Scheduling and Route Selection for Military Land Moves Using Genetic Algorithms

David Montana, Garrett Bidwell, Gordon Vidaver, and Jose Herrero
BBN Technologies
10 Moulton Street, Cambridge, MA 02138
{dmontana,gbidwell,gvidaver,jherrero}@bbn.com

Abstract- We investigate the problem of scheduling the move of a large amount of military equipment from a fort or depot to a port. This problem differs from traditional distribution scheduling problems in a number of ways including: (i) the trucks need to be grouped into convoys, (ii) there is a single source location and a single destination, and (iii) there are potentially so many trucks traveling the same set of roads that the effects on other traffic must be considered. We have divided the problem into two parts: (i) selecting a fixed set of routes and (ii) forming the trucks into convoys and selecting routes and departure times for each convoy. We describe how we have used genetic algorithms to solve each of these problems. We emphasize how the ability to incorportate domain knowledge into the genetic algorithms has allowed us to easily create algorithms well suited to the particular constraints of the problems.

1 Introduction

1.1 The Military Land Move Problem

When deploying troops and equipment from the United States to foreign soil, the longest portion of the journey is either by sea or by air. However, before reaching the ship or plane, there is usually a land move required to reach the port. Similarly, after disembarking the ship or plane, there is usually a land move required to reach the theatre. These land moves have their own associated scheduling problems separate from but intertwined with the scheduling problems associated with the air and sea legs. A detailed description of the military transportation problem including the land move component is given in (Matthews & Holt, 1996). (For a description of our work on a multi-agent society that fits all these different scheduling problems into a coordinated scheduling system, see (Montana et al., 1999).)

So far, we have concentrated our efforts on the first of these land moves, from a fort or supply depot to a seaport or airport. We assume that all small items are packed into larger containers, so that all the items to be moved are containers, vehicles (trucks, tanks, helicopters, etc.) and big items (e.g., expandable bridges). There are three different modes of transportation: military truck, commercial truck, and train. Some of the military trucks and other light vehicles that are being transported by ship are self-transportable, i.e. do not need some other vehicle to carry them to the port.

As for most scheduling problems, there are a variety of constraints of both the hard and soft variety. The hard constraints specify what constitutes a legal schedule, while the soft constraints specify preferences against which to optimize. Some of the hard constraints are:

- The business rules constrain which items can go on which mode of transportation. For example, a helicopter must go on a railcar due to its size, and expandable bridges must go on commercial trucks.
- Military trucks must be formed into convoys, where there are hard limits on the minimum and maximum number of vehicles in a convoy. The reason for the minimum is that each convoy requires support vehicles, which are in short supply. The reason for the maximum is that it becomes hard to keep the convoy together as the numbers of vehicles increases.
- Trucks have finite capacity and hence limitations on the cumulative weight and area or volume (depending on whether or not the truck is a flatbed) of the cargo.
- Each item has a time that it is scheduled to be loaded onto the ship, and it cannot be scheduled to arrive later than this time.
- There are only certain roads on which military traffic can travel.

Some of the soft constraints are:

- The schedule should minimize the difference between the required loading time of an item and the time that it arrives at the port. This minimizes the costs associated with keeping items in a staging area.
- There is a suggested maximum number of military vehicles per hour on each particular piece of road (i.e., link in the transportation graph) so as not to block civilian traffic too much. We refer to this as the link's capacity.

- The routes traveled by the convoys should not have too many turns, and convoys should not cross each others' paths. The reason is to avoid adding unnecessary confusions that can lead to convoys being broken.

The military land move problem differs sufficiently from other standard ground transportation problems that the techniques used to solve these other problems generally do not apply. The differences between it and other standard vehicle routing problems include:

- In the standard problems, there are usually either multiple pickup locations or multiple dropoff locations, as opposed to moving everything between just two points.
- In the standard problems, there are no constraints involving grouping into convoys or overuse of roads.

The standard problems instead tend to have constraints on the time each vehicle can take to traverse its route (referred to as "time windows") (Thangiah, 1995) or the capacity of vehicles (Filipec, Skrlec & Krajcar, 1998). The standard problems also have each truck traveling a different route and attempt full coverage of a set of geographic locations. A fuller set of references to techniques for solving the standard vehicle routing problems is given in (Gendreau, Laporte & Potvin, 1997) and (Golden & Assad, 1988). An example of another non-standard ground transportation problem that requires different techniques to solve because of its different constraints is described in (Gabbert et al., 1991).

