mission at a time and reserve their full capacity for that specific task, this is often the case in humanitarian logistics. In many situations, vehicles are loaded with multiple products in a single warehouse, deliver them to the population in need, before returning to another warehouse for the next mission [1]. Assumption 3 states that each area is covered by the closest depot: this is an assumption followed in [2] and is common in humanitarian logistics as certain items have limited lifetime outside their storage facilities. Finally, assumptions 4 and 5 are common in the literature of general logistics, and also apply in humanitarian operations (see, e.g., the sufficiency assumption in [3]).The assumptions made above do not only hold for humanitarian relief operations; instead, they also apply to casualty transportation (see, e.g., [4]), in which injured people in need of medical assistance are transported from their locations to a medical center. This is easily adapted in our model by appropriately changing the terms of depots and customers to locations with people in need of medical assistance and medical units/hospitals.

To that end, we investigate a variant of open vehicle truckload pickup and delivery problem with vehicle rent period restriction and delivery time restrictions. Our objective is to minimize the overall cost of transportation while ensuring that emergency resources reach the affected areas in due time. We present a path generation algorithm and a solution algorithm to obtain fast and optimal vehicle acquisition, scheduling, and routing decisions. In our study, we have used delivery time as the main factor for vehicle path building, but we can easily accommodate any other priority setting in our path generation algorithm.

In land-based transportation, a significant portion of the material volume is moved through truckload deliveries. Each truckload shipment involves a pickup location and a delivery destination, most often with dispatch and arrival time restriction on the trip ends. We study a truckload vehicle scheduling and routing problem with delivery time targets from the perspective of a central resource coordinator. The coordinator has a set of depots from which truckload deliveries are sent to a set of clients when orders are placed. The vehicles are acquired from a third-party logistics (3PL) service provider. Given a set of orders each with client location with delivery time and its nearest depot location, the supplier faces several challenges, namely i) vehicle acquisition decisions: how many vehicles to acquire from dedicated contract carriers and for how long each truck should be rented, ii) scheduling decisions: at what time each acquired truck should start its journey and when should it return to the carrier, and iii) routing decisions: grouping of orders and assigning a truck to each group. The objective of the supplier is to minimize the total transportation cost of acquiring through minimizing the number of vehicles and total empty travel distance. The problem is a special case of the classic vehicle routing problem with time windows (VRPTW) and can be termed as the truckload open vehicle pickup and delivery problem with time windows (TOVPDPTW). The number of research studies on the VRPTW and its variants in the operations research/transportation science literature is significant as the problem is of interest to both academic and practitioner societies. Interested readers are referred to the review works [5, 6], which can provide an excellent overview of the studies in this area. Among the many variants of VRPTW, the pick-up and delivery problem with time windows (PDPTW) [7], the open vehicle routing problems (OVRP) [8], the dial-a-ride problem (DARP) [9], and vehicle routing problem with crew scheduling [10] etc. have gained much attention.

The VRPTW and its variants and other special cases have been studied extensively. These studies consider a series of restrictions and formulate and solve the underlying optimization problems using a set of techniques. The state-of-the-art for formulating a decision problem for the VRPTW or its special cases is based on arc or path based set covering models, which are then solved to optimality using branch-and-price or branch-and-cut-and-price with various valid inequalities as acceleration techniques. A recent study by Ceselli et al. [11] can be regarded as a representative work for the current practice in dealing with large-scale RPTW problem. This study focuses on the traditional VRPTW with a heterogeneous fleet of vehicles, incompatible products, and driver working hour restrictions. The problem is formulated as a set covering problem and we then employ a column generation solution approach to efficiently tackle larger scale instances. The pricing problems are formulated as resource-constrained shortest path problems, which are solved by a bidirectional dynamic programming algorithm. Ropke and Cordeau [12] propose a branch-and-cut-and-price algorithm for solving a pickup-and-delivery problem with time windows. They use a set partitioning formulation with two classes of shortest path pricing problems. Several sets of valid inequalities are also incorporated in the branch-and-cut cut framework. A similar problem is studied by Baldacci et al. [7]: an exact algorithm based on a set partitioning integer formulation with a bounding procedure is devised. The algorithm is a combination of dual ascent heuristics and cut-and-column generation procedure that is capable of finding near-optimal solutions. Furtudo et al. [13] study a PDPTW along with pairing and precedence restrictions of pick-up and delivery points. They present a compact formulation and use an exact solution method. A special case of the PDPTW, referred to as pick-up and delivery with transfer for shuttle routes is studied by Masson et al. [14]. The authors present three models and develop a branch-and-cut-and-price algorithm for solving them. They also present three families of valid inequalities for speeding up the solution process.

