Proceedings of the 2010 Industrial Engineering Research Conference
A. Johnson and J. Miller, eds.

Adaptive Large Neighborhood Search for the Inventory Slack Routing Problem

Abstract ID: 310

Adam Montjoy
Jeffrey W. Herrmann
Institute for Systems Research
University of Maryland, College Park, MD 20742, USA

Abstract

The inventory slack routing problem (ISRP) addresses a critical issue in emergency preparedness. Public health officials must develop plans for distributing medication to points of dispensing, which give medication to the public in case of a bioterrorist attack. Unlike other vehicle routing problems, which use a cost objective function, the objective in the ISRP is to maximize the slack in the deliveries so that sites can operate without interruption if disruptions occur. This paper presents an adaptive large neighborhood search and compares its performance to other routing and scheduling approaches on a set of problem instances.

Keywords
Emergency Preparedness, Scheduling, Vehicle Routing, Logistics

1. Background

Some public health emergencies would require the quick and efficient distribution of medication and supplies to a large number of people to prevent a catastrophe. For instance, the widespread release of anthrax in a metropolitan area could result in casualties equivalent to that of a small nuclear explosion if antibiotics are not distributed quickly [4]. 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 points of dispensing (PODs) from a central location. 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 to 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 [4]. Clearly, there are many uncertainties in medication distribution. For this reason, planners need a robust plan that calls for delivering medication to PODs much earlier than it is needed. This improves the likelihood that the PODs will open on-time and will not run out of medication during operations.

Much research has been done to develop models to improve emergency preparedness planning. Hupert *et al.* [4] modeled the hospital surge after a large-scale anthrax attack and emphasized 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 [1, 2, 5].

The Inventory Slack Routing Problem (ISRP) that we present is similar to the Vehicle Routing Problem (VRP) [6] and Inventory Routing Problem (IRP) [3]. 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. In addition, the limited availability of medication at the depot adds an additional constraint to the problem. Finally, to hedge against uncertainties, the goal is to increase the interval (or slack) between each delivery and the time that POD would exhaust its supplies (not minimize cost). In particular, the objective function of the ISRP is to maximize the minimum slack.

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 operation. 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. 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. Therefore, we assume that each and every site is assigned to exactly one vehicle, and each vehicle always follows the same route to visit the assigned sites.

2.1 Notation

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, $v = 1,...,V$
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 $I(t)$.

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:

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

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

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

All delivery quantities are non-negative. Each vehicle has a fixed capacity and can carry a maximum of C units, that is $\sum_{k \in \sigma_{vj}} q_{vjk} \leq C$ 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 $\sum_{j=1}^{r_v} q_{vjk} = (T_2 - T_1)L_k$ for $v = 1,...,V$

and $k \in \sigma_v$.

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:

$$w_{vk} = p_{n+1} + c_{n+1,[1]} + p_{[1]} + c_{[1],[2]} + ... + p_k \tag{3}$$

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:

$$Q_{vjk} = \sum_{i=1}^{j-1} q_{vik} \tag{4}$$

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:

$$s_{vjk} = T_1 + \frac{Q_{vjk}}{L_k} - (t_{vj} + w_{vk}) \tag{5}$$

The evaluation of a solution is the minimum slack over all vehicles, sites, and trips: $S = \min\{s_{vjk}\}$.

3. Solution Approach

The ISRP, like other versions of the VRP and IRP, is NP-hard, so obtaining an exact solution is computationally expensive. In this paper, we will study the solutions generated by an Adaptive Large Neighborhood Search (ALNS). We will adapt the procedure used by Pisinger and Ropke [9] to solve vehicle routing problems to assign sites to vehicles and sequence those routes. We will then use a scheduling algorithm to determine the trip start times and delivery quantities.

The ALNS begins with an initial solution and iteratively destroys and rebuilds the solution by randomly choosing and applying a number of quick heuristics. Associated with each heuristic is a weight that determines its selection probability. With each iteration, the new solution is either accepted or rejected and the heuristic weights are updated according to their performance.

3.1 Removal Heuristics

Our ALNS uses four removal heuristics that are appropriate for the ISRP and have the ability to diversify the search. These removal heuristics take a complete sequence of sites for each vehicle, relax a specified number of sites, and output a partial solution.

