

A Hybrid Approach To Convoy Movement Planning in an Urban City

Ramesh Thangarajoo and Lucas Agussurja

The Logistics Institute - Asia Pacific,
National University of Singapore,
Block E3A, Level 3, 7 Engineering Drive 1,
Singapore 117574

Hoong Chuin Lau

School of Information Systems,
Singapore Management University
80 Stamford Road,
Singapore 178902

Abstract

In this paper, we consider a high-fidelity Convoy Movement Problem motivated by the coordination and routing of convoys within a road transportation network in an urban city. It encompasses two classical combinatorial optimization problems - vehicle routing and resource constrained scheduling. We present an effective hybrid algorithm to dynamically manage the movement of convoys, where we combine the standard Dijkstra's shortest-path algorithm with constraint programming techniques. The effectiveness of the algorithm is illustrated with testing on varying problem sizes and complexity.

Introduction

The Convoy Movement Problem in general involves the coordination and routing of convoys within a transportation network. The concept of movement in convoys or groups originated from the need for a defensive strategy, against threats, with or without an armed escort. Convoys are common in the military to transport large volumes of inventory, equipment and manpower effectively. Movement in convoys provides the benefits of safety in numbers, resource sharing and vehicle breakdown management. Unfortunately, it raises challenges such as slower movement to maintain integrity and impact on traffic. Unlike a typical Vehicle Routing Problem, vehicles traveling together in convoys has an impact on the traffic flow. The transportation network actually becomes a constraint and the focus of coordination has to be shifted to the utilization of the network as a resource from merely considering individual link distance (or travel time) of the network. This has generated consistent interest over the past with the most recent solution model by (Chardaire et al. 2005).

In the convoy movement problem, the link capacity is typically fixed at 1 since we do not allow crossing of convoys at nodes. It is thus a generalization of the Edge-Disjoint Path problem, which has been shown to be NP-hard (Karp 1972). The burden of generating good routes and schedule is further complicated by the restriction that a convoy is not allowed to even wait at the junction for some other convoys to pass (unlike normal traffic flow).

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Generally, in real applications involving both routing and scheduling, the algorithms used tend to be sequential and loosely integrated. The focus tends to be either scheduling or routing, with simple heuristic such as despatching rules used for the other (e.g. in applications such as AGVs(Automated Guided Vehicles) (Qui et al. 2002)). Tackling a combined routing and scheduling problem is efficient, but often compromises the quality of the solution. Likewise, in our convoy movement problem, an attempt to solve the problem in either sequence would result in a loss in solution quality or even feasibility. The remedy would be to either perform iterative search, and/or perform local search to recover the loss in optimality or feasibility. From a heuristic point of view, though this is viable, it does not contribute to the efficiency of the algorithm. By having a more integrated approach, a better and more efficient result can be achieved.

In this paper a fast, efficient and scalable method is presented for planning the movement of convoys over a real transportation network. This work was motivated by a real military logistic application in planning the movement of up to a hundred convoys through a high-fidelity road network. In manual planning, a plan took hours to generate and was usually far from optimality. A further challenge is the operational issue that the problem changed while the plan was executed. This could be in terms of network links which became inaccessible due to congestion or enemy attacks. In other instances the convoy movements requirements themselves could change, as commanders decide to change plan in view of additional intelligence information gather. While a replanning effecting the changes and executed portion of the plan would have been possible, the long planning time did not allow such an option and quick fixes were used at the expense of solution quality. A much faster planning solution of good quality was thus required.

The solution approach we propose for the convoy movement problem is intuitively outlined as follows. It involves decomposing the original problem into a routing and a scheduling problem solved iteratively in a tightly-coupled fashion. Algorithmically, constraint programming is used in conjunction with Dijkstra's shortest path algorithm. By targeting critical arcs and convoys, the makespan of the underlying scheduling problem is progressively improved until no further improvements are possible.

