

Applying Genetic Algorithms to Convoy Scheduling

Edward M. Robinson¹ and Ernst L. Leiss²

1 Binary Consulting, Inc. 4405 East-West Highway, Suit 109,
Bethesda, MD 20814 USA, erobinson@binary-consulting.com

2 Ernst L. Leiss, Dept. of Computer Science, University of Houston,
coscel@cs.uh.edu

Abstract. We present the results of our work on applying genetic algorithms combined with a discrete event simulation to the problem of convoy scheduling. We show that this approach can automatically remove conflicts from a convoy schedule thereby providing to the human operator the ability to search for better solutions after an initial conflict free schedule is obtained. We demonstrate that it is feasible to find a conflict free schedule for realistic problems in a few minutes on a common workstation or laptop. The system is currently being integrated into a larger Transportation Information System that regulates highway movement for the military.

1 Introduction

The objective of this work is to automatically remove conflicts from a convoy schedule. The technique applied was to use a genetic algorithm approach combined with a simulation engine and real world data.

1.1 Convoys

Convoys are used to move equipment and people from one point to another [1], with equipment being trucks and vehicles that can perform the movement. A large military container ship can carry 800 containers and 1200 vehicles, which must be unloaded and moved quickly from the port to their final destination. This movement is managed by a Movement Control Team (MCT) that must organize and schedule the convoys and, at the same time, must integrate the convoy movement with the ongoing transportation of services within their area of responsibility. Data obtained from experienced transporters (human operators) report that 100 convoys per day are not uncommon. The challenge facing the MCT is to schedule the convoys and daily movements so that roads are uniformly used and, more importantly, that two or more

convoys do not run into conflict for a given resource. The term used by transporters for removing conflicts from a schedule is “deconfliction”.

An example of a conflict occurs when two convoys attempt to exit the same gate (assuming that only a single convoy at a time will use the gate) at the same time. Other conflicts include two convoys trying to cross (at right angles) through a single intersection, one convoy passing another on a single route leg, and two convoys merging onto the same highway segment. While these conflicts are most common, other issues can arise based on local rules and regulations so the system must be able to be extended to support these cases. Our system currently detects convoys attempting to merge or cross at a single node and one convoy overtaking another. Military doctrine restricts convoys to one at a time on a given road segment in most cases and further requires a 20-30 minute gap between convoys. This aspect is being added to the automatic conflict removal module as it is being integrated into the target system.

The workflow for convoy scheduling starts when a convoy commander submits a request for a convoy clearance to the MCT. This request includes a list of trucks and other vehicles, a route or strip map, the origin and destination, and a requested time of departure. The MCT will either grant the request without changes or change the departure time or route if the situation requires it.

1.2 Convoy Scheduling

To determine if convoys will run into conflicts, information is supplied regarding the speed of the convoys, the maximum speed on each road segment, the number of vehicles and their dimensions, the required gaps between vehicles, and the routes in a common format. The length of the convoy and speed along a given segment (the lesser of the convoy speed and segment speed) can be used along with the convoy length to calculate the pass time of the convoy (the time elapsed between lead vehicle and trail vehicle crossing the same point). Figure 1 illustrates convoy structure and length.

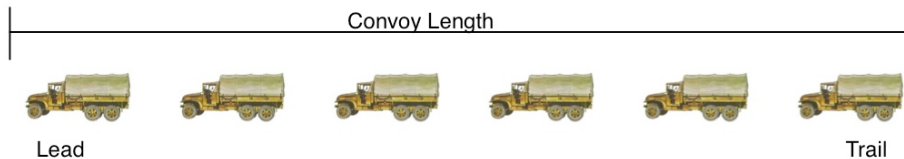


Fig. 1. Example of convoy organization.

Using the speed of the convoy and the pass time each convoy can be stepped forward in time along its given route tracking the times that the lead and the trail pass a given node. A conflict occurs when the lead from another vehicle reaches a node between the crossings of the lead and trail of the original convoy. Another conflict will occur if one convoy passes another on a single segment. This can be observed if the convoys reach the node at the end of the segment in a different order

than they passed the node at the beginning of a segment. We coin the term “inversion” for this kind of conflict.

1.3 Goals

Our primary goal for this project was to create a module for a Transportation Information System (TIS) that automatically adjusts an initial schedule to remove conflicts. Allowable changes include adjusting the departure time or selecting an alternate route (with the same origin and destination). The suitability of new schedules is ranked according to removal of all conflicts followed by a weighted evaluation of the changes to the schedule including closeness to requested departure time, convoy priority, and number of route changes.