Problems involving scheduling of truckload pick-up and delivery, as well as timetabled bus trips, are natural variants of the PDPTW. A vehicle routing and scheduling problem with full truckload (VRPFL) with multiple depot and pick-up time windows is presented by Arunapuram et al. [15]. The authors use a column generation scheme inside a branch-and-bound framework to solve the integer programming model for the problem. Hadjar et al. [16] use a similar solution approach, to solve a multiple depot vehicle scheduling problem with time windows. In addition to column generation in a branch-and-bound framework, they also use variable fixing, tree search, and cutting planes to speed up the solution process. A multi-vehicle truckload pick-up and delivery problem with time windows (TPDPTW) is studied by Yang et al. [17], in which the authors consider the costs of empty travel distances of vehicles (travel in which vehicles carry nothing), delay, and rejections of jobs in developing optimal on-line and off-line policies. Compared with heuristic rolling-horizon policies, the policy given by re-optimizing the off-line model is found to be more competitive. The authors suggest that taking into consideration future job arrival distributions in scheduling and routing decision making process can result in a better policy. Scheduling of bus trips is a special family of pick-up and delivery problems; different formulation techniques and solution approaches have been developed for tackling such problems. Kliewer et al. [18] present a time-space network model for a multi-depot multi-vehicle-type bus scheduling problem. An extension of this work was done to incorporate crew scheduling in the same modeling framework [10]. As a solution method, a column generation scheme us used, where the pricing subproblems are formulated as constrained shortest path problems in a time-space network. For solving the pricing problems, a dynamic programming algorithm is devised and used.

Among the few studies that consider vehicle tour durations is the dial-a-ride problem studied by Cordeau [9]. This problem is a variant of PDPTW with capacity and route duration constraints. A branch-and-cut algorithm with several families of valid inequalities is used to solve the model. Archetti and Savelsbergh [19], on the other hand, consider hours of service regulations for drivers in the truckload deliveries with pick-up time windows. A backward search-based algorithm is presented, which can generate a feasible schedule in polynomial time, if one exists. In a more recent study, an exact method for integrated vehicle routing with time windows and truck driver scheduling (VRTDSP) is presented by Goel and Irnich [20]. The exact branch-and-price algorithm is based on an auxiliary network which enables one to model all possible driver activities on its arcs. This network is then transformed into a parameter-free network which is used in formulating elementary shortest path problems as the pricing problems.

Open vehicle routing problems (OVRP) have been receiving increasing attention in recent years due to the rise in 3PL logistics service providers and subscribers. An excellent review of the research works in OVRP is done by Li et al. [21], which can provide an overview of the algorithms, datasets, and computational results in this area. One of the first exact methods for OVRP is presented by Letchford et al. [8]; they formulate the OVRP as an integer linear programming (ILP) model and develop a branch-and-cut based algorithm. This algorithm is adapted from the branch-and-cut algorithm used for closed VRP and combined with the separation algorithm involving several sets of valid inequalities. Salari et al. [22] present a heuristic procedure for improving an initial feasible solution for an OVRP. This heuristic is based on ILP techniques, where an existing solution is destroyed by randomly removing customers from a path, and then solving an ILP to find an improved solution. Although these studies do not consider time windows for customers, they are among the very few studies that attempt to solve the OVRP using an exact method. An OVRP with time windows (OVRPTP) is studied by Repoussis et al. [23]; the authors presented a binary ILP model for a generalized OVRP, for which the traditional VRP is a special case. A greedy look-ahead route construction heuristic algorithm is proposed which introduces a new criterion for inserting a customer into a vehicle path. This criterion utilizes time windows related information efficiently and keeps track of total waiting time of a vehicle along a partially formed path as well as helps to determine strategies for finding seed customers for building paths. Several other studies investigate the

OVRP under various conditions. These studies use heuristic and metaheuristic algorithms as a solution approach, and include hybrid evolution strategy [24], multi-start algorithms [25], hybrid genetic algorithms [26], variable neighborhood search algorithms [27], among others.

In this study, we investigate a variant of the open vehicle pickup and delivery problem from the perspective of a central resource coordinator. The coordinator is tasked with arranging truckload deliveries at the client locations at target delivery times after picking up the load from designated resource depots. To meet delivery requirements, the coordinator must rent a sufficient number of vehicles and provide them with tour plans while abiding by all restrictions of vehicle rent period and delivery time at client location. It should be noted here that a vehicle tour neither begins nor ends in a central depot, rather it starts from the first pickup location (resource depot) of the first task and ends in the last delivery location (client site) of the last task on the tour. The literature on open vehicle routing problems with time windows (OVRPTW) are scarce; most of these studies adopt heuristic and metaheuristic approaches for solving the underlying optimization problem. The variant that we study here is an open vehicle pickup and delivery problem with time targets and without a central tour starting point. To the best of our knowledge, this particular problem variant has not been investigated fully in the literature, and related problems have been solved using heuristic algorithms. To fill this gap, we present two models to formulate our optimization problems and solve both models to optimality and compare their results. The first model is solved using a commercial solver, while we present a fast vehicle path generation algorithm in conjunction with a column generation algorithm to solve the second model.

In Section 2, we propose a mixed integer programming model to formulate the decision problem and present the results obtained from this model. Next, in Section 3, we present a column generation scheme to solve the problem that is based on a task conflict graph. We reformulate our problem as a new integer linear program and use the conflict graph to generate our initial set of columns. We then present a pricing subproblem to select new candidate columns. In Section 4, we compare the computational results obtained from the two approaches from the previous sections of the manuscript. In Section 5, we discuss the limitations of the current study and present potential future extensions.

## 2. Mixed Integer Linear Programming Model

The mixed integer linear programming model presented in this section takes as an input the list of orders available at the beginning of the planning horizon of a specified length. Each order is characterized by:

1. Destination client location and its nearest resource depot.
2. Target delivery time window.

3. Task duration (sum of loading time, travelling time from pickup to destination, unloading time).

The model objective is to minimize acquisition, operating, and empty traveling costs of the vehicles while meeting delivery targets and vehicle renting time constraints. At the beginning of the planning horizon, based on the task list, vehicle acquisition, scheduling and routing decisions are made. The model assumptions and the decisions are the following:

### 2.1. Assumptions

The assumptions we make in this work are as follows.