The paper is organized as follows. In the following sec-

tion, we define the convoy movement problem and the related work. The next section presents our proposed hybrid method and shows how the algorithm designed works for each phase of the problem. The following sections present the computational experiments, results and conclude the paper.

Problem Description

The complete convoy movement plan in our application entails a 2-phase movement of the convoys. In the first phase, all the convoys will move from their respective starting point to their respective first destination point. The first phase ends when all the convoys have reached their respective first destination. In the second phase, the convoys will move again from their respective first destination point to their respective second destination point. What is given for each convoy is the release time of the convoy in the first phase and the deadline of the convoy to reach its destination in the second phase. The optimization criteria is to maximize the time gap between the end of the first phase and the start of the second phase. The decision variables are: the route taken by each of the convoy, and the start time of each convoy in both phases. The three main constraints of the problem are: (1) No two convoys are at the same link at any point of time, (2) In the first phase, no convoy is to start before its release time, and (3) In the second phase, all convoys are to arrive at their respective destination before the deadline. The objective is to maximize the time gap between the end of phase 1 and the start of phase 2. Since phase 1 and phase 2 do not overlap, this is equivalent to minimizing the plan's makespan for both phase 1 and phase 2. The plan's makespan is defined as the total time taken starting from the start time of the first convoy to the arrival time of the last convoy.

The convoy movement problem can be seen as a routing and scheduling problem involving movement through a network (directed graph) with start and destination time constraints. Each convoy can be seen as a job, and N denotes the index set of n jobs required to travel through a network $G = (V, E)$.

For phase 1, define α_i^1 , the starting time and β_i^1 the route of convoy i from s_i its start node to f_i^1 its finish node. Similarly for phase 2, define α_i^2 and β_i^2 from f_i^1 which also acts as the phase 2 start node to f_i^2 the respective finish node.

A route β_i (applies to both β_i^1 & β_i^2) is a sequence of edge $(e_{i_1}, e_{i_2}, \dots, e_{i_w})$ such that e_{i_j} and $e_{i_{j+1}}$ are connected for $1 \leq j \leq w - 1$. Let $C(\beta_i)$ be time taken to execute the route β_i , i.e.

$$C(\beta_i) = \sum_{j=1}^w \frac{L(e_{i_j})}{\min\{p_i, S(e_{i_j})\}} + \gamma \sum_{j=1}^{w-1} T(e_{i_j}, e_{i_{j+1}})$$

The second term of the equation captures the fact that going through short links with many turns can be slower than going through a straight long link where γ is a number reflecting the relative weight given to this term.

Given $\alpha = \{\alpha_1, \dots, \alpha_n\}$, $\beta = \{\beta_1, \dots, \beta_n\}$, the objective

$V = \{v_1, v_2, \dots, v_m\}$	Set of nodes.
$E = \{e_1, e_2, \dots, e_q\}$ $\subset V \times V$	Set of directed links
$L(e_j)$	Length of the link.
$S(e_j)$	Maximum speed of the link.
$E(v_j, v_k)$	Directed link from v_j to v_k .
$T(e_j, e_k)$	$\begin{cases} 1 & \text{if } e_j \text{ is connected to } e_k \\ & \& e_j \text{ to } e_k \text{ is a turn} \\ 0 & \text{otherwise} \end{cases}$
$N = \{1, 2, \dots, n\}$	Convoys' index set.
r_i	Release time of convoy.
$s_i \in V$	Starting node of convoy.
$f_i^1 \in V$	First destination node of convoy.
$f_i^2 \in V$	Second destination node of convoy.
t_i	Deadline of convoy to reach d_i^2 .
l_i	Length of convoy.
p_i	Speed of convoy.
α_i^1	Starting time of convoy i from s_i to f_i^1 .
α_i^2	Starting time of convoy i from f_i^1 to f_i^2 .
β_i^1	Route for convoy i from s_i to f_i^1 .
β_i^2	Route for convoy i from f_i^1 to f_i^2 .
$\beta_i[e_j, e_k]$	$(E(s_i, v_x), \dots, e_j, e_{j+1}, \dots, e_{k-1}, e_k, \dots, e_w, \dots, E(v_y, f_i))$, Subsequence of the route
$U(\alpha_i, \beta_i, e')$	$\beta_i[e_j, e_k] = (e_j, e_{j+1}, \dots, e_{k-1}, e_k)$. Time, convoy i reaches end of link e' . $U(\alpha_i, \beta_i, e') = \alpha_i + C(\beta_i[e_{i_1}, e'_{i_1}])$