Random Removal. One of the goals of the ALNS is to avoid exclusively searching around local critical points in favor of searching for a global best solution. This heuristic removes a fixed number of randomly selected sites.

Worst Removal. This heuristic removes sites that have the smallest slack on one or more of their trips. The minimum slack is calculated for each site, the sites are reordered from worst to best, and a site to remove is chosen based on a random number and a parameter. A random number r is drawn and the site relaxed is r^p through the ranking of worst to best slacks. The larger p is, the probability is higher that sites with the smallest slack will be chosen. This is done in order to avoid the same sites being removed repeatedly. This algorithm is repeated until a desired number of sites are removed.

Related Removal. The related removal heuristic was proposed by Paul Shaw [8] as the sole heuristic for a large neighborhood search. The motivation for removing sites that are related to one another is that, when the removed sites are drastically different, often they are inserted in the same place. Removing similar sites provides better opportunities to generate a new solution. Sites are considered related if they are geographically close and have similar dispensing rates. The relatedness between sites i and j is defined as follows:

$$R(i, j) = \alpha c_{ij} + \delta |L_i - L_j| \tag{6}$$

The constants α and δ depend on the magnitudes of the travel times and dispensing rates, as well as their importance relative to the other. When R is smaller, the two sites are more related. This heuristic operates by first randomly selecting a site to remove. Then it randomly chooses another site, where the most related sites are more likely to be selected (the procedure is similar to that used in the the Worst Removal heuristic). This continues until the desired number of sites have been removed.

Longest Travel Time Removal. Sites with large travel times tend to have smaller slacks. Thus, this heuristic aims to remove sites that have a long trip to deliver inventory. The heuristic is the similar to the Worst Removal heuristic but ranks the sites by their travel time.

3.2 Insertion Heuristics

The insertion heuristics take a partial solution and insert the removed sites to form a complete solution. When a partial solution is presented to an insertion heuristic, the first step is to ensure that each vehicle has at least one site to service. This rule could be implemented in the removal heuristics to prohibit removing a site if it is the only site on a vehicle's route.

Random Insertion. Similar to the random removal heuristic, this heuristic serves the purpose of diversifying the search. This heuristic iteratively places a removed site randomly within a vehicle's sequence on each pass until all sites have been placed.

Best Position of Vehicle with Lowest Travel Time Insertion. For each removed site, this heuristic identifies the vehicle with the smallest travel time, considers each possible position in its route, and inserts the site in the position that yields the highest minimum slack. Two different procedures are used to sequence the sites for consideration. In the first procedure, the sites first inserted were those to be first relaxed, which with the worst and longest travel removal heuristics, entails inserting the "worst" sites first. The second procedure inserts the sites backwards such that the "worst" sites are inserted last. The motivation for this is that it may sometimes be of benefit to place difficult sites first while other times it may be best to insert them last.

Best Position of Vehicle with Lowest Total Demand Insertion. This method is similar to the previous insertion technique. It selects the vehicle with the lowest total demand. This method also has two different procedures for the order in which sites are inserted into the solution.

Nearest Neighbor Insertion. Upon each iteration for this heuristic, the relaxed site is placed before or after the site geographically closest already in the partial solution according to which yields the higher minimum slack for the vehicle. This move operation is motivated by influencing vehicles to visit sites closely located.

3.3 Selecting Heuristics and Accepting Solutions

Like the procedure of Pisinger and Ropke [9], our ALNS selects a removal heuristic and an insertion heuristic each iteration. A heuristic's selection probability is proportional to its weight. Both selected heuristics are rewarded in three cases: (1) a new global best solution is found, (2) the new solution is better than the previous one and has not been accepted before, and (3) the new solution is not better than the previous one and it has not been accepted before. If rewarded, the heuristics' observed weights are increased by 5 (in case 1), 3 (in case 2), or 1 (in case 3). The search process is divided into segments of 50 iterations. At the beginning of the segment, each heuristic has an observed weight of zero. At the end of any segment, the ALNS calculates new weights based on the weights from the previous segment and the observed weights.