Additional goals include support for incremental rescheduling based on changes in the field and extensibility to take into account local rules, guidelines, and opportunities (such as using roadside rest areas to allow one convoy to pass another). A final, but operationally vital constraint for all modules in the TIS is that the system must be field portable, i.e., it must be able to operate on a laptop without Internet access. Implied by this is a relative speedy operation; execution times of several hours are not conducive to responsiveness to developments and changes in the field. Execution times within 10 minutes are considered acceptable as conveyed by experienced transportation personnel.

2 Related Work

Convoy deconfliction is unique enough that there is limited amount of research on the subject. However, there do exist systems that automatically remove conflict from schedules as well as papers describing work on building conflict free convoy schedules while minimizing total time.

2.1 MOBICON

MOBICON [1,2] is a mature system used for scheduling convoys within the continental United States (CONUS). There are no publications that discuss the algorithm used to remove conflicts from convoy schedules. However, it is a reasonable assumption that MOBICON performs this function. MOBICON is tightly integrated with the rules and regulations for executing convoys and obtaining permission from each state to allow the convoy to use its highways; this would make it difficult to extract an extensible core algorithm. Also, MOBICON is a mainframe application, which clearly violates the requirement to be field portable.

2.2 Convoy Movement Problem (CMP)

The Convoy Movement Problem (CMP) attempts to find the minimum overall time for routing multiple convoys from the origins to destinations. [3] showed that

the problem is NP-complete in simple cases, and more complicated in more realistic situations. The fundamental culprit is the aimed-for optimality of a solution. Optimality is not required – at present, convoys are scheduled manually, with very limited computational support; therefore, the presently applied scheduling algorithms are obviously not optimal. Consequently, it is far more useful to apply heuristics in some form, which will improve solutions, but do not necessarily guarantee optimality. The approach taken in [4] recognized this and combined genetic algorithms with a branch and bound approach; however, the results, especially the time requirements of the programs rendered it rather impractical.

3 Our Approach

Our approach to meeting the objectives described in section 1.3 combines genetic algorithms with a simulation engine. The decision to use genetic algorithms (GA) was based on discussions with domain experts in convoys and interaction in the past with researchers in the GA area that emphasized the speed of recalculation in the face of changing conditions. The work done in [3, 4] supports our decision. Also, the notion of modifying a DNA string closely aligns with modifying a string of offset times. The results of the genetic algorithm directly communicated to the domain expert in the field with no translation other than appending the unit of time and name of convoy to the result.

There exist many conflict free schedules for a given set of convoys (a trivial approach would be to start each convoy on a different day). An optimal schedule was believed to be too expensive to calculate (since at best it is NP-complete) but discussions with domain experts determined that an optimal schedule was *not* the goal. The goal was to find a conflict free schedule that favored the priorities and original request times. Such a goal is ideal for a heuristic approach.

Further discussion with domain experts showed that the conditions for conflict and techniques for working around conflict change from location to location. The ideal system would support easy extension and the ability to adapt local business rules into the solver. Experience recommended the use of a discrete event simulation as the method for evaluating a schedule for conflicts. The simulation approach has the benefit of being easy to explain and validate with the domain experts as changes were made. The ability to refine the fitness function with data collected from probes inserted into the simulation to monitor key events and the ability to extend the simulation to handle new constraints and restrictions has been a major win over the more simplified closed forms of convoy interaction.

3.1 Genetic Algorithm Structure

The DNA string in our genetic algorithm consists of offset times to be applied to the requested departure time to determine the actual departure time for the convoy. This offset is taken from a set of offsets, which is a configuration parameter. A nice benefit of this approach is that the resulting DNA string when combined with the convoy names is straightforward to read (ex: convoy 1 starts 15 minutes earlier,

convoy 2 starts on time, convoy 3 starts 20 minutes later). In practice, the offsets are generally in multiples of 5, 10, or 15 minutes.

Mutation was applied by randomly selecting a given offset entry and selecting a randomly different offset time. For crossover, two “parents” are selected at random from the top half of the population and crossing the parent DNA strings at a randomly selected crossover point creates two new “children”. Each single string represented an alternate schedule and individual offsets (convoys) were treated independently.

Randomly selected offsets were used to create the initial population strings along with a single zeroed out string to represent the schedule “as-is” to ease tracking performance. The following steps were applied to each generation:

- Mutate
- Breed
- Evaluate
- Sort

The evaluation step was used to determine the fitness of each string (schedule), which combined a simple evaluation of weights with the simulation of the schedule to determine the number and type of conflicts.