Table 1: Notations

is to find α and β minimizing:

$$F(\alpha^1, \alpha^2, \beta^1, \beta^2) = \min_{i \in N} \alpha_i^2 - \max_{i \in N} (\alpha_i^1 + C(\beta_i^1)).$$

The following gives the problem constraints:

1. A convoy departs after its release time, $\forall i \in N$:

$$\alpha_i^1 \geq r_i$$

2. A convoy reaches its destination before its deadline, $\forall i \in N$:

$$\alpha_i^2 + C(\beta_i^2) \leq t_i.$$

3. No overlapping (in time) constraint between jobs, $\forall i \in N, \forall j \in N, i \neq j$, $(e_{i_k}, e', e_{i_l}) \subseteq \beta_i$ and $(e_{j_k}, e', e_{j_l}) \subseteq \beta_j$

$$\begin{cases} U(\alpha_i, \beta_i, e') + \frac{l_i + \rho}{\min\{p_i, S(e_{i_l})\}} \geq U(\alpha_j, \beta_j, e_{j_k}) \\ \quad \text{if } U(\alpha_i, \beta_i, e_{i_k}) \leq U(\alpha_j, \beta_j, e_{j_k}) \\ U(\alpha_j, \beta_j, e') + \frac{l_j + \rho}{\min\{p_j, S(e_{j_l})\}} \geq U(\alpha_i, \beta_i, e_{i_k}) \text{ otherwise.} \end{cases}$$

Where ρ is the minimum physical distance between any two jobs.

The first two constraints limit the start and end times of a job. The third constraint seeks to limit the occupancy of an arc to one job at any one time. This can be extended to limit simultaneous occupancy of an arc to its maximum capacity.

Other constraints of interest would be disjunctive constraints that limit the use of arcs within a defined set simultaneously, to a specified value.

Related Work

Convoy movement was first introduced and described in (Lee et al. 1994a). A branch-and-bound algorithm for solving a basic version of the problem, and a description of its implementation on a network of transputers, is given in (Lee et al. 1994b). Subsequently, (Lee et al. 1996) relaxed the starting time of convoys to allow them to start anytime after their release time and not at the release time. Three algorithms were proposed, Branch and bound(B&B, a hybrid of GA and B&B and a pure GA approach. The performance of the algorithms were problem dependent. (Chardaire et al. 2005) models the problem using time as a problem dimension and thus uses a time-space network. Lagrangian Relaxation was applied to this model and the resulting Lagrangian dual function was evaluated using a modified version of Dijkstras shortest path algorithm that is applicable to very large, implicitly-defined graphs. This method was compared with the B&B method in (Lee et al. 1994b) by testing the same set of problems was shown to be more efficient.

The convoy movement problem has been assumed to be difficult to solve in practice. However, (Tuson & Harrison 2005) argued that real instances of convoy planning may be not that difficult. Although the convoy movement problem is NP-hard in general, this is only a statement of the worst-case time-complexity of a problem. Real-world instances were tested using the B&B algorithm and also Lagrangian Relaxation, and they can be solved within reasonable time. The authors explained this by observing most convoys are sufficiently far apart from each other (in both time and space) that they are never likely to intersect. This means that a large number of the convoys can be routed quite independently of the rest of the movement, with relatively few ordering decisions left to be made.