1.2 Our Approach

We have split the military land move problem into two separate problems. The first problem is to find an optimal set of routes for use by the convoys. The second problem is to handle the remaining scheduling tasks: packing up trucks, forming the trucks into convoys, selecting departure times for convoys, assigning convoys to one of the routes from the precomputed set of routes, and scheduling trains and commercial trucks. Optimizing a route specifically for each convoy while simultaneously optimizing the other parts of the schedule is too computationally complex without much payoff. Additionally, to reduce confusion it can be better to have a small number of fixed routes that are well marked and that everyone knows. The one disadvantage of splitting it into two problems is that it makes it trickier in the future to adaptively modify routes in response to changing traffic conditions or, in the case of in-theatre land moves, intelligence data on which roads are passable. However, our multi-agent approach (Montana et al., 1999) provides a solution for this.

In our solution of these two optimization problems, we have combined heuristics of many varieties with genetic algorithms, an approach that Davis (1991) refers to as hybridization. Grefenstette (1987) gave the first evidence that initializing the population of a genetic algorithm with non-random individuals derived by heuristics could greatly improve the genetic algorithm performance. (Burke, Newall & Weare, 1998) documented how heuristic initialization only improved performance as long as diversity was maintained in the initial population. Davis (1991) went beyond just advocating for the use of heuristics for initialization but rather, where appropriate, in all aspects of the genetic algorithm, including domain-specific chromosomes and operators. The military land move problem is a very large problem with potentially thousands of tasks that needs to be solved in minutes. A reasonably fast solution is more important than global optimality, and we can sacrifice a small amount of optimality for a fast solution. Starting from scratch cannot achieve our speed objectives, and so we must incorporate heuristics.

One final aspect of our approach worth mentioning is our use of multi-objective evaluation functions. In both subproblems, we have multiple criteria to optimize, and we trade off between them by combining them into a single evaluation function.

2 Route Optimizer

What makes our problem different from standard route optimization problems is that we are looking for a set of routes, rather than an individual route, that jointly optimize an evaluation function. A set of multiple routes can potentially handle more military vehicles per hour than a single route without exceeding the soft quota on military vehicles for a particular road. Since the joint capacity of a set of routes can only be determined by looking at the routes as a set rather than individually, we must perform the optimization over the space of route sets.

So, the problem is to find an optimal set of routes from point A to point B traversing a given transportation network. In addition to the objectives of (i) optimizing joint vehicle capacity and (ii) achieving some sort of balance between the routes in terms of capacity, we also seek to (iii) minimize the travel time of the routes and (iv) minimize the number of switches from one road to another. This last objective is important because turning onto another road can make it difficult for the long line of trucks all following one another.

The transportation network, provided by the Military Traffic Management Command, contains the list of links (i.e., sections of roads) allowed to be used by military vehicles. For each link, we know not only its beginning and ending node/location but also the constraints such as capacity in military vehicles per hour and expected speed for a convoy on this link.

Representation - Each chromosome is a list of

[(1,2,3) (1,4,7,8,6,3)]
[(1,4,7,5,2,3) (1,5,6,3)] $\mapsto$ [(1,2,3) (1,5,6,3)]

Figure 1: This figure illustrates the route crossover operator taking the first route from the first parent and the second route from the second parent to make a child with two routes. The numbers in the routes are the identifiers of the nodes in the transportation network.

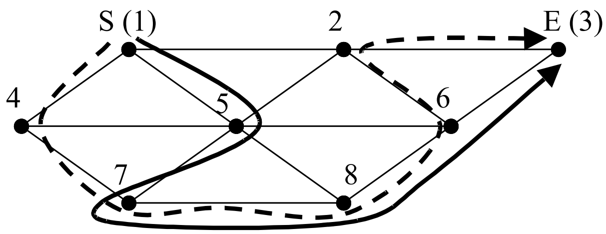

Figure 2: This figure illustrates the "wrap" mutation operator. The initial route from node 1 to node 3 is (1,5,7,8,6,3), and then 6 is chosen as the start of the mutation and 7 as the end. The dashed route shows the result (1,4,7,8,6,2,3). The section between 6 and 3 has been mutated to go through 2, and the section between 1 and 7 has been mutated to go through 4. The section (7,8,6) remains unchanged.

routes, and each route is the list of nodes in the transportation graph through which it passes. The number of routes in each chromosome is a constant fixed at the beginning of the run. Routes are of variable length.