The initial fitness function was determined through interviews with experienced transportation personnel to closely support transportation mission objectives. Inversions were considered worse than contention for a given node and the presence of any conflict was considered worse than other considerations such as closeness of actual departure time to requested departure time, closeness of predicted arrival time to requested arrival time, and favoring a higher priority to selected convoys that might carry fuel or ammunition. Several weights and formulas were tested for convergence on sample data with the final technique to calculate fitness being

$$\sum \frac{100}{inversions} + \frac{10}{conflicts} + \frac{1}{\sum offsets}$$

3.2 Simulation

A discrete event simulation (see Fig. 2) was used to step each convoy through its route starting at the time determined by adding the offset to its requested departure time. Active agents were used to model the lead and trail vehicles; route nodes and legs (edges) were modeled as resources, which were used to track which convoy was utilizing the node at a given time. A global simulation clock was used to determine which event occurs next and to issue an event trigger to the agent that had scheduled the event. Node resources allow locking by the head of a convoy and unlocking by the trail of a convoy. If the node is already locked (indicating that this resource is already being used by another convoy), the node resource will flag the conflict with a monitor that keeps count of all conflicts encountered.

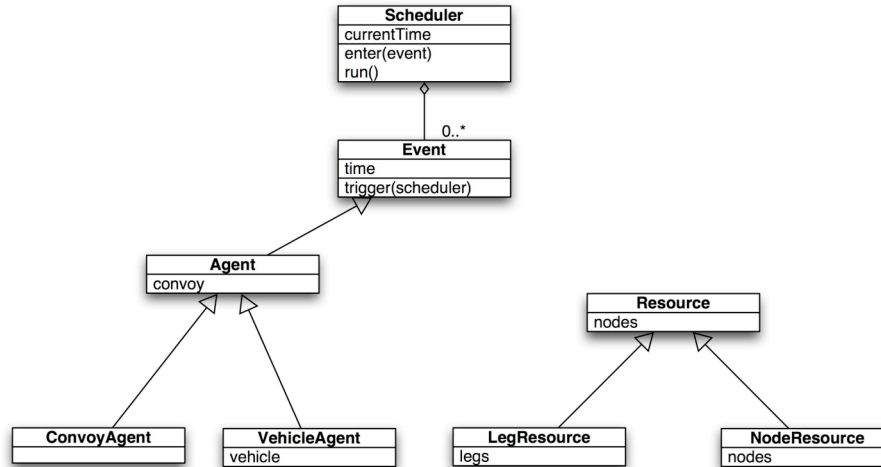


Fig. 2. Class model used to execute the simulation.

Simple queues located with leg resources are used to detect inversions. A convoy is added to the queue when it enters a leg of a route; it is removed from the queue when it exits it. A convoy that overtakes the first convoy would attempt to remove itself out of order and the leg resource would register that with a monitor that keeps a count of inversions.

3.3 Convoy Data

Convoy data were taken from the database used to test the TIS as well as generated from templates. Also, data collected from a simulation of a single convoy were vetted against documented training examples. These data were stored in an object model serving the simulation. The object model's class diagram is illustrated in Figure 3.

The convoy schedule holds all of the convoys and their information. Routes are shared amongst the convoys since that is the case in the field. Additionally, military doctrine allows for convoys to be hierarchically structured into march-units, serials, and columns. This is modeled with a tree structure in the object model to simplify logic. Vehicle information is used to provide the length of the vehicle, which is used in turn to calculate convoy length and pass time. Additionally, the height and weight of a vehicle is needed to determine the validity of switching routes. If an alternate route has a low or weak bridge the convoy may not be able to traverse that route. Also, cargo contents may impact the ability of a convoy to cross a given leg. For example, fuel trucks are not allowed to pass near water in some European countries.

