THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2009-13

Solving the Inventory Slack Routing Problem for Medication Distribution Planning

Adam Montjoy, Stephanie Brown, and Jeffrey W. Herrmann

Solving the Inventory Slack Routing Problem for Medication Distribution Planning

Adam Montjoy, Stephanie Brown, and Jeffrey W. Herrmann

Abstract

This paper presents a two-stage approach for solving the inventory slack routing problem in order to improve medication distribution planning, which is a critical issue in emergency preparedness. Public health officials must plan the logistics for distributing medication to points of dispensing (PODs), which will give medication to the public in case of a bioterrorist attack such as anthrax, while medication is still arriving. Our approach separates the problem into two subproblems: (1) the "routing problem" assigns sites to routes for each vehicle, and (2) the "scheduling problem" determines when the vehicles should start these routes and how much material should be delivered on each trip. This paper formulates the problem, describes the approach, and presents the results of using this approach to construct solutions for a variety of scenarios.

Keywords: Emergency Preparedness, Scheduling, Vehicle Routing, Logistics

Contents

List of Figures

List of Tables

1 Background

Events in the last ten years have highlighted the increased need for emergency preparedness by government officials. Events such as the terrorist attacks on September 11, 2001, Hurricane Katrina, and the 2008 earthquake in Chengdu, China, have provided real world examples of ill-preparation for major disasters [7]. Thus, it is important for government officials to anticipate disasters and plan accordingly. Mathematical models and decision support tools can be used to support planning activities.

Some scenarios could require the quick and efficient distribution of medication to a large number of people. For instance, the widespread release of anthrax in a metropolitan area could result in casualties equivalent to that of a small nuclear explosion [18]. In this scenario (and others involving mass vaccination against communicable diseases such as smallpox and influenza), it is logical to create Points of Dispensing (PODs) such that large populations can be given medication without having to travel to one location. PODs may be setup in schools, recreation centers, churches, and other non-medical facilities. The medication to be distributed at these PODs must be delivered quickly from a central depot as soon as it arrives.

The proposed research is motivated by work with public health officials in the state of Maryland who must plan the logistics for distributing medication to the PODs from a central location. We consider the problem at the state and local levels (not the national level). After the decision for mass dispensing is made, county public health departments will begin preparing to open multiple PODs simultaneously at a designated time. The state will request medication from the federal government, who will deliver an initial but limited supply of medication to a state receipt, storage, and stage (RSS) facility (which we call the "depot"). Contractors will deliver more medication to the depot, but the state will begin shipping medication from the depot to the PODs before everything arrives from the contractors. The deliveries to the depot arrive in batches that we call "waves."

Poor medication distribution plans will delay the time that some PODs receive medication. This can delay the opening of these PODs, and some residents may not get their medication in a timely manner, which increases their risk of death or illness. Clearly, there are many uncertainties in medication distribution, including the timing of shipments to the depot, the time needed to load and unload vehicles, travel times, and the demand for medication at each POD. For this reason, planners need a robust plan. In particular, it is better if the plan calls for delivering medication to PODs much earlier than it is needed. This improves the likelihood that the PODs will open on-time, will not run out of medication during operations, and will dispense medication to the largest number of people in a timely manner.

Specifically, the problem addressed has some features of the inventory routing problem but also has some unique assumptions, constraints, and objectives. In this case, a set of PODs are served by a given set of vehicles delivering a quantity of one item during a short time span. Thus, the objective is not to minimize the cost or maximize the profit. Instead the objective is to increase the time between a POD running out of supplies for each delivery made. This value will be known as the slack.

Much research has been done to develop models to improve emergency preparedness planning. Hupert *et al.* [18] have presented a model to predict the hospital surge after a large-scale anthrax attack. The researchers emphasize the importance of timely antibiotic distribution, making logistics of delivery equally important. Similarly, much work has been done to create simulation methods and planning tools for PODS in makeshift locations such as school gymnasiums [2, 1, 19].

The operations of firefighters, emergency medical services, and police departments have motivated research into location models [12, 4, 10] and dynamic vehicle routing models [22, 24, 15]. However, these models are not relevant to the medication distribution problem, which is more closely related to the inventory routing problem [13, 8, 3, 21] and the production-distribution scheduling problem [11]. Still, the models used for those problem are also not directly relevant.

Planning humanitarian logistics is related to the Vehicle Routing Problem (VRP) and Inventory Routing Problem (IRP). These problems have been applied to a variety of commercial, military, and government applications. The following description of the VRP is by Toth and Vigo [23].

The VRP details the delivery of a set of goods to a set of customers by a set of vehicles. These goods are stored at a depot, or a set of depots, and are delivered by a road network. This road network is usually detailed using a graph with arcs representing roads and vertices as the sites and depots. The solution to the VRP specifies a route for each vehicle that begins and ends at the depot. Typical VRP problems have the following characteristics: customer locations, demands for the customers, time windows for the customers, loading/unloading times, and a set of available vehicles that can be used.

In many cases, it may not be possible to fully satisfy all of the customer demand, and priorities or penalty functions must be employed. With this, it is possible to formulate various objective functions to obtain a solution, including minimization of global transportation cost, minimization of vehicles used, balancing routes for travel times and load, and minimization of penalties. The VRP is a well-researched technique with many heuristics, mathematical programming, and search techniques available.

The VRP has many variations including the Inventory Routing Problem (IRP). The following description of the IRP is by Campbell *et al.* [9]. The IRP differs from the VRP because the the delivery company decides when and what quantity to deliver to customers, as long as they do not run out. The objective is minimization of cost over the planning horizon while preventing customers from running out of product. A single product is delivered from a single depot to a set of n customers over a specified time period. These customers are served by a homogenous fleet of V vehicles with a capacity of Q. A problem solution should answer three questions: when to serve a customer, how much to deliver, and which routes to follow?

Most solutions detailed in literature focus on short-term scenarios solved by mathematical programming techniques. There is a lack of basic heuristics for solving IRPs. The Inventory Slack Routing Problem (ISRP) that we present is similar to an IRP but has some unique assumptions. The main concern is to supply medication as quickly as possible, not to minimize cost. As Hupert *et al.* [18] emphasize, delaying the start of POD operations will significantly increase the number of people hospitalized. In addition, the limited availability of medication at the depot adds an additional constraint to the problem. Finally, because there is uncertainty in loading/unloading, travel times, and demand, it is necessary to have overall maximum slack to hedge against these uncertainties. The objective of the ISRP is to maximize the minimum slack in order to develop a more robust plan.

We are interested in studying heuristics to develop fast procedures that can be easily implemented in spreadsheets for use by emergency preparedness planners. Also, such heuristics will be useful in column generation and other decomposition approaches. We are also developing search algorithms and mathematical programming approaches to generate good distribution plans. This paper proposes four heuristics for creating routes and scheduling deliveries. We describe the creation of instances for testing these heuristics and present the results of this testing.