Unfortunately, this may not be the case within urban densely-populated cities and countries where the road network is highly complex and the the area of movement is small. Furthermore, emergency operations may require large number of convoys to move within a limited duration through a dense transportation network. This is the context under which our problem is to be tackled.

Although not many publications can be found on convoy movement per se, other related applications have been presented and are worth discussing here. For example, the strategic level routing of hazardous materials through a given route network was presented in (Iakovou et al. 1999). It considers a multicommodity flow problem with multiple origins-destinations, which involves selection of paths for the transportation of a plethora of hazardous materials. For safety reasons, various constraints are imposed during the transportation process. The authors proposed a Mixed Integer Programming model for this problem. Again, the selection of paths is based on the assumption that the sets of all paths between each origin-destination pair are available. The authors propose a two-phase solution procedure. In the first phase, good lower bounds and upper bounds for

the problem are obtained by relaxing the constraints using Lagrangian multipliers. Then the subgradient search algorithm is applied to search for good dual multipliers and, at the same time, to update the upper bound of the objective functions with the best feasible solution encountered. In the second phase, the gap between bounds is closed by generating lower bounds one by one in an increasing sequences and comparing with the best feasible solution encountered. The procedure stops when the lower bound is greater than or equal to the upper bound. (Nozick et al. 1997) recognizes time varying patterns of accident rates and exposure parameters for the application of routing and scheduling of hazardous materials transportation.

Another line of research similar to convoy problem is the commercial routing and scheduling of vehicles, such as trains over a rail network. They share the constraint of arcs allowing only a single convoy or train to traverse at any one time as seen in (Chih et al. 1990), (Florian et al. 1976) and (Kwon et al. 1998). In (Potvin et al. 2006), a dynamic vehicle routing and scheduling problem with time windows is described where both real-time customer requests and dynamic travel times are considered. Different reactive dispatching strategies are defined and compared through the setting of a single tolerance parameter. The results show that some tolerance to deviations with the current planned solution usually leads to better solutions.

Hybrid Solution Approach

The following pseudo-code gives the solution approach. It consists basically of 2 modules (components): (1) Routing module and (2) Scheduling module. The routing component utilizes the standard shortest path algorithm. Let the function `SHORTEST-PATH(v_i, v_j, D)`, $v_i, v_j \in V$, $D \subseteq E$, computes the shortest path from node v_i to node v_j without using the links in D .

procedure SOLVE:

1. $\forall i \in N, D_i \leftarrow \{\}$
 2. $\forall i \in N, \beta_i^1 \leftarrow \text{SHORTEST-PATH}(s_i, d_i^1, D_i)$
 3. $\alpha^1 \leftarrow \text{SCHEDULE-ROUTES}(\beta^1, r_1, \dots, r_n)$
 4. $obj \leftarrow F_2(\alpha^1, \beta^1)$
 5. $\delta \leftarrow \max$
 6. **while** $\delta > err$ **do**
 7. $D \leftarrow \text{UPDATE-LINKS}(\alpha^1, \beta^1)$
 8. $\forall i \in N, \beta_i^{1'} \leftarrow \text{SHORTEST-PATH}(s_i, d_i^1, D_i)$
 9. $\alpha^{1'} \leftarrow \text{SCHEDULE-ROUTES}(\beta^{1'}, r_1, \dots, r_n)$
 10. $obj' \leftarrow F_2(\alpha^{1'}, \beta^{1'})$
 11. **if** $obj' > obj$ **then do**
 12. $\alpha^1 \leftarrow \alpha^{1'}$
 13. $\beta^1 \leftarrow \beta^{1'}$
 14. **end if**
 15. $iter \leftarrow iter + 1$
 16. **end while**
 17. **output** α^1, β^1
- end procedure**

