APPLICATION OF TECHNIQUES FROM THE VEHICLE ROUTING PROBLEM WITH SPLIT DELIVERIES AND TIME WINDOWS TO THE MILITARY INVENTORY ROUTING PROBLEM WITH MULTIPLE-CUSTOMER ROUTES

Marcus McNabb, Air Force Institute of Technology, 2950 Hobson Way, Wright Patterson AFB, OH 45433, 505-853-7143, marcus.mcnabb@us.af.mil
Jeffery Weir, Air Force Institute of Technology, 2950 Hobson Way, Wright Patterson AFB, OH 45433, 937-255-3636 ext.4523, jeffery.weir@afit.edu
Matthew "JD" Robbins, , Air Force Institute of Technology, 2950 Hobson Way, Wright Patterson AFB, OH 45433, 937-255-3636 ext.4539, matthew.robbins@afit.edu

ABSTRACT

The inventory routing problem consists of aspects of both a vehicle routing problem and inventory management. This paper considers such a problem with stochastic supply and deterministic demand. This variant is of particular interest in a military context because the stochastic supply represents the uncertainty of a vehicle reaching its destination(s). This paper focuses on solutions to the routing portion of the problem using ant colony optimization coupled with a strong local search. A suite of test problems is proposed and preliminary results for one problem are presented.

INTRODUCTION

The inventory routing problem (IRP) is a combination of the vehicle routing problem (VRP) and inventory management. McCormack [1] proposes a military inventory routing problem (MILIRP) in which the delivery vehicles operate in a hostile environment and the risk of loss of vehicles must be taken into account. This is a novel approach to the IRP untouched by the larger research community. McCormack proposes a direct delivery model of vehicle routing. This research will expand McCormack's work by incorporating a vehicle routing metaheuristic that allows for multiple customers per route.

LITERATURE REVIEW

An IRP solution identifies which customers are scheduled for a delivery during the current time period, how much supply will be carried to each of those customers, routes for each vehicle, and a delivery schedule for each route [2]. Formulations of the IRP vary and can be defined by its characteristics. Coelho et al. list seven such characteristics by which IRP instances may be classified [2]:

- Time horizon. The time horizon considered by the problem may be finite or infinite.
- Structure. The problem may contain one depot and one customer (one-to-one) one depot and many customers (one-to-many), or many depots and many customers (many-to-many).
- Routing. Possibilities are direct in the case of only direct deliveries, multiple in the case that vehicles visit multiple customers, or continuous in the case that no central depot exists.
- Inventory policy. The policy governing the customers' inventory levels may be a maximum level in which the customer has a maximum capacity that may not be exceeded or an order-up-to level in which a delivery to a customer always contains the quantity required to fill the customer's inventory.

- Inventory decisions. In some cases, inventories may be allowed to go into the negative, resulting in a shortage. Shortages may be modeled as back-orders, in which case the shortage is fulfilled in a later time period, or lost sales, in which case the shortage is not filled. Alternatively, the inventory may be restricted to be non-negative. In this case, if the inventory reaches zero, a direct delivery is made to the customer at the expense of a cost penalty.
- Fleet composition. The vehicle fleet may be homogeneous or heterogeneous.
- Fleet size. The fleet may consist of a single vehicle, multiple vehicles, or be unconstrained.

Coelho et al. [2] also use the demand properties to distinguish between problem types. Specifically, if the demand is known *a priori* for each time period, then it is deterministic. Alternatively, the demand may be unknown, in which case it is generally modeled using a probability distribution.

McCormack [1] defines the MILIRP with the following characteristics: finite time horizon, one-to-many structure, direct delivery routing, maximum level inventory policy, non-negative inventory decisions, homogeneous fleet composition, and multiple vehicles fleet size. However, the second nomenclature offered by Coelho et al. does not adequately describe the MILIRP. The MILIRP models the customer demand deterministically but Coelho et al. [2] assume a deterministic supply whereas the MILIRP assumes stochastic supply. This is because the MILIRP takes hostile threats to the vehicles into account, meaning a vehicle dispatched on a route may reach none, some, or all of the customers. Therefore, the supply is stochastic based upon the probability of success associated with a vehicle's route. McCormack is the first to entertain the notion of stochastic supply within the context of an IRP. Mu et al. [3] consider vehicle breakdowns for a VRP but only allot one spare vehicle because breakdowns rarely occur.

Kleywegt et al. [4] offer a dynamic programming approximation scheme for the IRP with stochastic demand that serves as a model for McCormack's work. One weakness to their algorithm is all of the routing problems are solved in a pre-processing stage, meaning all combinations of deliveries must be solved. Then, within the dynamic program, the optimal set of routes for any set of inputs is already given. This approach is not feasible for larger problems due to the fact that the VRP is NP-hard [5]. Kleywegt et al. alleviate this concern by restricting routes to no more than three customers. This research extends the work of McCormack [1] and Kleywegt et al. [4] by using a multiple customer routing concept (vs. direct deliveries for McCormack) wherein the routing solutions are generated during the problem solving process (vs. during the preprocessing stage for Kleywegt et al.) with no explicit limit on the number of customers per route. See [2] for a detailed review of the IRP.

PERTINENT REVIEW OF MCCORMACK'S MODEL

To model the risk to the vehicles, McCormack [1] proposes the imposition of a hex grid onto the geography of the problem wherein each hex is assigned a threat level (high or low). The threat map is assumed static within each time period but may change between time periods. Transitions between hexes are assigned a probability based on the threat level of the current hex and the hex into which the vehicle is moving. McCormack then uses Dijkstra's shortest path algorithm [6] to construct paths between each customer and the depot in which the shortest path is the path with the maximum probability of survival. McCormack then employs a dynamic programming approach to solve the MILIRP with direct deliveries. These are the pertinent details of the MILIRP necessary to develop a routing metaheuristic to embed within the MILIRP. See [1] for more details on McCormack's formulation and direct delivery solution method for the MILIRP.