2 Problem Formulation

In the ISRP, a set of vehicles must deliver material from a depot to a set of sites that will consume this material. Not all of the material is available at the depot at the beginning of the time frame. Instead, material will become available in waves, which are deliveries to the depot at different points in time. The sites will start operating at a designated time. Each site consumes material at a given rate, and this demand may vary from site to site. The vehicles must deliver enough material from the depot to the sites to satisfy the total demand over the time horizon. The following section details the notation to be used. Note than an example is provided in the Appendix.

Although, in theory, a vehicle could follow a different route each time it leaves the depot, and a site could be served by multiple vehicles, this makes supervising and performing the deliveries more complex in practice. We therefore assume that each and every site is assigned to exactly one vehicle, and each vehicle always follows the same route to visit the sites assigned to it.

2.1 Notation

t - Time in minutes

T_1 - Time, in minutes, that sites will begin operating

T_2 - Time, in minutes, that sites will end operating

$I(t)$ - Cumulative amount of material delivered to the depot between time 0 and t

V - Number of vehicles

C - Vehicle capacity in units of material

σ_v - Route assigned to vehicle v, $v = 1, ..., V$

σ - Routes for all vehicles

n - Number of sites

L_k - Demand in units per minute for sites $k = 1, ..., n$

p_k - Load (unload) time, in minutes, at sites $k = 1, ..., n + 1$

c_{ij} - Time, in minutes, to travel from site i to j

2.2 Formulation

In the ISRP, $t = 0$ refers to the first instant that material is available at the depot, $t = T_1$ is the time that the sites begin operating, and $t = T_2$ is the time that the sites stop operating. There are n sites denoted by $k = 1, ..., n$. The demand rate for sites is denoted as L_k material per time unit, which in this paper is minutes. Thus, site k has a total demand of $(T_2 - T_1)L_k$ units of material.

The depot, denoted by $k = n + 1$, receives material in multiple "waves" that arrive at different times. The times and quantities are known in advance and are used to determine the discontinuous, non-decreasing cumulative function $I(t)$. In our example, there are three waves. At $t = 0$, 48,000 units are delivered; at $t = 180$, 98,000 units are delivered; and at $t = 360$, 73,000 units are delivered. Figure 1 shows $I(t)$.

The time to load or unload a vehicle at site k is given by p_k. The time to travel from site i to site j is c_{ij}. An instance will have V vehicles at the depot where vehicle v has a capacity of C units.

A solution specifies, for each vehicle, a route, the number of trips that it makes, the time to start each trip, and the quantity to deliver to each site on each trip. Let r_v be the number of trips that vehicle v makes. Each trip j of vehicle v starts at time t_{vj} by loading at the depot and follows sequence σ_v. The quantity q_{vjk} is delivered to each site $k \in \sigma_v$ on trip j. Let y_v be the total duration of a trip by vehicle v.

The following constraints must be satisfied for a solution to be feasible.

The quantity shipped from the depot cannot exceed the amount delivered to the depot:

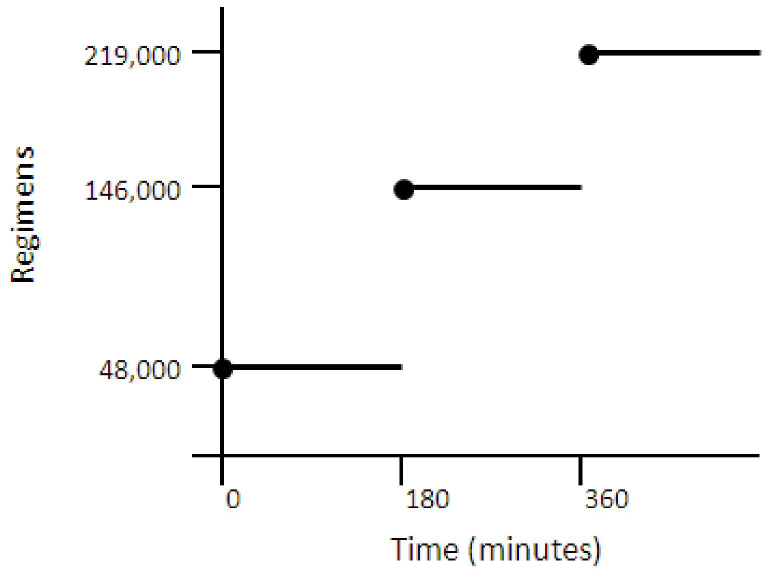

Fig. 1: Cumulative delivery function from R1

$$\sum_{(a,b):t_{ab}\leq t_{vj}} \sum_{k\in\sigma_a} q_{abk} \leq I(t_{vj}) \;\; v = 1, ..., V; j = 1, ..., r_v$$

A vehicle cannot begin a new route until it returns to the depot:

$$t_{vj} \geq t_{v,j-1} + y_v \;\; v = 1, ..., V; j = 2, ..., r_v$$

All delivery quantities are non-negative. Each vehicle has a fixed capacity and can carry a maximum of C units, that is $\sum\limits_{k\in\sigma_{vj}} q_{vjk} \leq C$ for all $v = 1, ..., V$ and $j = 1, ..., r_v$. All route start times are non-negative such that $t_{vj} \geq 0$ for all $v = 1, ...V$ and $j = 1, ..., r_v$. Each site must receive all required medication, that is $\sum\limits_{j=1}^{r_v} q_{vjk} = (T_2 - T_1)L_k$ for $v = 1, ..., V$ and $k \in \sigma_v$.

A feasible solution for our example is shown in Table 1. To evaluate a solution, we need to calculate its minimum slack. Let w_{vk} be the duration until vehicle v visits site k after it begins a trip. This is calculated as follows, where $[a]$ is the a-th site in route σ_v:

$$w_{vk} = p_{n+1} + c_{n+1,[1]} + p_{[1]} + c_{[1],[2]} + ... + p_k$$

For a site $k \in \sigma_v$, let Q_{vjk} be the quantity delivered to site k by vehicle v on trips before trip j:

$$Q_{vjk} = \sum_{i=1}^{j-1} q_{vik}$$

Note that $Q_{v1k} = 0$. If, on trip j, the vehicle's delivery at site k were delayed, then the site would run out of inventory at time $T_1 + Q_{vjk}/L_k$.

The slack for site k on trip j can be found as follows:

$$s_{vjk} = T_1 + \frac{Q_{vjk}}{L_k} - (t_{vj} + w_{vk})$$

The evaluation of a solution is the minimum slack over all vehicles, sites, and trips: $S = \min\{s_{vjk}\}$. Slack values for the example are given in Table 2.

3 Solution Approach

It is easy to see that the ISRP, like other versions of the VRP and IRP, is NP-hard, which makes it computationally expensive to obtain an exact solution. Therefore, our immediate research goal is to develop simple heuristics that can construct feasible solutions. A solution is a schedule for each vehicle with a starting time to begin loading for each trip, specified sites to visit, and a quantity to bring to each site on that trip.

The overall approach can be seen in Figure 2. This approach constructs a solution by separating the ISRP into two subproblems: routing and scheduling. A combination of different routing techniques will be discussed in the following section. The scheduling subproblem is further separated into scheduling for each vehicle by using the routes have been found.