Routing Module

The routing algorithm follows the standard Dijkstra's algorithm which gives the shortest paths from a start node to all other nodes in the network. Since we do not need all the shortest paths but one, the algorithm is stopped when the destination node is found. Recall that the result of the algorithm is a shortest path tree and we denote by $\pi[v_k]$ the parent node of v_k in the shortest path tree. We also use Q_p to denote a priority queue that sort an element $(v_i, L(e_j))$ in non-decreasing order of $L(e_j)$. The following is the algorithm given in pseudo-code:

procedure SHORTEST-PATH(v_k, v_l):

1. $\forall v_j \in V, v_j \neq v_k, \rho[v_j] \leftarrow \infty$
2. $\forall v_j \in V, v_j \neq v_k, \pi[v_j] \leftarrow \emptyset$
3. $\rho[v_k] \leftarrow 0, \pi[v_k] \leftarrow v_k$
4. $\forall v_j \in V, (v_k, v_j) \in E, Q_p \leftarrow (v_j, L(v_k, v_j))$
5. $\forall v_j \in V, (v_k, v_j) \in E, \rho[v_j] \leftarrow L(v_k, v_j)$
6. $\forall v_j \in V, (v_k, v_j) \in E, \pi[v_j] \leftarrow v_k$
7. **while** $Q_p \neq \emptyset$ **do**
8. $v_j \leftarrow Q_p$
9. **if** $v_j = v_l$ **then return** π, ρ
10. **else do**
11. $\forall v_h \in V, (v_j, v_h) \in E,$
12. **if** $\rho[v_j] + L(v_j, v_h) < \rho[v_h]$ **then do**
13. $\rho[v_h] \leftarrow \rho[v_j] + L(v_j, v_h)$
14. $\pi[v_h] = v_j$
15. **end if**
16. **end if**
17. **end while**

end procedure

Scheduling Module

The following definitions are used for describing the constraints handled by the scheduling module:

- A route β_i is a sequence of edge $(e_{i_1}, e_{i_2}, \dots, e_{i_w})$ such that e_{i_j} and $e_{i_{j+1}}$ are connected for $1 \leq j \leq w - 1$. Let $C(\beta_i)$ be time taken to execute the route β_i , i.e.

$$C(\beta_i) = \sum_{j=1}^w \frac{L(e_{i_j})}{\min\{p_i, S(e_{i_j})\}} + \gamma \sum_{j=1}^{w-1} T(e_{i_j}, e_{i_{j+1}})$$

The second term of the equation captures the fact that going through short links with many turns can be slower than going through a straight long link. Note: when the notation β_i is used, it applies to both β_i^1 and β_i^2 .

- Denote $\alpha^1 = \{\alpha_1^1, \dots, \alpha_n^1\}$, $\alpha^2 = \{\alpha_1^2, \dots, \alpha_n^2\}$, $\beta^1 = \{\beta_1^1, \dots, \beta_n^1\}$, and $\beta^2 = \{\beta_1^2, \dots, \beta_n^2\}$. The objective is to maximize the function:

$$F(\alpha^1, \alpha^2, \beta^1, \beta^2) = \min_{i \in N} \alpha_i^2 - \max_{i \in N} (\alpha_i^1 + C(\beta_i^1)).$$

That is, to maximize the time interval between the end of phase 1 and the start of phase 2. Since these 2 phases are independent of each other (i.e. all convoys must finish phase 1 before phase 2 can start), the problem can be broken into two:

1. find α^1 and β^1 minimizing:

$$F_2(\alpha^1, \beta^1) = \max_{i \in N} (\alpha_i^1 + C(\beta_i^1))$$

2. and find α^2 and β^2 maximizing:

$$F_1(\alpha^2, \beta^2) = \min_{i \in N} \alpha_i^2$$

Which is equivalent to maximizing $F = F_1 - F_2$.