ROUTING METAHEURISTIC FOR THE MILIRP

This research incorporates a routing metaheuristic to change the delivery model from direct to multiple. The metaheuristic used is an ant colony optimization (ACO) metaheuristic based on the results from Chapters III and IV. In those chapters, the authors showed the Max-Min Ant System variant of ACO coupled with 2-opt*, Or-opt, and Relocate local search operators offers strong solutions to the VRP with time windows and split deliveries (SDVRPTW). This algorithm is now adapted to the MILIRP. Following in the nomenclature of McCormack [1], customers are combat outposts (COP), the depot is a brigade support battalion (BSB), and the vehicles are cargo unmanned aerial systems (CUAS).

First, a subset of COPs is defined for whom split deliveries are not allowed. This consideration is a military one where the risk of loss of the CUAS is substantial. Therefore, based on the threat map described above, a threat vector for the set of COPs designates each COP as residing in either a high or low threat environment. Those COPs in a high threat environment will not accept a split delivery, while those COPs in a low threat environment will accept a split delivery. Hence, the threat vector is an *n*-dimensional vector, where *n* is the number of COPs and the i^{th} element is 1(0) if the i^{th} COP is in a high (low) threat environment.

The inputs to the routing algorithm are a distance matrix, a probability matrix, a threat vector, the number of CUAS allotted, the current inventory level of each COP, and a theta vector. The distance matrix is a matrix with the distance from each node (COPs and BSB) to every other node. These distances are the physical distances associated with the path of highest survivability between two nodes. The probability matrix denotes the probability of survival for each of the paths in the distance matrix. The threat vector is described above. The number of CUAS is bounded above by some maximum number of available crews or vehicles. The current inventory levels of the COPs are the levels of inventory at the current time step. The theta vector defines the parameters for the value curve, described in greater detail below.

Note several usual inputs to the vehicle routing problem with time windows (VRPTW), such as demand, service times, and time windows, are missing from the list of inputs used here. Service times are fixed at zero because the model emulated uses a drop system. In this particular research, the time windows are defined as the entire current time interval, *t*, in the interest of run time concerns and simplicity.

This algorithm also introduces several aspects not typically present on traditional VRP models. First, this algorithm introduces a distance limit because the CUAS have a limited range. Therefore, in both the solution construction and improvement phases, each route is restricted by this limit.

The biggest difference between the MILIRP and a traditional VRP is the demands are not fixed in the MILIRP. In a traditional VRP, the demands are fixed and a feasible solution must satisfy those demands. However, in addition to solving a routing problem, the routing portion of this algorithm must also determine what the demand of each COP should be. This differs from the concept of stochastic demand because the demand for each COP is fixed within the current time step, *t*, but the set of deliveries chosen for that time step may satisfy all, part, or none of this demand.
Within the dynamic program, a set of thetas, $[\theta_1, \theta_2, \theta_3]$, is developed iteratively. This set of three coefficients defines a quadratic function as $f(x) = \theta_1 + \theta_2 x + \theta_3 x^2$ which in turn defines the value of deliveries to the COPs. The thetas are constant between the COPs but may change between time steps. The input to the quadratic function is a COP inventory level and the output is the value of that inventory

level. In general, the quadratic function will be either a strictly increasing function or an increasing function on some interval $[0, x]$ where $x <$ COP capacity. In this case, the function reaches a maximum at inventory level equal to x and transitions to decreasing on the interval [x, COP capacity]. The reason for this second scenario is because while delivering any amount to a COP may be beneficial when viewed as an individual delivery, it may be detrimental in the overall scheme because of the risk incurred with deliveries. More specifically, the risk may outweigh the reward of making a small delivery to a well-stocked COP, yielding a decrease in the overall value of the solution when including that delivery.

Given this set of thetas, the concept of the value of a delivery, route, and solution can be defined. The value of a delivery is simply the delta between the current value and the future value (i.e., the value of that COP's inventory if the delivery is made). The value of a route is simply the sum of the values of each delivery. The value of a total solution is slightly more complicated because, in the case of a split delivery, the values are not additive because the value curve is non-linear. Therefore, the total delivery amount to each COP from all routes is calculated and then the future value of each COP's inventory is calculated in the same fashion as for a single delivery. The value of the total solution is then the sum of these future COP inventory values.

Now, given these thetas and the definition of the values above, an initial demand is defined. This initial demand is calculated based on the delta between a current inventory value and inventory level at which the maximum value is achieved. If the delta is negative—meaning the COP's inventory level is above the level at which the maximum value occurs—then the delta is defined as zero. These deltas are then divided by the sum of all the deltas to give a demand for each COP that is a proportion of the total demand. This proportional demand is then multiplied by the product of the number of vehicles and vehicle capacity to define an initial demand. The demands are then rounded down and each demand is checked against the COP's maximum capacity. If any demand fulfillment would exceed the COP's maximum capacity, then that COP's demand is set to the difference between the current inventory and COP capacity.

The dynamic program also requires routing solutions for any number of CUAS up to the maximum bound given as an input parameter to the routing subroutine. Therefore, the routing algorithm returns solutions for one CUAS, two CUAS, and so forth. A zero CUAS solution is defined as having a value of zero.

The goal of the routing portion is to return a solution with both a high value and secure routes. The algorithm uses a lexicographic ordering of these two priorities with solution value being more important than routing security. Therefore, the ACO metaheuristic is modified thusly: first, solutions are constructed based on value. When the ants choose a COP to add to the current route, high value COPs (i.e., those lower in inventory) are more attractive than low value or higher inventory COPs. Compare this to a traditional ACO in which a physical distance metric constitutes the heuristic information and geographically closer COPs are more attractive. This algorithm uses these values as the heuristic information and combines this with a pheromone matrix in the same manner as a traditional ACO metaheuristic.