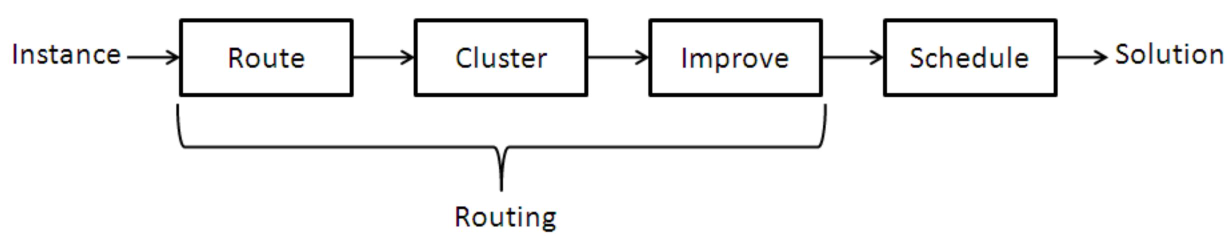

Fig. 2: Basic approach to obtaining a solution

4 Routing

The routing subproblem creates routes for each vehicle. It assigns sites to each vehicle and determines the order in which they are visited. The ISRP differs from traditional VRP because the objective is not to minimize total travel time. Instead, it is desirable to create routes that are nearly the same duration so that the minimum slack is not too small.

Bramel and Simchi-Levi [6] present two categories for this type of routing: (1) route first-cluster second methods and (2) cluster first-route second methods. With a route first-cluster second method, a tour is created through all of the sites, and then the sites (and the route) are divided into a desired number of partitions. Gillett and Miller's [14] sweep algorithm is a popular example of the route first-cluster second approach. One major drawback for these methods is that vehicles may be poorly utilized since the routing is done first. Algorithms have also been developed for cluster first-route second methods. These methods devote more priority to the clustering phase. Because these methods tend to require more computational effort and we are interested in heuristic approaches at this time, we will consider a route first-cluster second method.

4.1 Route First

When routing, it is first necessary to create a "big route" that visits all of the sites. We consider two different methods that do not use X-Y coordinates. In many real world situations, the X-Y coordinates are not as important as the travel times between sites, and, in some situations, the X-Y coordinates may be unavailable.

4.1.1 Nearest Neighbor

The nearest neighbor (NN) technique generates a tour through all of the sites. The tour starts at the depot. The next site selected for the tour is the site that has the shortest travel time from the current site and has not already been visited. This is repeated until no sites remain. Once all of the sites have been visited, the tour ends with the depot.

4.1.2 2-opt Exchange

Given an initial tour, the 2-opt exchange systematically removes two edges in the tour and reconnects the vertices to obtain a tour of shorter length. This algorithm finds all pairs of edges that will decrease the tour length. Of all these pairs, the pair chosen is the one that will make the greatest decrease in travel time of the tour. The 2-opt algorithm is continued until no more improving pairs can be found [20]. Although many pairwise, or 2-opt, exchange implementations start with a randomly generated trip through the sites, we use the nearest neighbor algorithm to first generate a route because it is computationally inexpensive.

4.2 Cluster Second

Once a big route has been obtained, it is necessary to divide the sites among all of the vehicles available. The sites are first divided between vehicles as equally as possible.

In the example, there are five sites, three vehicles, and the big route using NN is found to be $6, 5, 4, 3, 2, 1, 6$. The initial clusters will be as follows:

Vehicle 1: $6, 5, 4, 6$

Vehicle 2: $6, 3, 2, 6$

Vehicle 3: $6, 1, 6$

It is important to note that this initial cluster ignores both the demand and the travel times. Thus, the durations (and demands) of the routes may vary widely, which can reduce the slack of any solution constructed from these clusters. Thus, we use an improvement algorithm to reduce the variation. We tested an improvement algorithm that considers the travel time and one that considers the total demand.

4.2.1 Improvement by Route Duration

Each cluster is assigned to a vehicle. The vehicles are sequenced by the position of their cluster in the big route. This improvement algorithm method strives to make the route durations as similar as possible by minimizing the range of route durations. This method begins by calculating each vehicle's route duration. In each iteration, the algorithm examines the vehicles with maximum and minimum travel times and considers moving sites at the beginning (or end) of one route to the previous (or next) vehicle's route. If the potential move decreases the range of route durations, then the routes are updated to reflect this change. This continues until no further improvement can be made.

Of course, this type of local search may not find the smallest possible range. The pseudocode can be seen in the appendix. This pseudocode, and formulation is subsequent sections, require the following notation.

y_v is the route duration for vehicle v, which can be calculated by the following equation, where r_v denotes the number of sites on route σ_v and $[a]$ denotes the a-th site on route σ_v.

$$y_v = p_{n+1} + c_{n+1,[1]} + p_{[1]} + c_{[1],[2]} + ... + p_{[r_v]} + c_{[r_v],n+1}$$

This cluster improvement algorithm will be applied to the example introduced in the previous section. The durations of the initial clusters are 90, 124, and 64 minutes. The initial range of durations is 60 minutes. The following clusters are the result, and the range of route durations is 1 minute.

Vehicle 1: $6, 5, 4, 6$; route duration: 90 minutes

Vehicle 2: $6, 3, 6$; route duration: 90 minutes

Vehicle 3: $6, 2, 1, 6$; route duration: 91 minutes

4.2.2 Improvement by Total Demand

This improvement algorithm attempts to reduce the range of total demand of the sites on the routes but searches in the same way as the previous algorithm. That is, we replace y by D, where the total demand, D_v, can be calculated for vehicle v by the following equation.

$$D_v = (T_2 - T_1) \sum_{k \in \sigma_v} L_k$$

5 Scheduling

After constructing routes for the vehicles, it is necessary to schedule their deliveries. A schedule specifies the quantity to be delivered to each site as well as the time for the vehicle to begin loading for departure. Because we are interested in developing heuristics for the ISRP, we will allocate material to vehicles and schedule the deliveries of each vehicle using the following policies.

Let f_k be the relative demand of site k:

$$f_k = \frac{L_k}{\sum_{i=1}^{n} L_i}$$

For each vehicle, we create a cumulative material function $J_v(t)$ that describes the material available to be delivered by vehicle v at time t. Recall that $I(t)$ describes the total material received at the depot by time t. Then, if σ_v is the route that vehicle v visits, $J_v(t) = I(t) \sum_{k \epsilon \sigma_v} f_k$.

Once $J_v(t)$ has been established for vehicle v, it is then possible to determine how many trips the vehicle will take to service the sites assigned to it, what time each trip will start, and how much quantity to deliver each trip. As defined in Section 4.2, let y_v be the route duration for vehicle v to complete its route. Vehicle v will begin loading for its first route as soon as the depot has stock. Vehicle v will carry as much material as it can at the first instance the depot has material, which is $J_v(0)$, without surpassing its capacity. When the vehicle returns at time y_v, if there is material still available for that vehicle, the vehicle will start loading at this time. Otherwise, the vehicle will wait until the next wave of deliveries to the depot. Once again, the vehicle will either carry all of the material allotted to it or the maximum capacity of the vehicle. This will continue until no more material is available for that vehicle. It is important to note that this method does not require the vehicle to be full to begin a trip. To do so would lead to vehicles sitting at the depot while material is available, which would reduce slack (unless the quantity available is small and the delay until the next wave is short).