Operators - There are three operators, a crossover and two mutations. The crossover operator, pictured in Figure 1, creates a child using the first k routes from the first parent and the last $N-k$ routes from the second parent, where N is the total number of routes and k is a randomly selected number.

There are two types of mutation operator: "no-wrap" and "wrap". The "no-wrap" mutation operator selects a route and chooses random start and end points within the route, with the start point constrained to be before the end point in the route. It then finds a new route between the two points using the same algorithm used by the initialization procedure to generate random routes (see below). In contrast, the "wrap" mutation operator does not constrain the start point to be before the end point. The mutation can "wrap around" the ends of the route and mutate the route from the randomly chosen start point to the ending destination of the route, and then again from the start of the route to the other mutation point (see Figure 2).

Evaluation Function - The evaluation function balances a number of competing concerns: (i) route travel time, (ii) truck load, (iii) road switches, and (iv) balanced load over all routes. The formula is given by

$$E = (1 - C_b\sigma_c)\sum_{i=1}^{n}\{[\mathrm{Cap}(\mathrm{r}_1, ..., \mathrm{r}_i) - \mathrm{Cap}(\mathrm{r}_1, ..., \mathrm{r}_{i-1})][e^{-\mathrm{T}(\mathrm{r}_i)}][1 - C_s\mathrm{S}(\mathrm{r}_i)]\}$$

where:

- r_i is an individual route,
- σ_c is the standard deviation in the individual routes' capacities,
- $\mathrm{S}(r_i)$ is the number of road switches in the route r_i divided by the number of nodes in the route,
- $\mathrm{T}(r_i)$ is the time to traverse route r_i,
- Cap(routes) is the joint vehicle capacity of the routes with an upper limit cutoff based on the capacity of the port to unload, and
- C_b, C_s and C_t are constants.

Initialization - The population is initialized with chromosomes each consisting of a set of randomly generated routes. To generate a route that is random but not just a random walk, we label all the nodes on the relevant sections of the transportation graph based on the shortest distance in terms of number of links to the destination. We then generate the route one link at a time, using an approach like simulated annealing where the next step is most likely to move towards the destination, can stay the same distance, and is least likely to move away. After generating the route, we remove any loops.

3 Convoy Scheduler

The convoy scheduler uses the routes generated by the route optimizer and performs all the scheduling required to move the items from the fort or depot to the port. While this move involves military trucks, commercial trucks, and commercial rail, only the portion involving military trucks is complex enough to warrant a genetic algorithm, with the other two modes of transport solvable using simple heuristics. We therefore focus our discussion on scheduling military trucks. (A shorter discussion of the convoy scheduler was presented in (Montana et al., 1998).)

As stated above, the hard constraints include not only weight and volume capacities for the trucks but also minimum and maximum sizes for the convoys. The time that each item is scheduled by the port to be loaded onto the ship is provided, and each item must be at the port by that time. The goals, or soft constraints, are (i) to minimize the amount of time that items sit at the port waiting to be loaded (i.e., minimize staging) and (ii) to minimize the disturbance to civilian traffic on the roads by not sending too many military vehicles on a given road within a given time span.

	Trucks									Convoys		
	1	2	3	4	5	6	7	8	9	1	2	3
	1	1	1	1	1	2	2	2	2	2	4	1

Figure 3: The chromosome pictured in this figure specifies two convoys. The first convoy has trucks 1-5 and travels route 2, and the second convoy has trucks 6-9 and travels route 4. The route for the third potential convoy is ignored because only two convoys are created.

The four basic pieces of information we need to determine are: (i) how trucks are packed, (ii) how trucks are formed into convoys, (iii) what time each convoy leaves the fort, and (iv) what route each convoy travels. The packing of the trucks is done by a heuristic that knows that an item should be packed with other items that need to arrive at around the same time. The departure time of each convoy, once the convoy is formed, is determined by a simple heuristic that selects the latest possible time that still ensures that all items transported by the convoy arrive on time. Hence, only the grouping of trucks into convoys and the selection of routes is done by the genetic algorithm.

Convoy formation is a grouping problem since we need to take the individual trucks and form them into groups. Hence, Faulkenauer's grouping genetic algorithm (1994) might appear to be the right approach. However, we want to take advantage of the fact that there is a natural time ordering of the trucks that affects the grouping into convoys. This time ordering is based on the earliest required arrival time of the items in that truck. Convoys should consist of trucks that are consecutive in this ordering, with the key information being where to draw the boundaries between convoys in this ordering. Faulkenauer's approach does not exploit this information and hence is much more inefficient than an approach that does. We outline such an approach now.