Next, the LS attempts to improve the value of the solution. This implementation of the ACO does not explicitly restrict the number of vehicles. Instead, it constructs as many routes as necessary to satisfy all of the COPs' demands and then chooses the highest value routes. The initial solution is partitioned into two sets: a set of "good" routes and a set of "bad" routes. The good routes are simply the first m routes where *m* is the number of CUAS allotted in this particular solution step. The 2-opt* and Or-opt operators then attempt to swap deliveries between the two partitions in an attempt to increase the overall value of the good routes. The total vehicle loads on the bad routes are ignored here to avoid excluding a valuable delivery only because the delivery or deliveries it is replacing in the good route do not fit onto the bad route.

In this phase, the Relocate operator cannot increase the value of the solution because it is an intra-route operator but it is included in the current LS phase because it is possible the 2-opt* and Or-opt operators may be able to further increase the value of the solution by allowing the Relocate operator to rearrange deliveries within a route. These LS operators are run iteratively and in a greedy fashion until the total value of the solution is at a local optimum. The traditional implementations of the LS operators are then applied to only the good routes portion of the solution in order to improve the security of those good routes. This is accomplished by using the probability matrix as an input instead of the distance matrix. Therefore, the "shorter" routes are those with higher probabilities of survival. See Chapter III for details on the LS implementation.

The metaheuristic iterates through these steps for some number of pre-defined iterations. The solution given as an output is for the initial demand as defined above. The next step is to alter this demand in an attempt to allow for a solution of greater value. Therefore, the initial demand is altered in the following way: a COP is randomly chosen and its demand is randomly incremented or decremented by one unit. Again, no demands are allowed to be negative and the sum of a COP's demand and its current inventory must not exceed the COP's capacity. The ACO metaheuristic is then applied to this new demand. If the total value of the solution returned by the routing metaheuristic is higher than that of the previous solution, or if the total value is equal to that of the previous solution but requires fewer vehicles, this solution is accepted and the demand alteration step is repeated. Otherwise, another COP is selected and its demand is randomly perturbed. This method is similar to that of Kleywegt et al. [4] except they use a best improvement schema. In other words, they explore each possible demand change and choose the change yielding the greatest improvement in value, repeating until no further improvement is possible. The size of the problem under investigation here yields this method highly impractical so instead a first improvement schema is implemented in which the first improving solution is accepted. This demand alteration loop continues until some pre-defined bound on non-improving iterations is reached, after which the highest value solution found so far is accepted. See Figure 1 for the pseudocode for the routing metaheuristic.

Pseudocode for routing subroutine

```
1: For i = 1 : number of CUAS allotted
        1: Define initial demand
        2: Define initial solution with value = 0
        3: While counter < consecutive non-improving solutions limit
                1: Implement ACO metaheuristic
                        1: Construct solutions based on value
                        2: Improve solutions based on value
                        3: Improve solutions based on probabilities of survival
                2: Compare solution to previous best
                        1: If higher value, accept new solution and reset counter to 0
                        2: If not higher value, increment counter
                3: Alter demand
```

Figure 1: Routing metaheuristic pseudocode

DEVELOPING TEST PROBLEMS

Testing this algorithm required the generation of a set of test problems. Real-world examples are not available in great enough numbers to sufficiently test the algorithm's performance and no current test set incorporates the threat map aspect. Furthermore, a real-world threat map presents security concerns. Therefore, a new set of notional test problems is generated. This new test set is modeled after the well-known and oft-used set of test problems for the VRPTW generated by Solomon [7]. The test set also expands upon McCormack's [1] use of a hex grid. The hexes are fixed at a size of two for the test set, meaning the distance from the center of a hex to the middle point on an edge is two.

In his test set, Solomon [7] uses customer sets with random, clustered, and random-clustered geographical orientations. Random orientation means customers are randomly scattered throughout the area of interest. In the clustered set, subsets of customers are grouped together. The random-clustered set is a mixture of these two, with some customers grouped together and others randomly spread throughout the area of interest. This test set emulates this geographical relationship with three customer sets, one of each type, with one minor change. In the problem of interest, customers who are very close to the depot are easily resupplied using ground transportation. Therefore, all of the customers in this test set are a minimum of two hexes away from the depot. The clustered data set groups the customers into four groups while the random-clustered set uses two groups. The number of customers is fixed at 36 because this is the approximate size of the real-world problems of interest. Similarly, the vehicle capacity is fixed at 8000 lbs with delivery increments of 500 lbs because this also reasonably approximates the real-world situation. Customer demands are random on the interval of [4000, 8000], meaning the test problems assume all of the customers' inventories are at least half full at the beginning of the problem.

The main thrust of the problem generation is in the generation of the threat maps. Using Solomon's ideas as a basis, three types of threat maps are developed: random, clustered, and random-clustered. Five distinct instantiations of each threat map are developed and used in conjunction with each of the instantiations of the customer locations, yielding a total of 45 test problems. The final parameter is in deciding the number of high threat hexes. In this problem set, the distance from the center of a hex to the center of any side is two kilometers with a 26x26 hex grid for a total of 676 hexes. Given this size, subject matter experts indicate approximately 10% of the hexes, or 68 hexes, should be high threat. For the random maps, this is accomplished simply by choosing 68 of the hexes to be high threat. For the clustered set, seven hexes are randomly chosen as high threat. Then, each adjacent hex (hexes one step from the original) is assigned a threat level with an 80% chance of being high threat. Each semi-adjacent hex (hexes two steps from the original) is assigned a threat level with a 40% chance of being high threat. To produce the desired parameter of 68 high threat hexes, if the number is too small, then an appropriate number of adjacent and semi-adjacent hexes are chosen to augment the original data with the adjacent hexes having twice the likelihood of selection compared with the semi-adjacent hexes. Similarly, if the number of high threat hexes is greater than 68, then the appropriate number of adjacent or semi-adjacent hexes are converted to low threat with the semi-adjacent hexes now having twice the likelihood of being converted. Combining these two for the random-clustered data set, the clustered hexes are assigned as above but with only three seed hexes as opposed to seven. Of the remaining low threat hexes, the appropriate number are randomly assigned as high threat such that the total number of high threat hexes is 68. For each of these data sets, the parameters are easily adjusted, allowing for generation of new threat maps to account for varying conditions such as an overall higher or lower threat level or for stronger or weaker clustering of the threats. See Appendix E for the associated data.