1. A truck is dedicated to perform each task.
2. Loading and unloading times are included in the calculation of task duration.
3. Each client is served from the nearest resource depot.
4. For divested vehicles, a fraction of the cost of the unused rent period is reimbursed by the contract carrier.
5. Supply of resources is sufficient at the depots.

## 2.2. Model Decisions

As we will discuss in the mathematical formulation, our decisions are threefold:

1. Number of vehicles to acquire and their lengths of renting periods.
2. Vehicle tours composed of assigned tasks.
3. Starting and ending times of each vehicle tour.

The decisions will be tied to specific decision variables that will be introduced later in the section.

### 2.3. Input Sets

Our input sets are:

$\mathcal{T}$ Set of tasks indexed by $t$; each task $t$ is characterized by a tuple as follows:

$$\{C_t(client\ ) \quad R_t(nearest\ resource\ depot\ ) \quad T_t(delivery\ time)\}$$

$\mathcal{K}$ Set of vehicles at supplier locations indexed by $k$; each vehicle $k$ is characterized by a tuple as follows:

$$\{r_k(rent\ period) \quad c_k(cost\ per\ unit\ time) \quad p_k(reimbursed\ amount\ \ per\ unit\ unused\ time)\}$$

To formulate the decision problem as a mixed integer programming (MIP) model, we created a graph $G$ consisting of node set ($\mathcal{N}$) and arc set ($\mathcal{A}$). These sets of nodes and arcs are defined as follows:

1. The node set ($\mathcal{N}$) is obtained by joining the task set ($\mathcal{T}$) with an artificial source node ($D$) and an artificial sink node ($S$).
2. The arc set ($\mathcal{A}$) is composed of three subsets of arcs:
   a. Pull-out arc subset ($A_{po}$): This subset contains pull-out arcs, each of which starts from global source node ends in a task node, i.e. for all pull-out arc $(i,j) \in A_{po}: i \in D, j \in \mathcal{T}$.
   b. Deadhead arc subset ($A_{dh}$): This subset contains deadhead arcs; a deadhead arc starts from and ends in task nodes, i.e. for all deadhead arc $(i,j) \in A_{dh}: i \in \mathcal{T}, j \in \mathcal{T}, T_i < T_j$, where $T_i$ is the delivery time of task $i$.
   c. Pull-in arc subset ($A_{pi}$): This subset contains pull-in arcs; each of these arcs starts from a task node and ends in global sink node, i.e. for all pull-out arc $(i,j) \in A_{po}: i \in \mathcal{T}, j \in S$.

Figure 1 below shows a simplified graphical representation of the problem. The set of arcs with the same color in the figure indicates a representative vehicle's tour composed of tasks.

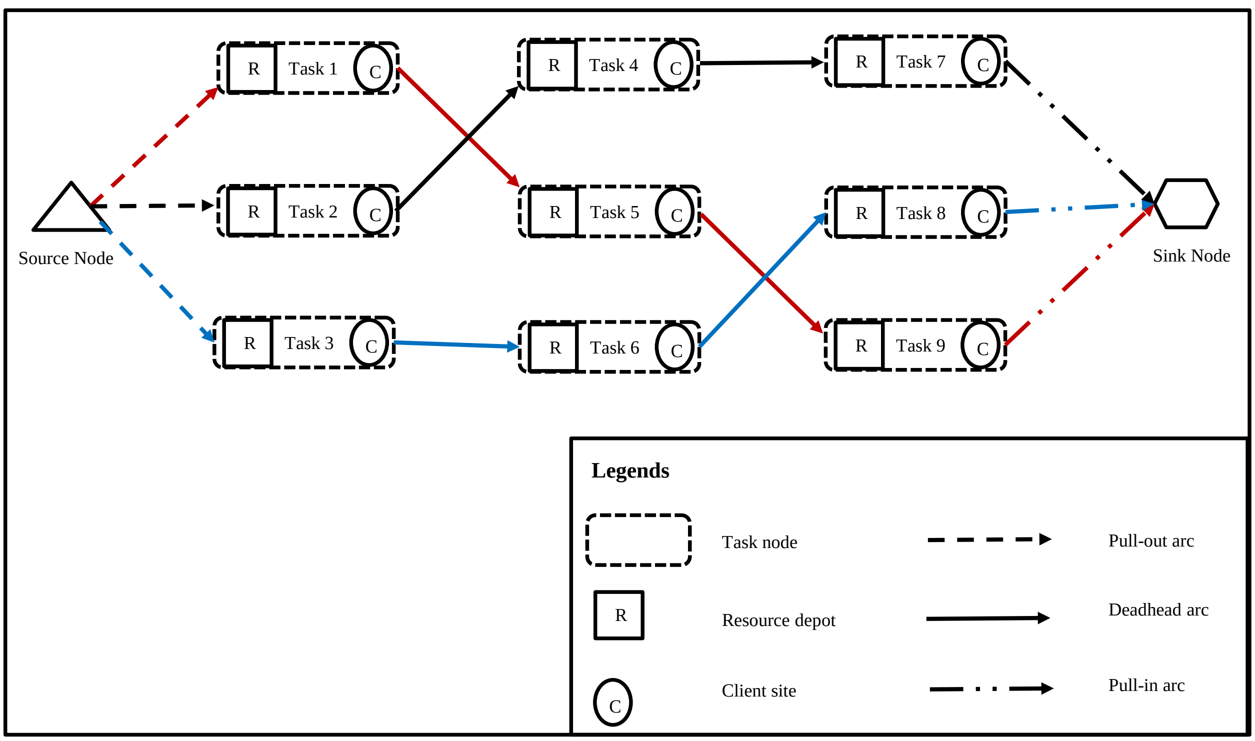


Figure 1: A simplified graphical representation of the network considered in the model.