Representation - We use a string-based chromosome consisting of two portions. The first part encodes the mapping of trucks to convoys and is of length *numtrucks*. Each slot has an integer between 1 and *maxconvoys* indicating in which convoy that truck is. The second part encodes the mapping of convoys to routes and is of length *maxconvoys*. Each slot has a number between 1 and *numroutes* indicating which route that convoy travels. Figure 3 shows a simple example of this chromosome structure.

Operators - We use the following set of operators which respect the structure of the chromosome:

- **Convoy-Route Crossover (30%)** takes the convoys from the first parent and the routes from the second parent. This is shown in Figure 4.
- **Convoy Mutation (30%)** loops over the boundaries between convoys and with a certain mutation probability moves the boundary a random amount in one direction or the other. This is illustrated in Figure 5.
- **Route Mutation (20%):** loops over the route assignments and with a certain mutation probability chooses a new random route number.
- **Combined Mutation (20%):** performs both a convoy mutation and a route mutation.

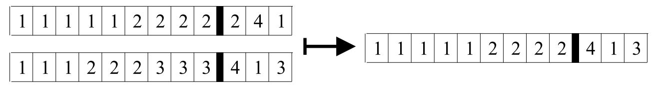

Figure 4: This figure shows the convoy crossover operator.

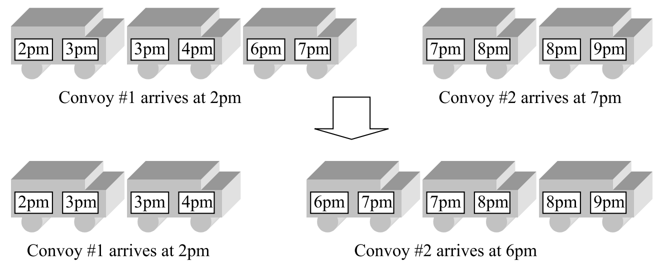

Figure 5: This figure shows the convoy mutation operator reducing the staging cost for a pair of convoys by regrouping the trucks.

Evaluation Function - There are two criteria in our evaluation function:

- **Staging Cost** is the sum over each item of the square of how long before its loading time the item arrives.
- **Link Overuse Cost** is the sum over each hour of each link in the routes of the square of the excess capacity utilized.

We combine them into a single evaluation function by taking a weighted sum.

Initialization - The initialization procedure generates random legal chromosomes to fill the initial population. It generates random convoys by starting at the beginning of the time-ordered list of trucks and picking random sizes for the convoys somewhere between the minimum and maximum size. It assigns convoys to routes by random selection of a route number for each convoy.

Extensions for Reuse of Trucks - The scheduling algorithm outlined above only works if there are enough trucks to complete the move with each truck making at most a single trip. In our demonstration scenario, this turned out not to be the case because there was such a large number of items required to be moved in a short time. Therefore, we have extended the algorithm to be able to have trucks make more than one trip with cargo.

	Trucks									Convoys		
	1	2	3	4	5	6	7	8	9	1	2	3
Epoch 1	1	1	1	1	1	2	2	2	2	2	4	1
Epoch 2	1	1	1	2	2	2	3	3	3	4	1	3

Figure 6: This figure illustrates how the problem is replicated for each epoch in the case when trucks need to be reused.

Our basic approach is to make multiple copies of the problem. The first trip of each truck is the first problem, the second trip of each truck is the second problem, etc. However, we cannot consider these as separate problems because a truck cannot start one trip until it has had time to return from the previous trip and load its next batch of cargo. We therefore jointly, rather than separately, optimize the different problems corresponding to the different trips. We also have to use a more complicated heuristic to compute departure times for the earlier trips to ensure that the trucks are back on time for the later trips. Figure 6 shows the new chromosome.

4 Results and Future Work

The sample problem on which we have tested our approach is that of moving all the equipment of the 1st brigade of the Army's 3rd Infantry Division from its home base of Fort Stewart to the port of Savannah. We also planned the same movement to the alternate port of Jacksonville, demonstrating our ability to reschedule in the case when the primary port is disabled. There were over 1500 large items to move, most of them either self-transportable vehicles or containers that fit one or two per truck. A port simulation application modeling the port of Savannah generated realistic port arrival times for the items. A transportation network for the southeastern United States was provided by the Military Traffic Management Command, the agency that has the job of controlling access to public roads by military vehicles.