The following are the constraints handled by the scheduling module (in the function: SCHEDULE-ROUTES):

1. deadlines must be met, i.e. $\forall i \in N$:

$$\alpha_i^2 + C(\beta_i^2) \leq t_i.$$

2. A convoy can start moving only after its release time, i.e. $\forall i \in N$:

$$r_i \leq \alpha_i^1$$

3. No overlapping (in time) constraint between convoys:

Let $\beta_i[e_j, e_k]$ be the subsequence of the route $\beta_i = (e_i, \dots, e_j, e_{j+1}, \dots, e_{k-1}, e_k, \dots, e_w)$, i.e. $\beta_i[e_j, e_k] = (e_j, e_{j+1}, \dots, e_{k-1}, e_k)$. Note that $\beta_i[e_j, e_k]$ is a valid route. Let $U(\alpha_i, \beta_i, e') = \alpha_i + C(\beta_i[e_j, e_k])$ be the time convoy i reach the end of a link e' . The following holds: $\forall i \in N, \forall j \in N, i \neq j$, if $(e_{i_k}, e', e_{i_l}) \subseteq \beta_i^1$ and $(e_{j_k}, e', e_{j_l}) \subseteq \beta_j^1$ then

$$\left\{ \begin{array}{l} U(\alpha_i^1, \beta_i^1, e') + \frac{l_i + \rho}{\min\{p_i, S(e_{i_l})\}} \geq U(\alpha_j^1, \beta_j^1, e_{j_k}) \\ \quad \text{if } U(\alpha_i^1, \beta_i^1, e_{i_k}) \leq U(\alpha_j^1, \beta_j^1, e_{j_k}) \\ U(\alpha_j^1, \beta_j^1, e') + \frac{l_j + \rho}{\min\{p_j, S(e_{j_l})\}} \geq U(\alpha_i^1, \beta_i^1, e_{i_k}) \text{ otherwise.} \end{array} \right.$$

Where ρ is the minimum distance between any two convoys. Similarly, the constraint is defined for β^2 .

4. Additional logical constraints:

- $\forall i \in N, \alpha_i^1 + C(\beta_i^1) \leq \alpha_i^2$.
- $\forall i \in N, e_{i_1}^1 = s_i, e_{i_w}^1 = d_i^1 = e_{i_1}^2, e_{i_w}^2 = d_i^2$.

The function SCHEDULE-ROUTES is computed by search with constraint propagation and heuristic to prune the size of the search tree. Given the set of routes β^1 , let the set $R(e)$ be the indices of the routes that contain the link e , i.e. $R(e) = \{i \in N | e \in \beta_i^1\}$. Define the following variables for each common link (common link is a link which is contained in more than one routes): $\{x_e | |R(e)| > 1\}$. The domain of the variable x_e is the set of all possible permutation of $R(e)$. Given a permutation $\pi = (\pi_1, \pi_2, \dots)$, we say π_j precede π_k ($\pi_j \prec \pi_k$) in π if π_j appears before π_k in the permutation. There is a constraint between 2 variables x_e and $x_{e'}$: $\mathcal{C}(x_e, x_{e'})$ if $\exists i, j \in R(e)$ such that $i \neq j$ and $i, j \in R(e')$. The constraint $\mathcal{C}(x_e, x_{e'})$ is defined as follows: $\mathcal{C}(x_e, x_{e'}) \equiv \{(\pi(R(e)), \pi(R(e')) | \forall i, j : i \prec j \text{ in } \pi(R(e)) \iff \prec j \text{ in } \pi(R(e'))\}$.

Every instantiation of a variable is followed by constraint propagation to maintain arc consistency. The heuristic based on the value of r_1, \dots, r_n is used to prune the search space.