RESULTS

Results are presented for a test problem with simulated inputs from the dynamic program. The customer data is the random set and the threat data is the C1 set from the test problems from Appendix E. These data sets are shown graphically in Figure 2 with customers as black dots, the depot as the green dot, and high threat hexes as red dots. In this example, the theta vector is [0,2000,-2] implying the quadratic function $f(x) = 0 + 2000x - 2x^2$ and the maximum number of vehicles is 6. The limit on non-improving iterations is used as a parameter for comparative results. Initial inventories for the customers are a random vector of values between 2000 and 8000. These initial inventories are developed once and held constant between the instances described below. The probability of a successful transition between low threat hexes is 0.999, the probability of a successful transition between low and high threat hexes is 0.994, and the probability of a successful transition between high threat hexes is 0.99. Other parameters include a range of 494 kilometers for the vehicles and a vehicle speed of 148 kilometers per hour. Average solutions for three replications for the problem are shown in Table 1. The limit on non-improving iterations is indicated in the table. In the cases with Or-opt, the local search phases consist of the 2-opt*, Or-opt, and Relocate operators while the cases without Or-opt use the 2-opt* and Relocate operators. While not a large enough sample from which to draw definitive conclusions, these results contain some interesting observations. Based on the results from Chapter III, the inclusion of Or-opt is expected to influence solution quality positively and run time negatively. However, only one of these expectations is met for this particular problem. The solution quality in terms of both value and probability of survival is nearly identical irrespective of the inclusion of Or-opt. The run time increases as expected. The results are also solved using two limits on the number of non-improving iterations—5 and 15 as indicated in Table 1. The case with a higher limit requires more run time as expected but again the solution quality in terms of both value and survivability are quite similar regardless of the limit with the exception of the six vehicle case. When allotted six vehicles, the higher limit runs are able to find better solutions in terms of value in both the cases where Or-opt is and is not used. This may indicate a higher limit is useful for larger number of vehicles. Therefore, a dynamic limit that increases as the number of vehicles increases may yield better results.

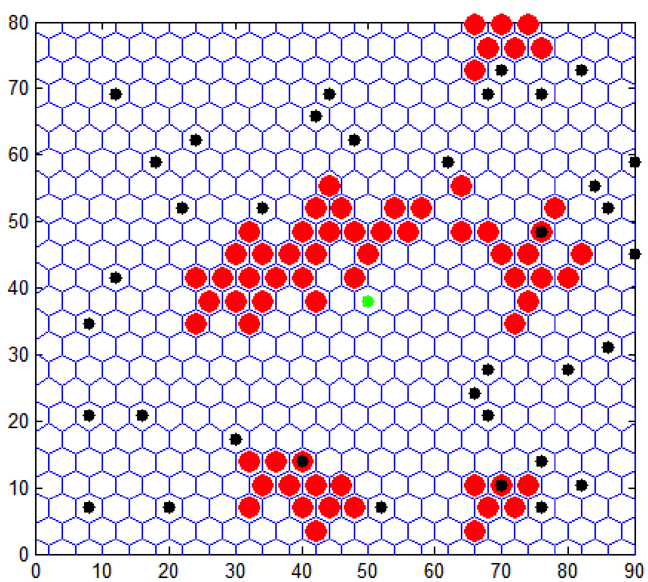

Figure 2: Random customers with clustered threats

Table 1: Routing algorithm results

Or-opt included?	Limit on number of non-improving iterations: 5 — Total solution time (s)	Number of vehicles	Value	Avg probability of survival for each vehicle	15 — Total solution time (s)	Number of vehicles	Value	Avg probability of survival for each vehicle
No	50.2	1	250.3	0.78	179.6	1	250.0	0.78
		2	407.9	0.84		2	408.0	0.86
		3	506.8	0.90		3	513.4	0.87
		4	552.6	0.91		4	558.3	0.90
		5	557.5	0.90		5	568.4	0.90
		6	470.1	0.91		6	521.0	0.91
Yes	74.7	1	250.0	0.78	237.1	1	251.0	0.78
		2	407.4	0.86		2	407.5	0.84
		3	508.8	0.89		3	516.1	0.90
		4	558.2	0.89		4	565.4	0.89
		5	568.9	0.90		5	566.8	0.90
		6	467.4	0.91		6	506.7	0.92

Overall, these results indicate two things: first, the viability of this method is shown because the routing algorithm is able to solve the problem as expected. Second, the unexpected solution characteristics point to the need for a more detailed analysis into both the method and the parameters used in this experiment. Furthermore, these solution characteristics may hint at an inherently different problem structure for the MILIRP compared to the SDVRPTW.

CONCLUSIONS

This research introduces multiple-customer routing to the MILIRP with results presented for a test problem. This novel approach allows for more flexibility than McCormack's [1] method. The next phase of research will involve subjecting the algorithm to the test problems to determine its effectiveness. Since no previous solutions exist for comparison, the goal is to field an algorithm that produces good solutions, as judged by subject matter experts, within a reasonable time frame. Beyond that, this method is merely an initial attempt and improvement is likely possible. Adapting more complex solution methods used on the IRP as documented in [2] will likely yield superior results for the MILIRP.