The material on a vehicle will be divided between the sites that the vehicle visits using their respective proportions of the total demand.

The pseudocode for the scheduling algorithm is as follows. The following notation is used.

t_{vj}: The time, in minutes, at which vehicle v begins trip j

f_k: Relative demand of site k

g_k: Relative proportion of vehicle delivery allocated to site k

R_{vj}: Quantity delivered on trip j by vehicle v

q_{vjk}: Quantity delivered to site k assigned by v on trip j

C: Vehicle capacity (units of material)

schedule(σ)

CALCULATE $f_1, ..., f_n$
CALCULATE $y_1, ..., y_V$
FOR $v = 1 : V$
    $J_v(t) = I(t) \sum_{k \epsilon \sigma_v} f_k$
    $t_{v1} = 0$
    $R_{v1} = \min\{J_v(0), C\}$
    $\theta = 2$
    REPEAT
        $t_{v\theta} = t_{v,\theta-1} + y_v$
        IF $J_v(t_{v\theta}) - \sum_{j=1}^{\theta-1} R_{vj} == 0$
            $t_{v,\theta} = \min t$ such that $J_v(t) - \sum_{j=1}^{\theta-1} R_{vj} > 0$
        END
        $R_{v\theta} = \min\{C, J_v(t_{v\theta}) - \sum_{j=1}^{\theta-1} R_{vj}\}$
        $\theta = \theta + 1$
    UNTIL $\sum_{j=1}^{\theta-1} R_{v,j} == (T_2 - T_1) \sum_{k \in \sigma_v} L_k$
    $r_v = \theta - 1$
    FOR $j = 1 : r_v$
        FOR $k \in \sigma_v$
            $g_k = \frac{f_k}{\sum_{i \in \sigma_v} f_i}$

```
            q_{vjk} = g_k R_{vj}
        END
    END
END
```

This scheduling algorithm produces the schedules in Table 1 for our example. The first column denotes the sites visited by that vehicle and each remaining column denotes a trip taken by that vehicle. The first row denotes the starting time in minutes for each trip. The solution is evaluated in Table 2 with a minimum slack of 478 minutes.

Tab. 1: Schedule for example.

Schedule for Vehicle 1

Site/Time	0	180	360
5	10,521	21,479	16,000
4	7,890	16,110	12,000

Schedule for Vehicle 2

Site/Time	0	180	360
3	13,151	26,849	20,000

Schedule for Vehicle 3

Site/Time	0	180	360
2	9,863	20,137	15,000
1	6,575	13,425	10,000

Tab. 2: Slack calculations for example.

			Slack		
Vehicle v	Site k	w_{jk}	s_{v1k}	s_{v2k}	s_{v3k}
1	5	45	555	507	595
	4	73	527	479	567
2	3	46	540	492	580
3	2	45	555	507	595
	1	74	526	478	566

6 Improvement

After a solution has been found, it may be possible to manipulate the quantities carried on each trip to increase the slack. Note that, as shown in Table 2, the slack at sites 2 and 1 in the second trip of Vehicle

3 are different. The slack at site 2 is larger than the slack at site 1 because Vehicle 3 visits that site before it visits site 1. If, in its first trip, Vehicle 3 delivered more material to site 1 (and less to site 2), the slacks could be the same, which would increase the minimum slack.

Because the slack for a delivery depends upon the material delivered to that site on previous trips, the goal of the Delivery Volume Improvement (DVI) algorithm is to adjust the delivery quantities on one route in such a way that the slacks for all sites on the next route are the same. The algorithm starts by setting the delivery quantities for the first trip and then proceeds to the next trip. Note that the trip start times and site delivery times are given and not changed by this algorithm.

Consider a vehicle v making a delivery to site k at in second trip (so $j = 2$). Let D_{vjk} be the time that this delivery occurs. We would like the slack of every delivery on this trip to be equal to K, which determines the delivery quantity during the first trip:

$$K = T_1 + \tfrac{q_{v1k}}{L_k} - D_{vjk}$$
$$q_{v1k} = (K + D_{vjk} - T_1)L_k$$

We want to find the largest possible K that is feasible with respect to the total material that the vehicle delivers on that trip. Let R_{vj} be the total material that vehicle v delivers on trip j. This is given and is not changed by the algorithm. Because the total of the delivery quantities in the first trip must equal R_{v1} and the delivery times equal w_k, then we can determine K and the delivery quantities as follows:

$$K = T_1 + \frac{R_{v1}}{\sum_{i \in \sigma_v} L_i} - \frac{\sum_{i \in \sigma_v} w_i L_i}{\sum_{i \in \sigma_v} L_i}$$
$$q_{v1k} = \frac{L_k}{\sum_{i \in \sigma_v} L_i}\left(R_{v1} - \sum_{i \in \sigma_v} w_i L_i\right) + w_k L_k$$

For subsequent trips, it easy to show that letting the delivery quantities be proportional to the site demands will suffice. Of course, it is important not to deliver more that a site needs, which affects the delivery quantities of the last trips.

It is important to note that if the minimum slack occurs in the first trip for any vehicle, then the procedure will not be able to increase the minimum slack. However, the procedure may increase the slack for deliveries on subsequent trips.

In the Delivery Volume Improvement (DVI) algorithm, let q' be the desired amount to deliver, and let DR_k be the remaining material needed at site k. Let s_v be the number of sites on route σ_v, and let $[i]$ be the i-th site on route σ_v.

DVI(σ)

FOR $v = 1 : V$

    FOR $k \in \sigma_v$

$DR_k = L_k(T_2 - T_1)$
END
FOR $j = 1 : r_v$
FOR $k = s_v, s_v - 1, ..., 1$
IF $R_{vj} < \sum_{i=1}^{k} DR_{[i]}$
IF $j == 1$
$q' = \frac{L_{[k]}}{\sum_{i=1}^{k} L_{[i]}}(R_{vj} - \sum_{i=1}^{k} w_{[i]} L_{[i]}) + w_{[k]} L_{[k]}$
ELSE
$q' = \frac{L_{[k]}}{\sum_{i=1}^{k} L_{[i]}} R_{vj}$
END
IF $q' < DR_{[k]}$
$q_{vj[k]} = q'$
$DR_{[k]} = DR_{[k]} - q'$
ELSE
$q_{vj[k]} = DR_{[k]}$
$DR_{[k]} = 0$
END
$R_{vj} = R_{vj} - q_{vj[k]}$
ELSE
$q_{vj[k]} = DR_{[k]}$
$DR_{[k]} = 0$
$R_{vj} = R_{vj} - q_{vj[k]}$
END
END
END
END
END

In the example previously introduced, the DVI algorithm updates the quantities and the slacks as shown in Tables 3 and 4. Note that, in the second and third trips, the slacks for sites 5 and 4 are the same and that the slacks for sites 2 and 1 are the same. (The agreement between vehicles 1 and 3 is a coincidence that reflects the similarity in delivery times.) The new minimum slack is 492 minutes.