The function UPDATE-LINKS disables a link from the most "constrained" route in the current schedule. The most constrained route is defined as the route that has the most deviation from its best possible schedule, i.e. given the current schedule α^1 and β^1 , the most constrained (convoy) i is one that maximizes:

$$\{\alpha_i^1 + C(\beta_i^1)\} - \{r_i + C(\text{SHORTEST-PATH}(s_i, d_i^1, \{\}))\}.$$

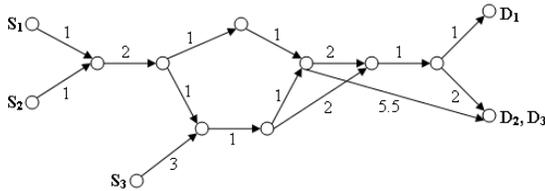
And the link to remove is $e \in \beta_i^1$ with maximum value of $|R(e)|$.

The algorithm runs until the difference in objective value δ is less than predefined value err .

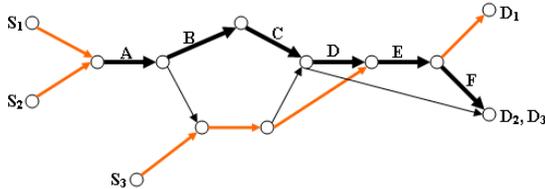
Thus, we have cast the scheduling problem into a constraint satisfaction problem. We use one pass constraint propagation to solve this constraint problem instead of using backtracking.

Example

Here, we illustrate the execution of our algorithm on a small problem instance. Consider the network in the following figure:

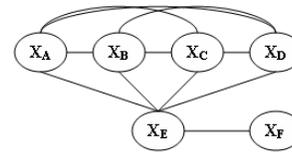


The number on a link denotes the cost (time taken) of the link. We have three convoys with the following source-destination pair (S_1, D_1) , (S_2, D_2) and (S_3, D_3) . We will run through the generation of the initial schedule. The convoys are released at time 0, 1 and 2 respectively. The following figure shows the shortest path of the 3 convoys with the possible conflicting links darken.



In scheduling, we can then define the following variables:

- x_A, x_B, x_C, x_D with domain $\{(1, 2), (2, 1)\}$
- x_E with domain $\{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
- x_F with domain $\{(2, 3), (3, 2)\}$



with the following constraint graph:

The following table shows when the links are used by each convoy initially, with the ones in conflict in bold:

	A	B	C	D	E	F
convoy 1	1-3	3-4	4-5	5-7	7-8	-
convoy 2	2-4	4-5	5-6	6-8	8-9	9-11
convoy 3	-	-	-	-	8-9	9-11

In determining which variable to instantiate next, we use a heuristic based on the release time of the convoys. Since convoy 1 has the earliest release time, we start instantiating the variables starting from the earliest link with convoy 1 involved, i.e x_A . For the link A, since convoy 1 arrives first at the link, we set x_A to be (1, 2) which means convoy 2 has to wait until convoy 1 passes the link before it can use the link, and the updated schedule now becomes:

	A	B	C	D	E	F
convoy 1	1-3	3-4	4-5	5-7	7-8	-
convoy 2	3-5	5-6	6-7	7-9	9-10	10-12
convoy 3	-	-	-	-	8-9	9-11

Now, we have the following current domain (after propagation) for the variables:

- x_A, x_B, x_C, x_D with domain $\{(1, 2)\}$
- x_E with domain $\{(1, 2, 3), (1, 3, 2), (3, 1, 2)\}$
- x_F with domain $\{(2, 3), (3, 2)\}$

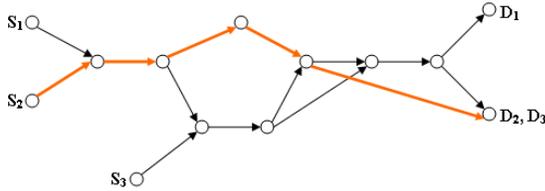
The next variable to be instantiated is x_E since convoy 2 has the next earliest release time and convoy 1 has no more conflicts. It is clear from the table above that x_E will be instantiated to (1, 3, 2) by comparing their arrival time. And because of that, x_F is then instantiated to be (3, 2). Thus, we have: $x_A, x_B, x_C, x_D = (1, 2)$, $x_E = (1, 3, 2)$ and $x_F = (3, 2)$. And the schedule is given by the following:

	A	B	C	D	E	F
convoy 1	1-3	3-4	4-5	5-7	7-8	-
convoy 2	4-6	6-7	7-8	8-10	10-11	11-13
convoy 3	-	-	-	-	8-9	9-11

The cost of the schedule is 13 (the completion time) or (11 + 2), and this is because convoy 2 has to wait for 2 unit time after its release time before it can actually start moving thus delaying its optimal arrival time 11 by 2 unit time.