The addition of time windows to the problem may require significant changes to the routing algorithm. In the current implementation, the time windows for all customers cover the entire period. In effect, this means time windows are not incorporated into the current method. In a future effort incorporating this aspect, the time windows will be randomly generated within each period. This randomness is preferred for the MILIRP because unpredictability is important for security purposes. A customer in the MILIRP does not want to be forced into a predictable pattern of deliveries.

If time windows are in effect, it may be necessary to place the customer(s) being removed from the good route onto a new route entirely within the 2-opt* and Or-opt operators during the value LS phase. This is necessary because the algorithm should not exclude an improving solution simply because the customer(s) being removed from the good route does not fit into the bad route. However, if time windows are simply ignored on the bad route, then it may be infeasible. This is not an issue for a bad route, but in later iterations the algorithm may try to move part of this infeasible route back into a good route, thereby yielding an infeasible solution.

Furthermore, the traditional VRPTW assumes a waiting time adds to the time of the route but not the cost (e.g., it doesn't cost a truck to sit in a parking lot other than the lost time). However, in the case of the MILIRP with CUAS as the vehicles, loitering does add to the cost of the solution because it decreases the range of the CUAS. This waiting time must be accounted for in the cost of the solution.

This research also uses a lexicographic ordering of route value and route security. Furthermore, physical distance is not accounted for except in limiting the total range of each vehicle. It is not used as an objective. The application of more sophisticated multi-criteria optimization techniques may yield solutions that better balance these competing objectives.

Appendix

This appendix provides the detailed data for the suite of test problems presented in the paper. Customer 37 is the depot for each of the following test problems. A problem instance is constructed by pairing a set of customer coordinates with a threat map and the initial customer inventory levels.

Customer coordinates and initial inventory levels:

Random customer coordinates			Clustered customer coordinates			Random-clustered customer coordinates				Initial Inventory
Customer	X Coord	Y Coord	Customer	X Coord	Y Coord	Customer	X coord	Y coord	Customer	Level
1	88	45	1	12	9	1	12	9	1	7500
2	76	48	2	19	12	2	19	12	2	6000
3	68	28	3	26	17	3	26	17	3	6000
4	8	21	4	17	15	4	17	15	4	8000
5	64	24	5	9	19	5	9	19	5	4000
6	84	55	6	23	6	6	23	6	6	4000
7	84	31	7	14	16	7	14	16	7	5000
8	84	52	8	17	20	8	17	20	8	4000
9	76	69	9	19	6	9	19	6	9	8000
10	48	62	10	21	18	10	57	61	10	7500
11	68	21	11	7	55	11	54	68	11	7500
12	12	42	12	12	61	12	59	68	12	5000
13	20	52	13	16	58	13	65	64	13	7000
14	80	10	14	9	68	14	70	71	14	5000
15	20	7	15	14	68	15	54	78	15	7000
16	28	17	16	11	77	16	59	75	16	4000
17	32	52	17	16	75	17	64	74	17	4000
18	8	7	18	19	75	18	66	77	18	6000
19	12	69	19	59	9	19	21	19	19	6500
20	76	14	20	64	14	20	65	41	20	8000
21	52	7	21	65	9	21	66	10	21	5000
22	16	21	22	72	9	22	28	8	22	6000
23	60	59	23	75	9	23	32	60	23	5500
24	40	66	24	74	15	24	13	33	24	8000
25	8	35	25	59	20	25	31	48	25	8000
26	24	62	26	64	5	26	49	14	26	5500
27	40	14	27	68	17	27	41	70	27	8000
28	44	69	28	57	61	28	61	66	28	6500
29	16	59	29	54	68	29	20	68	29	5000
30	88	59	30	59	68	30	64	34	30	7500
31	68	69	31	65	64	31	69	22	31	7500
32	76	7	32	70	71	32	33	29	32	4000
33	80	28	33	54	78	33	32	10	33	8000
34	68	10	34	59	75	34	60	57	34	4000
35	80	73	35	64	74	35	15	43	35	6500
36	68	73	36	66	77	36	12	72	36	5000
37	48	38	37	48	38	37	48	38	37	4500

Random threats:

R1

X coord	Y Coord
0	0
0	27.71281
2	51.96152
2	86.60254
4	6.928203
4	13.85641
6	17.32051
8	20.78461
8	55.42563
8	62.35383
8	76.21024
10	38.10512
12	69.28203
14	65.81793
14	79.67434
18	45.03332
18	58.88973
18	79.67434
20	34.64102
20	48.49742
20	83.13844
22	24.24871
22	31.17691
24	69.28203
24	76.21024
26	10.3923
28	34.64102
28	83.13844
30	10.3923
32	55.42563
32	62.35383
38	51.96152
38	58.88973
40	6.928203
40	55.42563
40	76.21024
46	51.96152
48	6.928203
48	62.35383
54	51.96152
54	58.88973
56	41.56922
58	24.24871
60	6.928203
60	13.85641
62	17.32051
62	38.10512
62	79.67434
66	45.03332
68	0
68	62.35383
72	6.928203
72	55.42563
74	79.67434
78	65.81793
80	69.28203
82	24.24871
82	31.17691
86	86.60254
88	6.928203
90	10.3923
90	38.10512
90	51.96152
90	58.88973
92	62.35383
96	13.85641
98	65.81793
102	24.24871

R2

X coord	Y Coord
0	27.71281
0	76.21024
4	55.42563
4	62.35383
4	69.28203
8	34.64102
10	65.81793
10	86.60254
14	10.3923
16	0
16	20.78461
16	69.28203
22	65.81793
22	86.60254
26	65.81793
28	20.78461
28	62.35383
30	51.96152
30	58.88973
32	76.21024
34	45.03332
36	69.28203
38	17.32051
38	38.10512
38	79.67434
44	0
46	72.74613
48	0
48	83.13844
50	79.67434
54	10.3923
54	45.03332
54	65.81793
54	72.74613
58	3.464102
58	58.88973
58	65.81793
60	6.928203
62	17.32051
62	24.24871
64	0
66	38.10512
66	51.96152
66	65.81793
68	34.64102
70	86.60254
74	65.81793
74	86.60254
76	0
76	62.35383
78	45.03332
78	51.96152
80	34.64102
82	3.464102
86	10.3923
86	72.74613
88	0
90	17.32051
90	38.10512
90	86.60254
92	20.78461
92	34.64102
94	65.81793
98	51.96152
98	58.88973
102	17.32051
102	45.03332
102	65.81793