## 2.4. Mathematical Model

Based on the definitions in the previous subsections, our notation can be viewed as the following.

Sets

$\mathcal{T}$ Set of tasks

$\mathcal{K}$ Set of vehicles

$\mathcal{A}$ Set of arcs

Parameters

$r_k$ Rent period length of vehicle $k \in \mathcal{K}$

$p_k$ Reimbursed amount for vehicle $k \in \mathcal{K}$ per unit unused time

$c_k$ Operating cost of vehicle $k \in \mathcal{K}$ per unit time

$d_{ij}$ Deadhead distance between task $i \in \mathcal{T}$ and task $j \in \mathcal{T}$

$T_i$ delivery time of task $i \in \mathcal{T}$

$\pi_i$ Duration of task $i \in \mathcal{T}$

$\epsilon$ Maximum allowable vehicle idle time between two consecutive tasks

Variables

$Y_{ijk}$ $\{0,1\}$: $Y_{ijk} = 1$, if arc $(i,j) \in \mathcal{A}$ is traversed by vehicle $k \in \mathcal{K}$

$\tau_k^s$ Tour starting time of vehicle $k \in \mathcal{K}$

$\tau_k^f$ Tour finishing time of vehicle $k \in \mathcal{K}$

The mathematical model for the decision problem is now presented below in equations (1)-(13).

$$Minimize \sum_{k \in \mathcal{K}} \sum_{(i,j) \in A_{po}} Y_{ijk}\, r_k c_k (1 - p_k) + \sum_{k \in \mathcal{K}} (\tau_k^f - \tau_k^s) c_k p_k \qquad (1)$$

Subject to:

$$\sum_{k \in \mathcal{K}} \sum_{(i,j) \in A_{po} \cup A_{dh}} Y_{ijk} = 1 \qquad \forall j \in \mathcal{T} \qquad (2)$$

$$\sum_{(i,j) \in A_{po} \cup A_{dh}} Y_{ijk} - \sum_{(j,i) \in A_{pi} \cup A_{dh}} Y_{jik} = 0 \qquad \forall j \in \mathcal{T}, k \in \mathcal{K} \qquad (3)$$

$$\sum_{(i,j) \in A_{po}} Y_{ijk} \leq 1 \qquad \forall k \in \mathcal{K} \qquad (4)$$

$$\sum_{(i,j)\in A_{po}} Y_{ijk} - \sum_{(j,i)\in A_{pi}} Y_{jik} = 0 \qquad \forall j \in \mathcal{T}, k \in \mathcal{K} \tag{5}$$

$$\tau_k^s \geq T_j - \pi_j - M(1 - Y_{ijk}) \qquad \forall (i,j) \in A_{po}, k \in \mathcal{K} \tag{6}$$

$$\tau_k^s \leq T_j - \pi_j + M(1 - Y_{ijk}) \qquad \forall (i,j) \in A_{po}, k \in \mathcal{K} \tag{7}$$

$$\tau_k^f \geq T_i - M(1 - Y_{ijk}) \qquad \forall (i,j) \in A_{pi}, k \in \mathcal{K} \tag{8}$$

$$\tau_k^f \leq T_i + M(1 - Y_{ijk}) \qquad \forall (i,j) \in A_{pi}, k \in \mathcal{K} \tag{9}$$

$$T_i \leq T_j + d_{ji} + M(1 - Y_{jik}) \qquad \forall (j,i) \in A_{dh}, k \in \mathcal{K} \tag{10}$$

$$T_i \geq T_j + d_{ji} - M(1 - Y_{jik}) \qquad \forall (j,i) \in A_{dh}, k \in \mathcal{K} \tag{11}$$

$$\tau_k^f - \tau_k^s \leq r_k \sum_{(i,j)\in A_{po}} Y_{jik} \qquad \forall k \in \mathcal{K} \tag{12}$$

$$\tau_k^s, \tau_k^f \geq 0; \quad Y_{ijk} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{13}$$

The objective function in (1) minimizes the total transportation cost. If a vehicle $k$ is acquired from a contract carrier with a renting period of $r_k$, a cost of $r_k c_k$ is incurred. If this vehicle is used for $(\tau_k^f - \tau_k^s)$ units of time, then the supplier receives a reimbursement amount of $r_k - (\tau_k^f - \tau_k^s)p_k$ from the contract carrier. Constraints (2) ensure that exactly one delivery is made for all task nodes, while constraints (3) restrict that an incoming truck to a task node must depart from that same node. Constraints (4)-(5) ensure that a vehicle tour can only have one starting arc and one ending arc, and they must start from the source node and end in the sink node. The tour start and end times for each vehicle are given by constraints (6)-(9). Feasible deadhead arcs between two consecutive tasks in a vehicle's tour are dictated by constraints (10) and (11). A vehicle tour length is constrained by the renting period of that vehicle, which is presented in constraint (12). The non-negativity and binary nature of the variables are shown in constraints (13). For the parameter $M$ used in constraints (6)-(11), we use the largest delivery time target among all the tasks in the task set.