If the vehicle trips are coordinated (for example, if every vehicle makes one trip for each wave), then we can apply a different type of DVI algorithm to all of the vehicles simultaneously and shift material from one vehicle to another to make all of the slacks the same [17]. Because the approach in this paper schedules each vehicle separately, it may yield a solution in which the number of trips per vehicle varies. Therefore, the DVI algorithm used here considers only one vehicle at a time.

7 Computational Results

We tested the solution approach on a set of instances in order to determine which heuristics generated the best solutions and to evaluate their relative computational effort. We considered both routing techniques

Tab. 3: Schedule for example with DVI.

Schedule for Vehicle 1

Site/Time	0	180	360
5	9,561	21,479	16,960
4	8,850	16,110	11,040

Schedule for Vehicle 2

Site/Time	0	180	360
3	13,151	26,849	20,000

Schedule for Vehicle 3

Site/Time	0	180	360
2	8,993	20,137	15,870
1	7,445	13,425	9,130

Tab. 4: Slack calculations for example with DVI.

			Slack		
Vehicle v	Site k	w_{jk}	s_{v1k}	s_{v2k}	s_{v3k}
1	5	45	555	495	583
	4	73	527	495	583
2	3	46	540	492	580
3	2	45	555	495	583
	1	74	526	495	583

(nearest neighbor and 2-opt) and both clustering objectives (by time and by demand). Combining these and the scheduling heuristic yielded four heuristics (which are essentially four variations of the same approach). A set of base instances was created and varied to capture different real-world scenarios. After obtaining solutions for each instance using the four heuristics, we recorded the minimum slack and computational effort.

7.1 Baseline Instances

To create the instances, we started with 17 baseline instances and varied them systematically to create 27 instances for each baseline. Therefore, we had a total of 459 instances. As shown in Table 1, the number of sites ranged from 2 to 199. The data for site location, site demand, depot location, and vehicle capacities were obtained from three sources: mass dispensing plans from Montgomery County, Maryland; California PODs from an example provided in the online routing software Toursolver; and the classical vehicle

Baseline instance	Number of sites	Number of vehicles	Number of waves
R1	5	3	3
F1	9	5	4
C1	9	5	5
M1	10	5	5
R3	10	5	4
R2	15	8	3
M2	15	8	3
C2	20	10	4
M3	50	25	5
V1	50	25	3
V2	75	38	5
V3	100	50	6
V4	150	75	7
V5	199	100	7
V11	120	60	5
V12	100	50	6
M4	189	71	4

Tab. 5: Summary of Baseline Instances

routing problems from Christofides [5]. For the Maryland and California sites (which had street addresses), Toursolver and Google Maps were used to calculate travel times between the sites. We invented demand and wave delivery information to be similar to real world mass dispensing plans from Maryland. All of the instances had loading times of 15 minutes. (The instances are available upon request.)

7.2 Constructing Instances

Given a baseline instance, we varied the number of vehicles, the average travel time, and the average demand. (Changes to the average demand also required corresponding changes to the amount delivered to the depot in each wave, though we did not change the timing of the waves.) We did not vary any other times because varying the travel times changes the loading/unloading times and the wave intervals relative to the travel times.

For each baseline instance, we set three values for the number of vehicles: the initial number V (shown in Table 5), $V - 0.2V$, and $V + 0.2V$. The last two values were rounded to the nearest integer. Large-demand instances were created by multiplying every site's demand by 3, and small-demand sites were created by dividing every site's demand by 3. Likewise, in the large-travel-time instances, all of the travel times were multiplied by 2; in the small-travel-time instances, all of the travel times were divided by 2. Thus, for each baseline instance, we generated 27 instances by combining the three values for the number of vehicles, the three sets of demands, and the three sets of travel times.

8 Results

This section reports on the results of using the four heuristics to generate solutions to the 459 instances that we constructed.

8.1 Computational Effort

As shown in Figure 3, the time required to generated solution increased as the number of sites increased and when the 2-opt routing heuristic was used. Other characteristics of the instances did not affect the computational effort. The choice of clustering objective did not affect the computational effort.

Running the 2-opt heuristic generally added 10 to 20 percent to the computational effort. The notable exception to this was the problem set M4 (which is not included in Figure 3). For these instances, with the 2-opt heuristic, the average time required was nearly 23 seconds. In all of the other problem sets, the sites surround a central depot. In the problem set M4, however, the depot is located outside of the region in which the sites lie. Thus, it appears that the nearest neighbor heuristic generates a poor route, for the 2-opt procedure spends a great deal of effort to improve the route.

8.2 Routing Heuristics Performance

The minimum slack in the best solutions found varied by problem set. Within a problem set, some combinations of number of vehicles, travel times, and demands had only solutions with low slack, while other combinations had solutions with much more slack. To compare the routing and clustering heuristics, we determined the average minimum slack of the solutions within a problem set and counted the number of times that each routing and clustering combination generated the best solution found. The results, shown in Table 6, show that clustering by duration generally generated better solutions.

The nearest neighbor procedure and the 2-opt procedure perform equally well. Table 7 shows that, when used with the cluster by duration objective, using the nearest neighbor procedure was slightly more likely to generate a better solution. In many cases, however, they generated equally good solutions. The performance of these heuristics was not affected by the relative length of the wave intervals. Changes in the demand often made no change in the quality of the solution, because the vehicles had sufficient capacity to carry the increase material. In some cases, increasing the site demands generated instances in which the vehicles

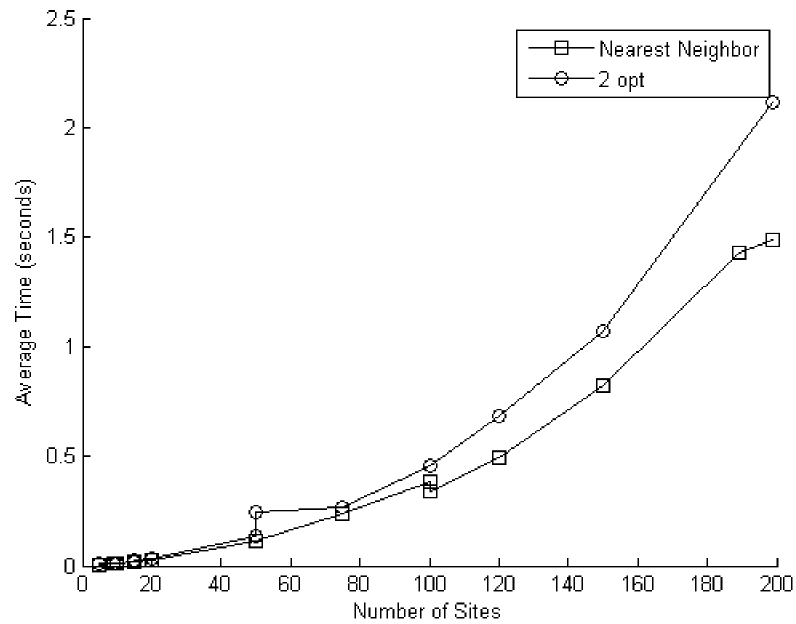

Fig. 3: Computational time for finding initial routes.

needed more trips to deliver the material; this naturally reduced the slack and led to poor-quality solutions. The relative performance of the heuristics did not change however.