R3

X coord	Y Coord
2	45.03332
4	6.928203
4	13.85641
6	31.17691
10	17.32051
12	20.78461
14	24.24871
14	72.74613
18	45.03332
18	79.67434
20	48.49742
20	76.21024
22	3.464102
22	31.17691
22	65.81793
24	13.85641
24	34.64102
34	17.32051
34	24.24871
34	45.03332
34	65.81793
34	86.60254
36	0
36	76.21024
40	55.42563
42	51.96152
42	79.67434
44	76.21024
44	83.13844
46	17.32051
46	24.24871
46	72.74613
52	41.56922
56	13.85641
56	55.42563
60	0
60	48.49742
62	65.81793
66	65.81793
68	69.28203
70	17.32051
70	31.17691
70	45.03332
70	58.88973
72	6.928203
72	13.85641
74	45.03332
74	51.96152
76	27.71281
76	83.13844
78	51.96152
78	65.81793
78	72.74613
80	20.78461
80	34.64102
80	41.56922
84	76.21024
86	17.32051
86	79.67434
88	0
88	48.49742
90	51.96152
92	41.56922
92	48.49742
94	3.464102
94	72.74613
96	27.71281
96	69.28203

R4

X coord	Y Coord
0	6.928203
0	27.71281
4	69.28203
6	24.24871
6	38.10512
6	58.88973
6	86.60254
8	41.56922
10	45.03332
10	51.96152
12	13.85641
12	41.56922
14	38.10512
18	51.96152
20	0
20	20.78461
20	48.49742
22	31.17691
24	34.64102
26	24.24871
26	72.74613
30	51.96152
32	34.64102
34	24.24871
36	55.42563
40	13.85641
44	27.71281
44	62.35383
44	69.28203
44	83.13844
50	10.3923
50	38.10512
52	6.928203
52	13.85641
56	0
56	6.928203
56	13.85641
60	20.78461
60	41.56922
64	0
64	6.928203
64	69.28203
66	58.88973
66	72.74613
68	0
68	6.928203
70	65.81793
72	62.35383
74	24.24871
74	45.03332
76	76.21024
78	24.24871
82	3.464102
82	31.17691
82	45.03332
84	27.71281
84	76.21024
86	3.464102
88	34.64102
90	45.03332
90	65.81793
92	76.21024
94	24.24871
96	34.64102
96	48.49742
98	31.17691
100	6.928203
102	79.67434

R5

X coord	Y Coord
4	48.49742
6	65.81793
6	86.60254
10	58.88973
10	72.74613
10	79.67434
12	41.56922
12	48.49742
12	76.21024
16	34.64102
16	69.28203
18	65.81793
20	55.42563
22	72.74613
22	79.67434
24	6.928203
26	86.60254
34	38.10512
36	55.42563
36	62.35383
38	86.60254
40	6.928203
40	13.85641
40	27.71281
40	41.56922
40	69.28203
42	65.81793
44	27.71281
44	41.56922
46	31.17691
46	86.60254
48	76.21024
52	20.78461
52	55.42563
52	69.28203
54	58.88973
56	62.35383
60	34.64102
62	31.17691
64	20.78461
64	55.42563
68	62.35383
70	24.24871
70	45.03332
70	79.67434
72	62.35383
76	20.78461
76	55.42563
78	17.32051
78	51.96152
80	27.71281
80	76.21024
82	3.464102
82	45.03332
84	34.64102
84	76.21024
88	13.85641
88	69.28203
90	38.10512
90	51.96152
94	3.464102
94	79.67434
96	6.928203
96	62.35383
98	65.81793
100	69.28203
100	76.21024
102	58.88973

Clustered threats:

C1

X coord	Y Coord
24	34.64102
24	38.10512
24	41.56922
30	38.10512
30	41.56922
30	45.03332
32	6.928203
32	10.3923
32	13.85641
32	34.64102
32	38.10512
32	41.56922
32	45.03332
32	48.49742
38	10.3923
38	13.85641
38	41.56922
38	45.03332
40	3.464102
40	6.928203
40	10.3923
40	13.85641
40	38.10512
40	41.56922
40	45.03332
40	48.49742
40	51.96152
46	6.928203
46	10.3923
46	48.49742
46	51.96152
46	55.42563
48	6.928203
48	41.56922
48	45.03332
48	48.49742
54	48.49742
54	51.96152
56	48.49742
56	51.96152
62	55.42563
64	3.464102
64	10.3923
64	48.49742
64	72.74613
64	79.67434
70	6.928203
70	10.3923
70	45.03332
70	48.49742
70	76.21024
70	79.67434
72	6.928203
72	10.3923
72	38.10512
72	41.56922
72	45.03332
72	48.49742
72	76.21024
72	79.67434
78	41.56922
78	48.49742
78	51.96152
78	76.21024
80	41.56922
80	45.03332
96	58.88973
96	69.28203