The route optimization algorithm ran in under one minute finding a set of two routes to the port of Savannah. It took two or three minutes to find a set of two routes to the port of Jacksonville. The longer time was due to the fact that Jacksonville is farther away from Fort Stewart. Hence, the routes are longer, and there are more possible routes to choose from. These tests were performed on a Sun UltraSparc 1 computer. The convoy scheduler required between five and twenty minutes to reach a good solution.

This scheduling is currently done manually. Hence, there is no other existing algorithm against which to compare our results. Since we did not have time to create other algorithms against which to test, our only alternative was to have the human experts evaluate the results. The human experts who viewed our results were impressed with the quality of the schedules and the ease with which they were found.

We have also incorporated our land move scheduling algorithms into a multi-agent society aimed at demonstrating the capability to automatically schedule all military transportation (Montana et al., 1999). There was a single agent running the route optimization algorithm, finding the route sets for each fort that needed to schedule a move to a particular port. The rationale for having a single agent finding all the route sets is to eventually allow it to ensure that no conflicts arise between different forts and depots planning moves at the same time potentially using the same roads. Each fort and depot had its own agent executing a copy of the convoy formation and scheduling algorithm. The test of this multi-agent society was simultaneously scheduling movements for all nine brigades of the 3rd Infantry Division, plus their support brigades, and all their spare parts and ammunition.

There are multiple directions in which we hope to take this work in the future. First, we want to transition the work from a proof-of-concept demonstration into an operational system. Second, we want to improve the current capabilities to handle dynamic updates, i.e. to modify the routes and schedule to reflect changes to the real world. This is especially important for the in-theatre land move scheduling problem, where the real world is less predictable. Third, we want to work on doing more about coordinating the moves of multiple forts to multiple ports, particularly in terms of road usage.

5 Conclusion

While the military land move problem is different from other standard transportation scheduling problems, using genetic algorithms has allowed us to easily devise an efficient solution. We have divided the land move problem into two subproblems: route selection and convoy formation and scheduling. For each of these subproblems, we have incorporated domain knowledge into the genetic algorithms to create algorithms well matched to the subproblems.

Acknowledgements

We would like to thank Todd Carrico, Stu Draper, Steve Milligan, and Sean Moore for their suggestions and guidance. We would like to thank Andy Schultz and Oliver Oberdorf for their help with software development.

References

Burke, E., Newall, J. & Weare, R. (1998). Initialisation Strategies and Diversity in Evolutionary Timetabling. *Evolutionary Computation,* 6(1), 81–102.

Davis, L. (1991). *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold.

Faulkenauer, E. (1994). A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems. *Evolutionary Computation,* 2(2), 123–144.

Filipec, M., Skrlec, D. & Krajcar, S. (1998). An Efficient Implementation of Genetic Algorithms for Constrained Vehicle Routing Problem. *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics,* 2231–2236.

Gabbert, P. et al. (1991). A System for Learning Routes and Schedules with Genetic Algorithms. In R. Belew & L. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms,* 430–436. San Mateo, CA: Morgan Kaufmann.

Gendreau, M., Laporte, G. & Potvin, J. (1997). Vehicle Routing: Modern Heuristics. In Aarts, E. & Lenstra, J. (Eds.), *Local Search in Combinatorial Optimization*. Chichester: Wiley, 311–336.

Golden, B. & Assad, A. (1988). Vehicle Routing: Methods and Studies. Studies in Management Science and Systems, 16, 7–45.

Grefenstette, J. (1987). Incorporating Problem Specific Knowledge in Genetic Algorithms. In L. Davis (Ed.), *Genetic Algorithms and Simulated Annealing,* 42–60. Los Altos, CA: Morgan Kaufmann.

Matthews, J. & Holt, C. (1996). *So Many, So Much, So Far, So Fast*. Washington, D.C.: United States Transportation Command and Office of the Chairman of the Joint Chiefs of Staff.

Montana, D., Brinn, M., Moore, S., & Bidwell, G. (1998). Genetic Algorithms for Complex, Real-Time Scheduling. *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics,* 2213–2218.

Montana, D., Herrero, J., Bidwell, G. & Vidaver, G. (1999). A Multi-Agent Society for Military Transportation Scheduling. Submitted to *Evolutionary Computation.*

Thangiah, S. (1995). An Adaptive Clustering Method using a Geometric Shape for Vehicle Routing Problems with Time Windows. In L. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms,* 536–543. San Francisco: Morgan Kaufmann.