### 2.5. Preliminary Result

We implemented the preliminary MIP model in AMPL [28] and solved it to optimality using CPLEX 12.7.1 solver. We have a computer with Dual Intel Xeon Processor (12 Core, 2.3GHz Turbo) with 64 GB of RAM, and a 64 bit operating system. We solved the model with randomly generated data for 10, 20, 30, 40, and 50 task problems. Each task has an origin in one of the three resource depot locations and a destination in one of ten potential locations. We have considered two classes of vehicles with respective cost and renting period parameters. The vehicle fleet has the size of 30, where each vehicle in class is characterized by a fixed cost, a rental period, and a reimbursement amount per unutilized unit of time for that vehicle class. We have used a 'mipgap' setting of 0.001 while solving the model, which indicates that the obtained results are provably within 0.1% of the optimal solution. For each problem size, we have solved the model with five sets of randomly generated instances. The average CPU times for different problem sizes are summarized in Table 1.

Table 1: Result summary of the preliminary model.

| Problem size (Number of task) | CPU time (seconds) |
| --- | --- |
| 10 | 1.82 |
| 20 | 4.37 |
| 30 | 44.59 |
| 40 | 251.89 |
| 50 | 910.52 |

## 3. Column Generation Algorithm

The results from the MIP model in Section 2 show that the model is capable of solving small instances (smaller number of tasks) in reasonable time, but it (expectedly) becomes computationally expensive as the problem size increases. To solve larger scale problems efficiently using a column generation approach, we consider a reformulation of the problem as described in the subsequent subsections. Column generation is a popular approach for solving VRPs with side constraints. In this approach, a smaller subproblem, referred to as the pricing subproblem (see also subsection 3.3), is solved to generate new, suitable columns, which are in turn fed to a restricted master problem (presented in subsection 3.2). We now begin our discussion for this approach with a method to generate an initial set of columns.

### 3.1. Generation of Feasible Task Combination Set

We have previously declared $\mathcal{T}$ as the set of tasks and $\mathcal{K}$ as the set of vehicles. Now, we define $C$ as the set of all *task combination subsets* that are *feasible*. To find these subsets, we first construct a graph $G(\mathcal{V};\mathcal{A})$ with:

1. Node set $\mathcal{V} = \mathcal{T}$;
2. Arc set $\mathcal{A} = \{(t_1, t_2)\text{: if tasks } t_1 \text{ and } t_2 \text{ can be assigned to the same vehicle}\}$.

This graph is constructed based on two conditions. An arc $(i, j)$ is in $\mathcal{A}$ if:

1. $T_i \leq T_j$ and $T_j \geq T_j + d_{ij} + \pi_j$
2. $T_i \leq T_j$ and $T_j \leq T_i + d_{ij} + \pi_j + \epsilon$

; where $d_{ij}$ is the deadhead distance between task $i \in \mathcal{T}$ and task $j \in \mathcal{T}$ and $\epsilon$ is the maximum allowable time that a vehicle can remain idle between two consecutive tasks.

From the graph generated as described above, we create the set of all feasible tasks, $C$, by following:

1. For each task $i$, we create the adjacent task node set $(ADJ)_i$, which is given by:

$$(ADJ)_i = \{j: j \in \mathcal{T} \ and \ (i, j) \in \mathcal{A}\}.$$

2. Considering each task $i$ as the starting point of a task combination subset, we generate subsets of cardinality $1,2, \ldots n$. Each of these (ordered by delivery time) subsets contains task nodes that can be directly or indirectly reached from task $i \in \mathcal{T}$. Let $S_i^n$ denote the task combination subset of cardinality $n$ and task $i \in \mathcal{T}$ as the first task of the combination. Then

$$S_i^1 = \{i\}$$
$$S_i^2 = \{\{i, j\}: j \in (ADJ)_i\}$$
$$S_i^3 = \{\{i, j, k\}: j \in (ADJ)_i \ and \ k \in (ADJ)_j\}$$

3. Following steps 1 and 2 for each task $i \in \mathcal{T}$, we generate task combination subsets. Each of these task combination subsets contains $i$ as the starting task. We merge all these task combination subsets to generate the feasible task combination set $C$.

Once we have generated all the feasible paths (task combinations), we present the following path-based integer linear programming formulation for our decision problem.

### 3.2. Restricted Master Problem

Sets

$C$ Feasible task combination set

Parameters

$l_c$ Duration of task combination $c$

$a_{ct}$ 1, if task $t$ is in task combination $c$

Variables

$y_{kc}$ 1, if vehicle $k$ is assigned to task combination $c \in C$

The restricted master problem is presented below with equations (14)-(18).

$$Minimize \sum_{c \in C} \sum_{k \in \mathcal{K}} y_{kc}[r_k c_k - p_k(r_k - l_c)] \tag{14}$$