C2

X coord	Y Coord
0	72.74613
0	76.21024
0	79.67434
6	69.28203
6	72.74613
6	76.21024
6	79.67434
8	69.28203
8	72.74613
14	72.74613
14	76.21024
46	45.03332
46	48.49742
48	45.03332
48	48.49742
48	51.96152
54	41.56922
54	45.03332
54	48.49742
54	51.96152
56	45.03332
56	48.49742
56	51.96152
62	65.81793
64	62.35383
64	69.28203
64	72.74613
70	69.28203
70	72.74613
72	58.88973
72	62.35383
72	65.81793
72	69.28203
72	72.74613
78	3.464102
78	6.928203
78	10.3923
78	58.88973
78	62.35383
78	65.81793
78	69.28203
78	72.74613
78	76.21024
80	3.464102
80	6.928203
80	10.3923
80	48.49742
80	51.96152
80	55.42563
80	58.88973
80	62.35383
80	65.81793
80	69.28203
80	72.74613
80	76.21024
86	3.464102
86	6.928203
86	10.3923
86	13.85641
86	55.42563
86	65.81793
86	69.28203
86	72.74613
88	3.464102
88	51.96152
88	55.42563
88	58.88973
94	3.464102

C3

X coord	Y Coord
0	38.10512
0	41.56922
6	38.10512
6	41.56922
8	31.17691
8	34.64102
8	38.10512
8	41.56922
8	48.49742
14	34.64102
14	38.10512
14	41.56922
14	45.03332
14	48.49742
14	51.96152
14	55.42563
16	34.64102
16	41.56922
16	45.03332
16	48.49742
16	51.96152
22	45.03332
22	48.49742
22	51.96152
32	41.56922
38	38.10512
38	45.03332
40	38.10512
40	41.56922
40	45.03332
46	13.85641
46	34.64102
46	38.10512
46	41.56922
48	10.3923
48	13.85641
48	41.56922
54	0
54	3.464102
54	6.928203
54	10.3923
54	13.85641
54	17.32051
56	0
56	3.464102
56	6.928203
56	10.3923
56	13.85641
56	17.32051
62	0
62	3.464102
62	17.32051
64	0
64	3.464102
70	3.464102
70	6.928203
70	31.17691
72	24.24871
72	27.71281
72	31.17691
78	24.24871
78	27.71281
78	31.17691
78	34.64102
80	24.24871
80	27.71281
80	31.17691
80	34.64102

C4

X coord	Y Coord
0	0
0	3.464102
0	6.928203
0	41.56922
0	45.03332
6	3.464102
6	6.928203
6	38.10512
6	41.56922
6	45.03332
6	48.49742
8	0
8	41.56922
8	45.03332
14	45.03332
30	6.928203
30	13.85641
32	3.464102
32	6.928203
32	10.3923
38	3.464102
38	6.928203
38	10.3923
38	13.85641
38	17.32051
40	6.928203
40	10.3923
46	13.85641
48	31.17691
48	34.64102
54	34.64102
54	38.10512
54	79.67434
56	31.17691
56	34.64102
56	38.10512
56	55.42563
56	76.21024
56	79.67434
62	31.17691
62	34.64102
62	38.10512
62	41.56922
62	48.49742
62	51.96152
62	76.21024
62	79.67434
64	31.17691
64	38.10512
64	45.03332
64	48.49742
64	51.96152
64	55.42563
64	58.88973
70	51.96152
70	55.42563
72	6.928203
72	48.49742
72	55.42563
78	6.928203
78	10.3923
78	13.85641
78	17.32051
80	6.928203
80	10.3923
86	6.928203
86	10.3923
86	13.85641

C5

X coord	Y Coord
24	31.17691
24	34.64102
24	38.10512
24	58.88973
24	62.35383
24	69.28203
24	72.74613
24	76.21024
30	31.17691
30	34.64102
30	38.10512
30	55.42563
30	58.88973
30	62.35383
30	65.81793
30	69.28203
30	72.74613
30	76.21024
30	79.67434
32	27.71281
32	31.17691
32	34.64102
32	55.42563
32	58.88973
32	69.28203
32	72.74613
32	76.21024
32	79.67434
38	34.64102
38	38.10512
38	55.42563
38	58.88973
38	62.35383
38	65.81793
38	72.74613
38	76.21024
38	79.67434
40	34.64102
40	38.10512
40	51.96152
40	55.42563
40	58.88973
40	62.35383
40	69.28203
40	76.21024
46	58.88973
46	62.35383
48	24.24871
48	27.71281
48	55.42563
48	58.88973
48	62.35383
54	31.17691
54	34.64102
56	24.24871
56	27.71281
56	31.17691
62	27.71281
62	31.17691
64	24.24871
64	27.71281
64	31.17691
72	3.464102
78	0
78	3.464102
80	0
86	0
86	3.464102

Random-clustered threats:

RC1

X coord	Y Coord
2	51.96152
4	20.78461
4	48.49742
6	17.32051
6	31.17691
6	45.03332
6	65.81793
8	13.85641
8	20.78461
10	17.32051
10	24.24871
10	79.67434
12	13.85641
12	20.78461
14	17.32051
14	38.10512
16	13.85641
18	10.3923
18	17.32051
18	24.24871
20	55.42563
22	10.3923
22	17.32051
24	41.56922
26	31.17691
26	72.74613
32	48.49742
38	45.03332
40	69.28203
42	31.17691
44	48.49742
44	55.42563
46	31.17691
50	58.88973
54	38.10512
54	45.03332
54	51.96152
54	58.88973
56	20.78461
58	45.03332
58	51.96152
58	58.88973
60	20.78461
60	27.71281
60	41.56922
62	51.96152
62	31.17691
64	13.85641
64	20.78461
64	27.71281
64	20.78461
66	24.24871
68	13.85641
68	20.78461
70	17.32051
70	24.24871
70	79.67434
72	20.78461
74	58.88973
82	45.03332
82	51.96152
84	34.64102
84	27.71281
88	41.56922
90	58.88973
92	0
94	3.464102
94	79.67434