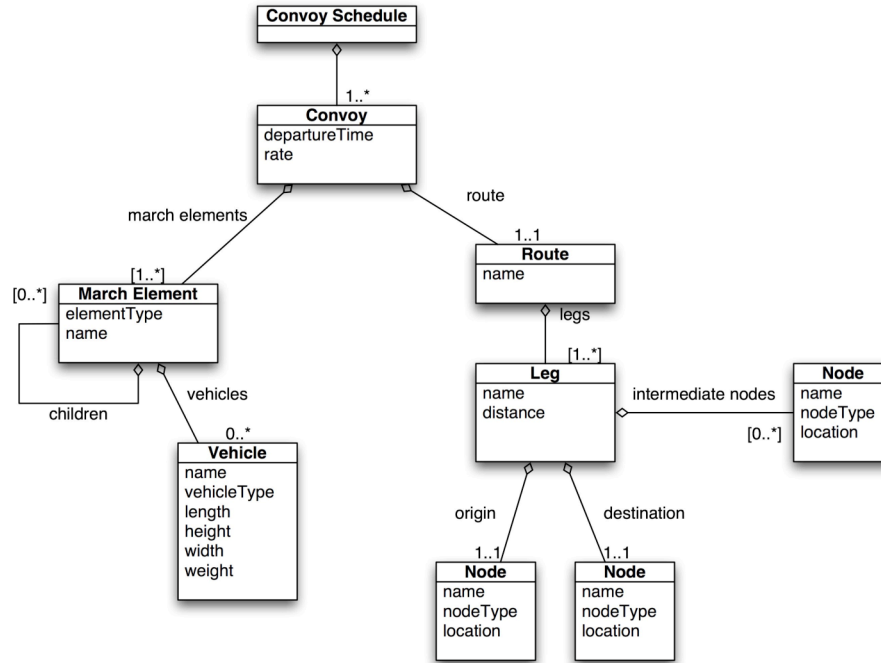


Fig. 3. Object model used for representing convoy in simulation.

4 Results

We tested our system using convoy sizes of 10, 25, and 50 each with 30 vehicles. We were able to reach conflict free schedule within 50 generations for each test case by varying the size of the time window. The results of these runs are given in Table 1. Each convoy requests departure between 9AM and 8PM on a single day. In each case, a conflict free schedule is found within a minute on a reasonably powerful workstation (a 2.7 GHz dual processor PowerPC). The code is written in Java 1.4.2 with no performance tuning or the use of the HotSpot server runtime. Tests were run on Pentium 4 laptops used in the field with comparable performance (within 2 minutes maximum processing time).

Table 1. Window sizes were adjusted to find a conflict free solution in fewer than 50 steps.

Number of convoys – Size	Steps to first conflict free	Time window in minutes	Average offset in minutes	Elapsed time in seconds
10 – 30	23	150	79	8.3
25 – 30	40	480	288	33.8
50 – 30	36	2160	1131	57.7

The time window, which is the maximum amount of time that a convoy schedule entry can be delayed or advanced, has a significant impact on the performance of the search algorithm. We constructed a test to compare the impact of window size on performance on the 25-convoy test data. The results in Table 2 show this performance. Each test run took approximately 2 minutes of user time to complete 150 steps.

Table 2. Test case showing the effect of changing window size on 25 convoys.

Time window in minutes	Steps to first conflict free	Average offset at step 50	Average offset at step 100	Average offset at step 150
360	62	141.3	109.6	110.0
390	36	170.2	121.7	92.9
420	31	178.2	135.6	98.0
450	44	233.1	154.6	117.1
480	46	173.0	150.0	125.7

Clearly the system is sensitive to the time window size with an apparent minimum average offset. This demonstrates the utility of having a human operator in the loop. The operator can guide the solution according to the tactical situation. If speed is the preeminent condition, the operator will stop the simulation as soon as a conflict free schedule is achieved. However, if more time is available, the operator can adjust the time window as well as allow the simulation to run longer in order to achieve a better solution.

5 Conclusion and Future Work

We have shown that a field portable system can be implemented to automatically find a conflict free schedule given an initial convoy schedule. The system is able to find a solution within a minute for up to 50 convoys and can continue searching according to the needs of and as directed by the human operator. This system is currently being integrated into an existing Transportation Information System in order to be fielded in the 2006 to 2007 timeframe.

In the future, we will address improving the performance of the code through standard tuning techniques and developing a larger set of test scenarios to analyze common situations (such as unloading large container ships or return trip deployment). We will also investigate different techniques to manage the sensitivity of the time window to finding better solutions. Finally, the transportation community is interested in utilizing the same techniques for non-convoy transportation uses such as supply chain management and backhaul optimization.

6 References

1. Army Field Manuals: FM 55-1, FM 55-30, and FM 55-65. GlobalSecurity.org
<http://www.globalsecurity.org/military/library/policy/army/fm/index.html>
2. R.J. Shun, Army Logistics Management College (ALMC). Automating Convoy Operations. <http://www.almc.army.mil/alog/issues/NovDec97/MS214.htm>
3. P. Chardaire, G.P. McKeown, S.A. Verity-Harrison, and S.B. Richardson, "Solving a Time-Space Formulation for the Convoy Movement Problem", *Operations Research*, vol. 53, no. 2, pp. 219-230, 2004.
4. Y.N. Lee, G.P. McKeown, and V.J. Rayward-Smith, The Convoy Movement Problem with Initial Delays, *Modern Heuristic Search Methods* (John Wiley & Sons, 1996).