Subject to:

$$\sum_{c \in C} \sum_{k \in K} y_{kc}\, a_{ct} \geq 1 \qquad \forall t \in \mathcal{T} \tag{15}$$

$$\sum_{c \in C} y_{kc} \leq 1 \qquad \forall k \in \mathcal{K} \tag{16}$$

$$\sum_{c \in C} y_{kc}\,(l_c - r_k) \leq 0 \qquad \forall k \in \mathcal{K} \tag{17}$$

$$y_{kc} \in \{0,1\} \qquad \forall k \in \mathcal{K}, c \in C \tag{18}$$

The objective function in (14) minimizes the total cost of vehicle assignment, which is the sum of differences between total vehicle acquisition costs and the total reimbursed amounts received for divesting vehicles before their rental periods are expired. Constraints (15) and (16) capture the fact that every vehicle can be assigned to at most one task combination and every task is to be assigned at least to one vehicle. The next constraint family, shown in (17), takes into account each vehicle rental period. This constraint only allows a vehicle to be assigned to a task combination if it can satisfy all the demands in the tasks of that assignment within its rental period. Finally, (18) restricts all of our variables to be binary.

The formulation presented in (14)—(18) has exponentially many variables; hence we cannot hope to directly solve large-scale problems following this formulation. Instead, we propose the following decomposition scheme.

1. Start with an initial set of task combinations $C_1 \in C$.
2. Define a restricted master problem (RMP) that solves the model in (14)-(18) over set $C_1$ alone.
3. Identify potentially good task combinations to enter in the restricted set $C_1$.
4. Re-solve RMP until we can declare optimality.

The initial restricted set of dominating sets $C_1$ will, most probably, not be enough to declare optimality; it is actually probable that it will not even be feasible. Now, we consider the following restricted dual problem to identify good task combinations to enter in $C_1$:

$$Miximize \sum_{k\in\mathcal{K}}(\lambda_k+\gamma_k)+\sum_{t\in\mathcal{T}}\mu_t \tag{19}$$

Subject to:

$$\lambda_k+\gamma_k+\sum_{t\in T}a_{ct}\mu_t \geq r_k c_k - p_k(r_k-l_c) \qquad \forall k\in\mathcal{K}, c\in C_1 \tag{20}$$

where

$\mu_t$ is dual variable for constraint (15)

$\lambda_k$ is dual variable for constraint (16)

$\gamma_k$ is dual variable for constraint (17)

We are then interested in admitting task combinations from the set $C_2=C\backslash C_1$ that violate constraint (20). Hence, our optimality condition is:

$$\lambda_k+\gamma_k+\sum_{t\in\mathcal{T}}a_{ct}\mu_t - r_k c_k - p_k(r_k-l_c) \geq 0 \qquad \forall k\in\mathcal{K}, c\in C_1 \tag{21}$$

If the above is nonnegative for all vehicles and all task combinations, we can terminate with an optimal solution. For a specific vehicle $\bar{k}\in\mathcal{K}$, we can solve then the pricing problem below to identify a good task combination to enter into the restricted task combination set:

### 3.3. Pricing problem

$$P(\bar{k}): \quad \lambda_k+\gamma_k(l_c-r_k)+Minimize\sum_{c\in C_2}\sum_{t\in\mathcal{T}}a_{ct}\mu_t\phi_c-\sum_{c\in C_2}\phi_c[r_k c_k-p_k(r_k-l_c)] \tag{22}$$

Subject to:

$$\sum_{c\in C_2}\phi_c=1 \tag{23}$$

$$\phi_c\in\{0,1\} \qquad \forall c\in C_2 \tag{24}$$

The pricing subproblem will give us the most violated constraint (if the objective function is negative) for each vehicle $\bar{k} \in \mathcal{K}$. Iterating over all vehicles, we can at most get $|\mathcal{K}|$ dominating sets to enter in our restricted set and re-solve the RMP.

### 3.4. Initial task combination sets (paths) generation for RMP

To create an initial task combination set $C_1$ for starting the RMP, the following procedure is used:

1. Considering each task $i$ as a starting point, a task combination set $S_i = \{i\}$ is generated.
2. This set is appended by adding task $\{j: j \in (ADJ)_i\}$ that results in least slack time between tasks. $S_i = \{i, j\}$.
3. Step 2 is repeated for the last element of $S_i$ until no more task can be added.
4. For each task $i$, the task that result in least slack time is removed and step 1-3 are repeated for the remaining tasks.

The RMP is solved with the path set $C_1$ generated following the above method. Rest of the feasible paths are stored in $C_2$. In Figure 2, the primary steps of the column generation algorithm are showed.