RC2

X coord	Y Coord
4	27.71281
4	69.28203
6	45.03332
6	38.10512
8	41.56922
10	51.96152
14	51.96152
18	31.17691
20	48.49742
22	31.17691
24	20.78461
26	24.24871
28	13.85641
30	31.17691
32	13.85641
34	24.24871
36	6.928203
36	13.85641
36	69.28203
36	76.21024
38	10.3923
38	17.32051
38	24.24871
38	79.67434
40	6.928203
40	13.85641
40	76.21024
42	3.464102
42	10.3923
42	51.96152
42	79.67434
44	69.28203
44	76.21024
46	79.67434
48	34.64102
48	76.21024
50	17.32051
50	38.10512
50	79.67434
54	72.74613
56	69.28203
56	6.928203
58	38.10512
60	69.28203
62	65.81793
64	69.28203
64	76.21024
66	51.96152
66	72.74613
66	58.88973
68	69.28203
68	76.21024
68	0
70	72.74613
72	20.78461
72	76.21024
72	62.35383
74	65.81793
74	72.74613
78	72.74613
82	10.3923
84	69.28203
88	76.21024
92	6.928203
92	13.85641
94	65.81793
96	48.49742
102	79.67434

RC3

X coord	Y Coord
8	20.78461
8	27.71281
12	13.85641
14	79.67434
16	6.928203
16	27.71281
18	72.74613
18	79.67434
24	76.21024
26	79.67434
28	41.56922
30	17.32051
32	62.35383
34	17.32051
34	24.24871
36	6.928203
36	34.64102
36	62.35383
36	69.28203
38	65.81793
38	72.74613
40	69.28203
42	65.81793
42	79.67434
44	69.28203
44	76.21024
46	65.81793
46	72.74613
48	0
48	6.928203
48	48.49742
48	69.28203
48	76.21024
50	79.67434
52	6.928203
52	76.21024
54	3.464102
54	17.32051
54	45.03332
56	0
56	6.928203
56	55.42563
56	13.85641
58	3.464102
58	10.3923
60	6.928203
62	3.464102
62	72.74613
64	48.49742
64	69.28203
66	3.464102
66	17.32051
66	24.24871
66	72.74613
68	6.928203
70	24.24871
70	58.88973
70	31.17691
72	48.49742
74	72.74613
78	31.17691
80	62.35383
82	65.81793
88	41.56922
90	38.10512
90	79.67434
92	41.56922
94	3.464102

RC4

X coord	Y Coord
0	13.85641
0	62.35383
0	69.28203
0	27.71281
2	58.88973
4	6.928203
4	20.78461
4	34.64102
4	69.28203
6	17.32051
6	65.81793
6	38.10512
8	13.85641
8	20.78461
8	62.35383
8	69.28203
10	17.32051
10	58.88973
10	65.81793
10	45.03332
12	20.78461
12	48.49742
14	10.3923
14	17.32051
18	17.32051
18	24.24871
18	31.17691
28	69.28203
32	76.21024
34	38.10512
34	65.81793
36	34.64102
38	58.88973
42	38.10512
44	6.928203
44	27.71281
44	83.13844
50	10.3923
54	3.464102
62	24.24871
64	41.56922
64	0
66	38.10512
66	45.03332
66	65.81793
68	41.56922
68	48.49742
68	0
70	45.03332
72	20.78461
72	48.49742
74	45.03332
74	51.96152
74	79.67434
74	45.03332
78	51.96152
84	0
84	41.56922
84	69.28203
86	58.88973
86	65.81793
86	3.464102
88	48.49742
90	65.81793
92	48.49742
94	45.03332
94	24.24871
96	48.49742

RC5

X coord	Y Coord
0	76.21024
2	51.96152
6	31.17691
10	65.81793
12	0
16	69.28203
18	24.24871
18	65.81793
20	62.35383
22	45.03332
26	65.81793
28	76.21024
28	20.78461
28	62.35383
30	10.3923
30	17.32051
40	34.64102
40	41.56922
42	38.10512
42	45.03332
44	34.64102
44	41.56922
44	48.49742
46	38.10512
48	13.85641
48	34.64102
48	41.56922
48	48.49742
48	69.28203
48	83.13844
50	38.10512
50	45.03332
54	38.10512
54	65.81793
58	65.81793
58	3.464102
60	0
60	6.928203
60	41.56922
60	62.35383
60	69.28203
62	10.3923
62	24.24871
62	65.81793
64	0
64	6.928203
64	62.35383
64	69.28203
66	3.464102
66	10.3923
66	24.24871
66	51.96152
66	72.74613
68	0
68	62.35383
68	76.21024
70	65.81793
70	72.74613
74	58.88973
76	0
82	38.10512
82	65.81793
84	55.42563
86	10.3923
90	38.10512
92	41.56922
94	24.24871
102	17.32051

Bibliography

[1] I. M. McCormack, "The Military Inventory Routing Problem with Direct Delivery," Air Force Institute of Technology, Wright-Patterson AFB, OH, 2014.

[2] L. C. Coelho, J.-F. Cordeau and G. Laporte, "Thirty Years of Inventory Routing," Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation, Montreal, 2012.

[3] Q. Mu, Z. Fu, J. Lysgaard and R. Eglese, "Disruption management of the vehicle routing problem with vehicle breakdown," *Journal of the Operational Research Society,* vol. 62, pp. 742-749, 2011.

[4] A. J. Kleywegt, V. S. Nori and M. W. P. Savelsbergh, "Dynamic Programming Approximations for a Stochastic Inventory Routing Problem," *Transportation Science,* vol. 38, no. 1, pp. 42-70, 2004.

[5] S. Mazzeo and I. Loiseau, "An Ant Colony Algorithm for the Capacitated Vehicle Routing," *Electronic Notes in Discrete Mathematics,* vol. 18, pp. 181-186, 2004.

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik,* vol. 1, no. 1, pp. 269-271, 1959.

[7] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research,* vol. 35, no. 2, pp. 254-265, 1987.