8.3 DVI Performance

The same body of instances were used to test the DVI procedure. The solutions improved were those from the heuristics previously introduced. Table 8 shows the number of instances that are candidates for DVI. These are the instances in which the minimum slack does not occur on the first trip of a vehicle. By only looking at DVI results for these instances, we can find the average improvement seen in Table 9. This table also shows the average over the problem sets. The two cluster by demand heuristics, which gave the worst results without DVI, showed the greatest improvement.

To analyze the overall solution quality for the four heuristics with DVI, we consider the solutions for all 459 instances, including those where the minimum slack occurs in a first trip and DVI was not used. Table 10 shows the average minimum slack for all heuristics. The two cluster by duration heuristics were similar and outperformed the cluster by demand heuristics. The same is true for the number of best solutions found. Neither routing heuristic dominated the other.

Routing Heuristic	Average minimum slack				Number of best solutions found			
	Nearest neighbor		2-opt		Nearest neighbor		2-opt	
Clustering objective	Duration	Demand	Duration	Demand	Duration	Demand	Duration	Demand
R1	452.67	433.78	449.52	447.89	24	18	3	3
F1	1255.04	1252.07	1263.48	1250.30	11	6	12	7
C1	394.85	384.85	394.85	384.85	23	7	23	7
M1	566.22	552.04	566.22	552.04	27	12	27	12
R3	1035.00	1038.00	1048.78	1041.11	9	9	15	3
R2	-431.89	-513.52	-437.96	-513.52	25	14	24	14
M2	380.78	373.22	381.11	373.22	24	15	27	15
C2	120.67	110.74	103.81	100.04	21	19	24	18
M3	1189.33	1125.22	1177.56	1135.33	21	6	9	9
V1	590.67	574.89	599.67	593.22	6	0	12	12
V2	600.89	590.67	598.00	583.11	15	0	12	0
V3	473.11	459.89	470.56	450.78	18	0	12	3
V4	471.44	414.00	477.44	472.78	0	0	27	12
V5	470.67	462.22	481.67	474.67	9	3	18	9
V11	504.89	497.00	501.56	500.67	24	12	0	0
V12	492.67	476.78	491.56	474.89	24	15	18	3
M4	1254.07	1262.96	1232.04	1262.52	1	12	15	11
Average	577.71	558.52	576.46	563.76	16.6	8.7	16.4	8.1

Tab. 6: Quality of solutions generated by each heuristic on each problem set.

Baseline Instance	Nearest neighbor	Same	2-Opt
R1	24		3
F1	11		16
C1		27	
M1		27	
R3	12	9	6
R2		27	
M2		24	3
C2	3	18	6
M3	12	12	3
V1	6		21
V2	15		12
V3	15	3	9
V4			27
V5	9		18
V11	27		
V12	9	3	15
M4	13		14
Total	156	150	153
Percent	34.0%	32.7%	33.3%

Tab. 7: Number of instances in each problem set that the nearest neighbor and 2-opt routing procedures generated the better solution.

Routing Heuristic	Nearest neighbor		2-opt	
Clustering objective	Duration	Demand	Duration	Demand
R1	27	27	27	27
F1	6	6	3	3
C1	9	9	9	9
M1	3	3	3	3
R3	27	27	27	27
R2	27	27	27	27
M2	4	4	4	4
C2	1	1	1	1
M3	27	27	27	27
V1	0	0	0	0
V2	0	0	0	0
V3	0	0	0	0
V4	0	9	0	0
V5	0	0	0	0
V11	0	0	0	0
V12	3	3	3	3
M4	3	3	3	3

Tab. 8: Number of instances in which minimum slack does not occur on first wave.

Routing Heuristic	Nearest neighbor		2-opt		
Clustering objective	Duration	Demand	Duration	Demand	Problem Set Averages
R1	20.30	24.67	22.63	20.15	21.94
F1	18.00	49.50	36.00	65.67	39.44
C1	28.78	37.11	28.78	37.11	32.94
M1	51.33	63.67	51.33	63.67	57.50
R3	21.11	27.00	17.78	22.89	22.19
R2	25.96	49.07	26.37	49.07	37.62
M2	0.00	0.00	4.00	0.00	1.00
C2	52.00	52.00	92.00	92.00	72.00
M3	21.11	57.22	21.00	53.11	38.11
V1	-	-	-	-	-
V2	-	-	-	-	-
V3	-	-	-	-	-
V4	-	27.33	-	-	27.33
V5	-	-	-	-	-
V11	-	-	-	-	-
V12	2.00	26.00	2.00	26.00	14.00
M4	24.67	18.33	27.33	21.33	22.92
Average	24.11	35.99	29.92	41.00	32.25

Tab. 9: Average improvement for each heuristic.

	Average minimum slack				Number of best solutions found			
Routing Heuristic	Nearest neighbor		2-opt		Nearest neighbor		2-opt	
Clustering objective	Duration	Demand	Duration	Demand	Duration	Demand	Duration	Demand
R1	472.96	458.44	472.15	468.04	18	18	17	9
F1	1259.04	1263.07	1267.48	1257.59	11	6	12	7
C1	404.44	397.22	404.44	397.22	23	7	23	7
M1	571.93	559.11	571.93	559.11	27	12	27	12
R3	1056.11	1065.00	1066.56	1064.00	18	15	12	6
R2	-405.93	-464.44	-411.59	-464.44	25	14	24	14
M2	380.78	373.22	381.56	373.22	21	12	27	12
C2	122.59	112.67	107.22	103.44	21	19	24	18
M3	1210.44	1182.44	1198.56	1188.44	15	6	12	12
V1	590.67	574.89	599.67	593.22	6	0	12	12
V2	600.89	590.67	598.00	583.11	15	0	12	0
V3	473.11	459.89	470.56	450.78	18	0	12	3
V4	471.44	423.11	477.44	472.78	0	0	27	12
V5	470.67	462.22	481.67	474.67	9	3	18	9
V11	504.89	497.00	501.56	500.67	24	12	0	0
V12	492.89	479.67	491.78	477.78	24	15	18	3
M4	1256.81	1265.00	1235.07	1264.89	1	12	15	11
Average	584.34	570.54	583.18	574.38	16.2	8.9	17.2	8.6

Tab. 10: Quality of solutions generated by each heuristic with DVI on each problem set.

9 Comparison to Upper Bound

To find an upper bound on the optimal minimum slack, we relax the problem by ignoring the vehicle capacity and assuming that each site has a vehicle available to deliver material to that site at each wave. Thus, each site is visited once each wave, and the delivery at site k occurs $p_{n+1} + c_{n+1,k} + p_k$ time units after the wave is delivered to the depot. We use a version of the DVI algorithm to assign delivery quantities to each site so that the slacks of the deliveries in the same wave are equal. The pseudocode for setting the delivery quantities is given in the appendix. The slacks are calculated based on these delivery quantities and times.