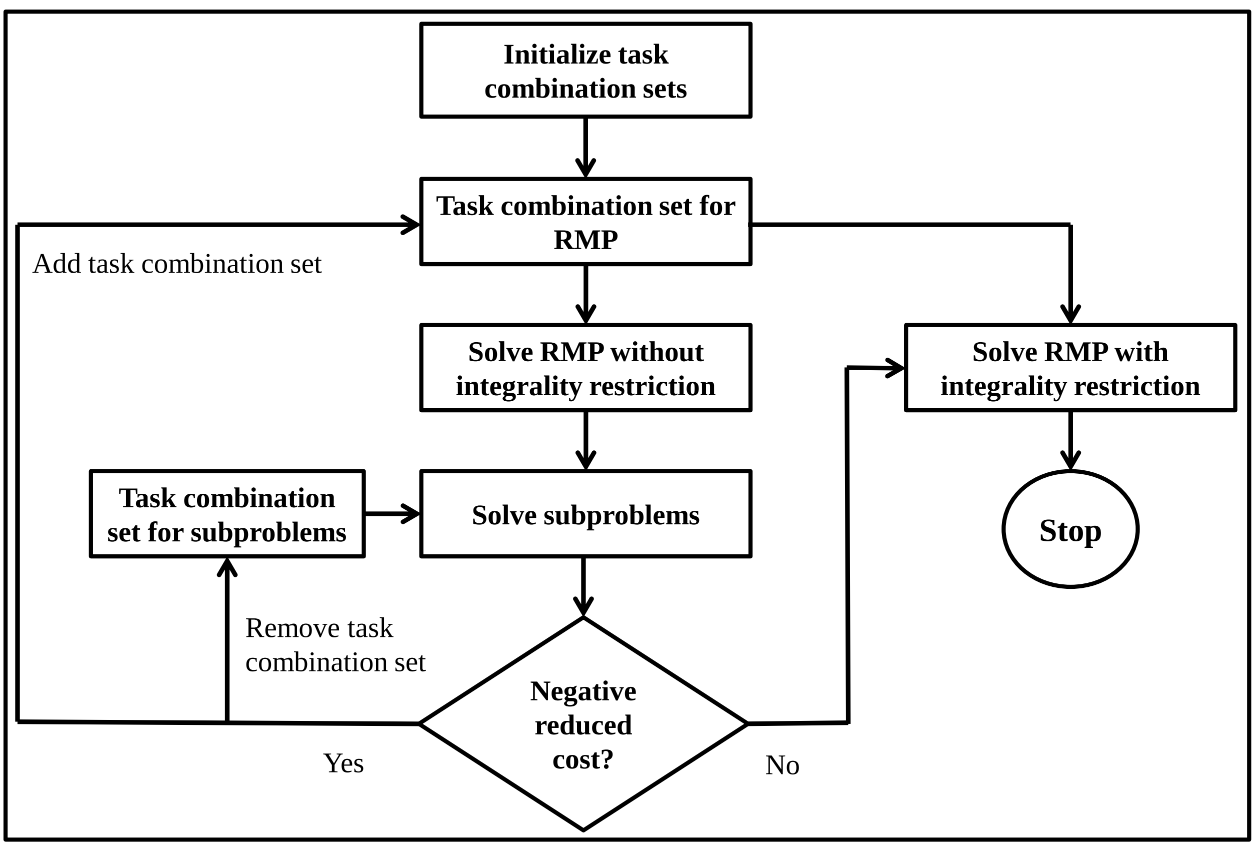


Figure 2: Flow diagram of the column generation algorithm.

## 4. Numerical Experiments

We used AMPL language and CPLEX 12.7.1 to implement the master and pricing problems, and our column generation algorithm. For each problem size, we have used the same five sets (used in our previous experiment) of randomly generated instances to run both the MIP model and the column generation algorithm. In every single case, the column generation algorithm outperforms the MIP model when solved to optimality in terms of CPU time, while providing the same (optimal) objective function values. The summary of our computational experiments is presented in Table 2.

Table 2: Comparison between the MIP model and column generation algorithm in terms of CPU time.

| Problem size (Number of task) | CPU time (seconds) | | | |
|---|---|---|---|---|
| | MIP model (CPLEX) | | Column generation | |
| | Average | Standard deviation | Average | Standard deviation |
| 10 | 1.82 | 1.69 | 1.46 | 0.38 |
| 20 | 4.37 | 3.42 | 2.10 | 0.79 |
| 30 | 44.59 | 47.12 | 4.20 | 2.88 |
| 40 | 251.89 | 118.37 | 9.03 | 3.61 |
| 50 | 910.52 | 350.82 | 17.47 | 4.50 |

## 5. Conclusion

In this study, we investigated a variant of open vehicle pickup and delivery problem from the perspective of a resource supplier. The supplier is required to rent vehicles from a contract supplier with rental period restrictions to complete a set of pickup and delivery tasks with delivery time targets. We presented two models for this vehicle acquisition, scheduling, and routing decision problem: an arc-based model and a path-based model. The arc-based MIP model is solved exactly using commercial solver CPLEX; the model provides the number of vehicles to rent, their starting times, and finishing times. The path-based integer model, on the other hand, is solved by a column generation algorithm. Paths (combination of tasks) are generated in the pre-processing steps. We presented two path generation algorithms: the first algorithm generates all feasible paths (compatible task combinations), while the second algorithm provides a set of paths with least idle times for starting the initial restricted master problem. Both algorithms work very fast and the path generation times can be considered negligible. We compared the results and solution times given by the two models for five different size problems. The path-based model and column generation algorithm outperform the commercial solver and the MIP combination in terms of solution time for all the cases. The reduction in solution time by the column generation algorithm is more notable as the size of the problem increases. The reduction in solution time achieved by the column generation algorithm and the fast path generation algorithms indicate that the model can be re-solved repeatedly with updated task list to emulate a dynamic vehicle scheduling and routing problem. During humanitarian crises, when the damage status and demand rates in the affected region are revealed intermittently over time, this re-solvable feature of our model will be most effective.