Our ALNS uses a simulated annealing procedure to determine if a new solution is accepted. The probability to accept a solution x' (given a current solution x) is given by $\min\{1, e^{\frac{f(x')-f(x)}{T}}\}$ where $T > 0$ is the temperature, which is updated according to a cooling rate such that $T = Tc$, where $0 < c < 1$. An ALNS can be adapted to utilize various acceptance methods such as simple rejection of poorer solutions or Tabu searches.

3.4 Parameter Selection

The number of sites removed in each iteration was kept small because moving a small number of sites can significantly affect the minimum slack. We set the selection parameter as $p = 5$ for the Worst Removal and Longest Travel Time

Removal heuristics in order to focus on the "worst" sites. The α and δ parameters of the Related Removal heuristic were chosen to scale the magnitudes of the values and put more weight on travel time.

The starting temperature of the simulated annealing procedure was selected by observing the objective value of an initial solution and choosing by a desired probability for a relatively lesser value. The cooling rate was then tuned to have reasonable acceptance probabilities towards the end of the search.

4. Scheduling

After constructing routes for the vehicles, it is necessary to determine the times at which each vehicle will begin its routes and the quantity to be delivered to each site on each route. We create the schedule using the following policy (described in detail in [7]). Each vehicle is allocated a portion of each wave that is proportional to the total demand of the sites that it visits. Each vehicle will begin loading for its first route as soon as the depot has material and will carry as much material as it can. When the vehicle returns to the depot, 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. Because the slack for a delivery depends upon the material delivered to that site on previous trips, we set the delivery quantities on one route in such a way that the slacks for all sites on the next route are the same. Our algorithm finds the largest possible slack that is feasible and uses that to determine the delivery quantities for each site on the first trip. For subsequent trips, 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.

5. Computation Results

We tested the approach on four sets of instances. Each set has 27 variants of a baseline instance. The sets of instances have 10, 15, 50, and 100 sites. We ran five replications of the ALNS on each instance (with 1000 iterations per replication) and averaged the min slack of these solutions. (To construct an initial solution, we used the nearest neighbor heuristic to generate a large route and then split the route into routes for each vehicle.) These results were compared to the min slack of solutions from an existing routing-and-scheduling heuristic (RSH) and an upper bound that we developed previously [7]. For each instance, the min slack of each solution was divided by the upper bound for that instance. For the RSH and the ALNS, Table 1 shows the average ratio over the instances in each problem set. Table 1 also shows the average time required to run the ALNS.

The results show that, on average, the ALNS finds solutions with more slack than those that the routing-and-scheduling heuristic constructs. On six instances, the ALNS found an optimal solution. The relative quality of the ALNS solutions decreases with the 100-site instances. For twelve instances in this set, the average min slack of the ALNS solutions was less than that of the routing-and-scheduling heuristic. The weights of the various removal and insertion heuristics fluctuated during the course of each search; no single heuristic dominated the others.

The average computation time generally increases as the number of sites increases, as shown. For the problem sets with 10 and 15 sites, increasing the number of vehicles reduced the average computational effort. For the other problem sets, increasing the number of vehicles had little impact on computational effort.

Table 1: Results by problem set

Number of Sites	RSH	ALNS	Average Computation Time (secs)
10	0.900	0.937	67.87
15	0.925	0.951	53.87
50	0.943	0.960	188.00
100	0.925	0.936	406.10

6. Conclusions

This paper formulated the ISRP problem and described an adaptive large neighborhood search for finding good solutions. Experimental results show that, on average, the ALNS finds solutions with more slack than those that the routing-and-scheduling heuristic constructs and that the average computation time generally increases as the number of sites increases. Future work will extend these results to a larger set of problem instances in order to understand better the performance of the search algorithm and the removal and insertion heuristics.

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

[4] 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.

[5] 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.

[6] 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.

[7] Montjoy, A., Brown, S., and Herrmann, J.W. (2009) Solving the Inventory Slack Routing Problem. Technical Report, Institute for Systems Research, University of Maryland, <http://www.lib.umd.edu/drum/handle/1903/9432>.

[8] Shaw, P., (1997) A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems. Technical Report, Department of Computer Science, University of Strathclyde.

[9] Ropke, P., and Pisinger, D., (2005) An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, **40.4**, 455-472.