We determined this upper bound for each instance and compared to the objective function values of the solutions found by the four heuristics with DVI. Table 11 compares the upper bounds to the results obtained from the Route by Nearest Neighbor and Cluster by Duration heuristic. This table presents the difference between the upper bound and heuristic solution for different groups of instances, classified by number of vehicles and time between waves. As the number of vehicles increases, the difference between the heuristic solution and the upper bound decreases. Also, as the time intervals between wave deliveries to the depot decrease, the difference decreases. The results for the other heuristics exhibit the same trend. From the results, it can be seen that, in many cases, the heuristics generate near-optimal solutions. For some instances, the gap between the upper bound and the heurstic solution is much larger because the heuristic generated a poor solution or the quality of the upper bound is poor.

Number of Vehicles	Wave Interval Size	Average Difference
Many	Large	53.35
Many	Average	22.84
Many	Small	17.00
Average	Large	66.65
Average	Average	32.37
Average	Small	23.12
Few	Large	155.49
Few	Average	67.41
Few	Small	50.02

Tab. 11: Average difference between upper bound and minimum slack of the Route by Nearest Neighbor Cluster by Duration solution for different sets of instances, classified by number of vehicles and time between waves.

10 Conclusions

This paper introduced the ISRP and presented a solution approach that separates the problem into two subproblems: routing and then scheduling. We solved the routing subproblem by using a route-first, cluster-second approach. We tried two different routing procedures and two clustering objectives. We solved the scheduling problem using a straightforward scheduling heuristic.

To test and compare the four routing and clustering heuristics, problem instances were created by systematically altering 17 baseline instances based on real-world data and classical VRP instances. Then solutions were obtained for each instance using all four heuristics and their results were compared for computational efficiency and quality of solution. It was found that using the 2-opt procedure increased the computational effort somewhat for most instances. Clustering by duration generated better solutions than clustering by demand. Using the nearest neighbor procedure to generate routes was neither better nor worse than using the 2-opt procedure. By adjusting the delivery quantities, the DVI algorithm improved the slack of many solutions but did not change the relative performance of the different heuristics. When comparing the solutions obtained from the heuristics to an upper bound, the heuristics performed well when an instance had sufficient vehicles and small time intervals between wave deliveries.

Future research includes creating heuristics for instances in which the vehicles have different capacities. Also, it is necessary to investigate when DVI can be performed on multiple vehicles simultaneously. More computationally expensive search techniques and column generation procedures may also be useful to construct high-quality solutions.

11 Acknowledgements

This publication was supported by award number 5H75TP000309-02 from the Centers for Disease Control and Prevention (CDC) to National Association of County and City Health Officials (NACCHO). Its contents are solely the responsibility of Montgomery County, Maryland Advanced Practice Center for Public Health Emergency Preparedness and Response and do not necessarily represent the official views of the CDC or NACCHO.

References

[1] Aaby, K., Abbey, R.L., Herrmann, J.W., Treadwell, M., Jordan, C.S., and Wood, K. (2006) Embracing computer modeling to address pandemic influenze in the 21st century. *Journal of Public Health Management and Practice*, 365-372.

[2] Aaby, K., Herrmann, J.W., Jordan, C.S., Treadwell, M., and Wood, K. (2006) Montgomery County's public health service uses operations research to plan emergency mass dispensing and vaccination clinics. *Interfaces*, **36.6**, 569-579.

[3] Baita, F., Ukovich, W., Pesenti, R., and Favaretto, D. (1998) Dynamic routing-and-inventory problems: A review. *Transportation Research*, **32**, 585-598.

[4] Ball, M.O., and Lin, F.L. (1993) A reliability model applied to emergency service vehicle location. *Operations Research*, **41**, 18-36.

[5] Beasley, J.E. OR-Library, (2009) Brunel University, 5 Aug. 2009, <http://people.brunel.ac.uk/ mastjjb/jeb/info.html>.

[6] Bramel, J., and Simchi-Levi, D. (1997) *The Logic of Logistics*, Springer, New York, NY.

[7] Brandeau, M.L., McCoy, J.H., Hupert, N., Holty, J.E., and Bravata, D.M. (2009) Recommendations for modeling disaster responses in public health and medicine: a position paper of the society for medical decision making. *Medical Decision Making*, 1-23.

[8] Campbell, A., Clarke, L., Kleywegt, A.J., and Savelsbergh, M.W.P. (1998) The inventory routing problem in *Fleet Management and Logistics*, Crainic, T.G., and Laporte, G. (eds), Kluwer Academic Publishers, Dordrecht, The Netherlands, Chapter 4.

[9] Campbell, A.M., Clarke, L.W., and Savelsbergh, M.W.P. (2001) *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

[10] Ceyhun, A., Selim, H., and Ozkarahan, I. (2007) A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers and Operations Research*, **34**, 705-726.

[11] Chen, Z.-L. (2008) Integrated production and outbound distribution scheduling: Review and extensions. To appear in Operations Research.

[12] Daskin, M.S., and Stern, E.H. (1981) Hierarchical objective set covering model for emergency medical service vehicle deployment. *Transportation Science*, **15**, 137-152.

[13] Dror, M., Ball, M., and Golden, B. (1985) Computational comparisons of algorithms for the inventory routing problem. *Annals of Operations Research*, **4**, 3-23.

[14] Gillett, B.E., and Miller, L.R. (1974) A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, **22.2**, 340-349.

[15] Haghani, A., Tian, Q., and Hu, H. (2004) Simulation model for real-time emergency vehicle dispatching and routing. *Transportation Research Record, no. 1882*, 176-183.

[16] Herrmann, J.W., Lu, S., and Schalliol, K. (2009) A routing and scheduling approach for planning medication distribution. *Proceedings of the 2009 Industrial Engineering Research Conference.*

[17] Herrmann, J.W., Lu, S., and Schalliol, K. (2009) Delivery volume improvement for planning medication distribution. *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics.*

[18] Hupert, N., Wattson, D., Cuomo, J., Hollingsworth, E., and Neukermans, K. (2009) Predicting hospital surge after a large-scale anthrax attack: a model-based analysis of CDC's cities readiness initiative prophylaxis recommendations. *Medical Decision Making*, **29**, 424-437.

[19] Lee, E.K., Chen, C.H., Pietz, F., and Benecke, B. (2009) Modeling and optimizing the public health infrastructure for emergency response. *Interfaces*, **3.5**, 476-490.

[20] Mertens, S. Improving Solutions. (1999) Otto-von-Guericke University, 21 July 2009, <http://www-e.uni-magdeburg.de/mertens/TSP/node3.html>.

[21] Moin, N.H., and Salhi, S. (2007) Inventory routing problems: A logistical overview. *Journal of the Operational Research Society*, **58**, 1185-1194.

[22] Sivanandan, R., Hobeika, A.G., Ardekani, S.A., and Lockwood, P.B. (1988) Heuristic shortest-path method for emergency vehicle assignment - a study on the Mexico City network. *Transportation Research Record, no. 1168.*

[23] Toth, P., and Vigo D. (2001), An overview of vehicle routing problems in *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1-26.

[24] Weintraub, A., Aboud, J., Fernandez, C., Laporte, G., and Ramirez, E. (1999) An emergency vehicle dispatching system for an electric utility in Chile. *Journal of the Operational Research Society*, **50**, 690-696.