As a future extension, we have several potential directions for making the model more applicable in real-world settings and modifying the solution algorithm accordingly. One of these directions includes the consideration of priority in developing paths (task combinations). We have used delivery times as the driving factor in path generation, but it can be easily replaced by some other priority rules: for example disaster affected areas with higher percentage of children, older and injured people may get higher priorities, and vehicles may be routed through these areas first. Including uncertainty in i) vehicle availability, ii) vehicle traveling time, and iii) delivery target time in the decision model is another potential extension of our model. To accommodate this uncertainty, we will redesign our path generation algorithms to obtain paths with probabilistic durations and modify our pricing problem to minimize the expected reduced cost of assigning a vehicle to a path. Another possible extension is the integration of vehicle GPS location data with the model; re-solving the model with current vehicle positions and statuses will provide more time efficient and effective decisions.

## References


[1] J.-B. Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," *Transportation Research Part E: Logistics and Transportation Review,* vol. 43, pp. 687-709, 2007.

[2] V. De Angelis, M. Mecoli, C. Nikoi, and G. Storchi, "Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola," *Computers & Operations Research,* vol. 34, pp. 1601-1615, 2007.

[3] D. Berkoune, J. Renaud, M. Rekik, and A. Ruiz, "Transportation in disaster response operations," *Socio-Economic Planning Sciences,* vol. 46, pp. 23-32, 2012.

[4] W. Yi and L. Özdamar, "A dynamic logistics coordination model for evacuation and support in disaster response activities," *European Journal of Operational Research,* vol. 179, pp. 1177-1193, 2007.

[5] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering,* vol. 57, pp. 1472-1483, 2009.

[6] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering,* vol. 99, pp. 300-313, 2016.

[7] R. Baldacci, E. Bartolini, and A. Mingozzi, "An exact algorithm for the pickup and delivery problem with time windows," *Operations research,* vol. 59, pp. 414-426, 2011.

[8] A. N. Letchford, J. Lysgaard, and R. W. Eglese, "A branch-and-cut algorithm for the capacitated open vehicle routing problem," *Journal of the Operational Research Society,* vol. 58, pp. 1642-1651, 2007.

[9] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Operations Research,* vol. 54, pp. 573-586, 2006.

[10] I. Steinzen, V. Gintner, L. Suhl, and N. Kliewer, "A time-space network approach for the integrated vehicle-and crew-scheduling problem with multiple depots," *Transportation Science,* vol. 44, pp. 367-382, 2010.

[11] A. Ceselli, G. Righini, and M. Salani, "A column generation algorithm for a rich vehicle-routing problem," *Transportation Science,* vol. 43, pp. 56-69, 2009.

[12] S. Ropke and J.-F. Cordeau, "Branch and cut and price for the pickup and delivery problem with time windows," *Transportation Science,* vol. 43, pp. 267-286, 2009.

[13] M. G. S. Furtado, P. Munari, and R. Morabito, "Pickup and delivery problem with time windows: a new compact two-index formulation," *Operations Research Letters,* vol. 45, pp. 334-341, 2017.

[14] R. Masson, S. Ropke, F. Lehuédé, and O. Péton, "A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes," *European Journal of Operational Research,* vol. 236, pp. 849-862, 2014.

[15] S. Arunapuram, K. Mathur, and D. Solow, "Vehicle routing and scheduling with full truckloads," *Transportation Science,* vol. 37, pp. 170-182, 2003.

[16] A. Hadjar, O. Marcotte, and F. Soumis, "A branch-and-cut algorithm for the multiple depot vehicle scheduling problem," *Operations Research,* vol. 54, pp. 130-149, 2006.

[17] J. Yang, P. Jaillet, and H. Mahmassani, "Real-time multivehicle truckload pickup and delivery problems," *Transportation Science,* vol. 38, pp. 135-148, 2004.

[18] N. Kliewer, T. Mellouli, and L. Suhl, "A time–space network based exact optimization model for multi-depot bus scheduling," *European journal of operational research,* vol. 175, pp. 1616-1627, 2006.

[19] C. Archetti and M. Savelsbergh, "The trip scheduling problem," *Transportation Science,* vol. 43, pp. 417-431, 2009.

[20] A. Goel and S. Irnich, "An exact method for vehicle routing and truck driver scheduling problems," *Transportation Science,* vol. 51, pp. 737-754, 2016.

[21] F. Li, B. Golden, and E. Wasil, "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results," *Computers & operations research,* vol. 34, pp. 2918-2930, 2007.

[22] M. Salari, P. Toth, and A. Tramontani, "An ILP improvement procedure for the open vehicle routing problem," *Computers & Operations Research,* vol. 37, pp. 2106-2120, 2010.

[23] P. P. Repoussis, C. D. Tarantilis, and G. Ioannou, "The open vehicle routing problem with time windows," *Journal of the Operational Research Society,* vol. 58, pp. 355-367, 2007.

[24] P. P. Repoussis, C. D. Tarantilis, O. Bräysy, and G. Ioannou, "A hybrid evolution strategy for the open vehicle routing problem," *Computers & Operations Research,* vol. 37, pp. 443-455, 2010.

[25] A. López-Sánchez, A. G. Hernández-Díaz, D. Vigo, R. Caballero, and J. Molina, "A multi-start algorithm for a balanced real-world Open Vehicle Routing Problem," *European Journal of Operational Research,* vol. 238, pp. 104-113, 2014.

[26] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," *Computers & operations research,* vol. 31, pp. 2037-2053, 2004.

[27] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *European Journal of Operational Research,* vol. 195, pp. 803-809, 2009.

[28] R. Fourer, D. Gay, and B. Kernighan, *Ampl* vol. 117: Boyd & Fraser Danvers, MA, 1993.