12 Appendix

12.1 Pseudocode

The pseudocode for improvement by duration algorithm is as follows. During implementation, with one iteration it is important to examine all maximum and minimum vehicles instead of the first encountered. In this code, the following notation is necessary: l_v denotes the first site assigned to vehicle v, and m_v denotes the last site assigned to vehicle v.

improvebyduration(σ)

REPEAT

  CALCULATE $y_1, ...y_V$

  $F = \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$

  $v_{max} = argmax\{y_1, ..., y_V\}$

  $v_{min} = argmin\{y_1, ..., y_V\}$

  $\sigma'_{v_{max}-1} = \sigma_{v_{max}-1} \cup \{l_{v_{max}}\}$

  $\sigma'_{v_{max}} = \sigma_{v_{max}} \setminus \{l_{v_{max}}\}$

  CALCULATE $y'_1, ..., y'_V$

  IF $\max\{y'_1, ..., y'_V\} - \min\{y'_1, ..., y'_V\} < \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$

    $\sigma_{v_{max}-1} = \sigma'_{v_{max}-1}$

    $\sigma_{v_{max}} = \sigma'_{v_{max}}$

    $y_{v_{max}-1} = y'_{v_{max}-1}$

    $y_{v_{max}} = y'_{v_{max}}$

  END

  $\sigma'_{v_{max}+1} = \{m_{v_{max}}\} \cup \sigma_{v_{max}+1}$

  $\sigma'_{v_{max}} = \sigma_{v_{max}} \setminus \{m_{v_{max}}\}$

  CALCULATE $y'_1, ..., y'_V$

  IF $\max\{y'_1, ..., y'_V\} - \min\{y'_1, ..., y'_V\} < \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$

    $\sigma_{v_{max}} = \sigma'_{v_{max}}$

    $\sigma_{v_{max}+1} = \sigma'_{v_{max}+1}$

    $y_{v_{max}} = y'_{v_{max}}$

    $y_{v_{max}+1} = y'_{v_{max}+1}$

  END

  $\sigma'_{v_{min}} = \{m_{v_{min}-1}\} \cup \sigma_{v_{min}}$

  $\sigma'_{v_{min}-1} = \sigma_{v_{min}-1} \setminus \{m_{v_{min}-1}\}$

  CALCULATE $y'_1, ..., y'_V$

  IF $\max\{y'_1, ..., y'_V\} - \min\{y'_1, ..., y'_V\} < \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$

    $\sigma_{v_{min}-1} = \sigma'_{v_{min}-1}$

$\sigma_{v_{min}} = \sigma'_{v_{min}}$
$y_{v_{min}-1} = y'_{v_{min}-1}$
$y_{v_{min}} = y'_{v_{min}}$
END

$\sigma'_{v_{min}} = \sigma_{v_{min}} \cup \{l_{v_{min}+1}\}$
$\sigma'_{v_{min}+1} = \sigma_{v_{min}+1} \backslash \{l_{v_{min}+1}\}$

CALCULATE $y'_1, ..., y'_V$

IF $\max\{y'_1, ..., y'_V\} - \min\{y'_1, ..., y'_V\} < \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$
$\sigma_{v_{min}} = \sigma'_{v_{min}}$
$\sigma_{v_{min}+1} = \sigma'_{v_{min}+1}$
$y_{v_{min}} = y'_{v_{min}}$
$y_{v_{min}+1} = y'_{v_{min}+1}$
END

$F' = \max\{y_1, ..., y_V\} - \min\{y_1, ..., y_V\}$

UNTIL $F == F'$

To determine the upper bound for an instance, we use the following pseudocode to determine delivery quantities. In this code, Y and m are introduced as values used to find the desired slack. These values are calculated in lines 1-13 of the pseudocode. W_j refers to the amount of inventory made available to the depot in wave j. The value t_j refers to the time that the wave delivery j is made to the depot. Lines 15-37 of the pseudocode calculate the quantities, q, for each vehicle. After determining these delivery quantities we determine the minimum slack of the deliveries.

upperbound
1 Renumber the sites so that $w_1 \geq ... \geq w_n$
2
3 $Y_0 = 0$
4 **FOR** $h = 1, ..., n-1$
5 $m_h = \sum_{i=1}^{h} L_i$
6 $Y_h = Y_{h-1} + (w_h - w_{h+1})m_h$
7 **END**
8 $m_n = \sum_{i=1}^{n} L_i$
9 $Y_n = Y_{n-1} + (T_2 - w_1 - T_2 + w_n)m_n$
10 **FOR** $h = n+1, ..., 2n-1$
11 $m_h = \sum_{i=h-n+1}^{n} L_i$
12 $Y_h = Y_{h-1} + (w_{h-n} - w_{h-n+1})m_h$
13 **END**
14
15 **FOR** $j = 1, ..., r-1$
16 $Q = W_1 + ... + W_j$
17 **FIND** h such that $Y_{h-1} \leq Q < Y_h$
18 **IF** $h \leq n$

$K_j = T_1 - t_{j+1} - w_h + \frac{Q - Y_{h-1}}{m_h}$
FOR $k = 1, ..., h$
$q_{jk} = (K_j - T_1 + t_{j+1} + w_k)L_k - \sum_{i=1}^{j-1} q_{ik}$
END
FOR $k = h + 1, ..., n$
$q_{jk} = 0$
END
ELSE
$K_j = T_2 - t_{j+1} - w_{h-n} + \frac{Q - Y_{h-1}}{m_h}$
FOR $k = 1, ..., h - n$
$q_{jk} = L_k(T_2 - T_1) - \sum_{i=1}^{j-1} q_{ik}$
END
FOR $k = h - n + 1, ..., n$
$q_{jk} = (K_j - T_1 + t_{j+1} + w_k)L_k - \sum_{i=1}^{j-1}$
END
END
END
FOR $k = 1, ..., n$
$q_{rk} = L_k(T_2 - T_1) - \sum_{i=1}^{r-1} q_{ik}$
END

12.2 Example Data

The following data is used for the examples described in this paper. Note that there are three waves (deliveries to the depot), as shown in the following table.

$T_1 = 600$ minutes

$T_2 = 1200$ minutes

$I(0) = 48,000$ units

$I(180) = 146,000$ units

$I(360) = 219,000$ units

$V = 3$ vehicles

$C = 112,000$ units per vehicle

$n = 5$ PODs

$L = \begin{pmatrix} 50 & 75 & 100 & 60 & 80 \end{pmatrix}$ units per minute

$p = 15$ minutes

$$c = \begin{pmatrix} 0 & 14 & 39 & 29 & 26 & 17 \\ 14 & 0 & 34 & 24 & 21 & 15 \\ 39 & 34 & 0 & 17 & 16 & 30 \\ 29 & 24 & 17 & 0 & 13 & 17 \\ 26 & 21 & 16 & 13 & 0 & 15 \\ 17 & 15 & 30 & 17 & 15 & 0 \end{pmatrix} \text{ with all times in minutes}$$

Wave	1	2	3
Time (minutes)	0	180	360
Quantity	48,000	98,000	73,000

Tab. 12: Deliveries to the depot.