Consider now the first iterative move, since convoy 2 is the most "constrained" convoy, we remove its link which is most

commonly used by other convoys, which in this case is the link E. When we invalidate link E for convoy 2, we have a new shortest path for convoy 2 as shown in the following:



And we have the following variables: x_A, x_B, x_C with domain $\{(1, 2), (2, 1)\}$ and x_E with domain $\{(1, 3), (3, 1)\}$. And the resulting schedule will be the following:

	A	B	C	D	E	F
convoy 1	1-3	3-4	4-5	5-7	7-8	-
convoy 2	3-5	5-6	6-7	-	-	-
convoy 3	-	-	-	-	8-9	9-11

with $x_A, x_B, x_C = (1, 2)$ and $x_E = (1, 3)$. In this new schedule, convoy 2 has to wait for 1 unit time, and it has to take a path which is 0.5 unit time longer than its shortest path, thus the cost of this schedule is $11 + 1.5 = 12.5$ which is an improvement of the initial schedule by 0.5 unit time. It is not difficult to see that the optimal schedule is one that balances between using longer paths and using conflicting shorter paths among the convoys.

Dynamic Planning

Furthermore, our hybrid solution framework was designed to compute a plan in a relatively short time period so as to allow the derived application to be used in a dynamic setting. Dynamic changes include delays in the progressive movement of convoys which may have impact the movement of other convoys according to plan. More extreme changes could be in terms of the convoy destinations, target arrival times as well as changes in the accessibility of the transportation network.

For dynamic planning, the convoys' current position, the current plan of the convoys, and the set of events that affects the current plan are seen as constraints for a re-optimization of the current plan. By fixing the convoys' current positions as the new start locations of the convoys, the static solution framework can be used in the dynamic mode. To avoid excessive changes to the original plan at the expense of optimality, the route of convoys not directly affected by events can be fixed.

Experimental Results

Practical Problem and Results

The prototype we developed for our convoy movement problem was tested on a 100 convoy test problem instance using a real transportation network with approximately 40000 links.

The geographical information system software ArcGis was used for the purpose of visualization of results.

The solution was evaluated by end users, and it gives an approximate 20% improvement in solution quality with vast improvements in computation time with respect to the existing non-AI method used. Consequently the prototype was integrated with a user interface and deployed as a complete application for field testing. In view of the sensitive nature of this problem, we will not discuss its results further.

Further test runs were also conducted for various sets of realistic convoy problem scenarios to verify the quality of solutions. Due to the absence of an existing tool, the optimality of the algorithm was verified against a MP (mathematical programming) model. These tests were limited to scaled-down versions of the real problems due to the computation time limitation. The tests were conducted on a simplified road network including only the major roads, with up to 10 convoys, all with different start points but sharing end points. Table 2 gives a summary of the results with Var., Con. and Coef. standing for the number of Variables, Constraints and Non-Zero Coefficients respectively. Our proposed HB(hybrid) approach was found to have obtained the optimal solution in all these cases with a linear computation time increase with respect to the number of convoys. In comparison, there is an exponential increase with the MP model due to the increase in variables, constraints and non-zero coefficients.

Problem Description					CPU Time/s	
Convoy	End Pt	Var.	Con.	Coef.	MP	HB
1	1	83	275	1026	2.25	8
2	1	268	984	4229	3.78	10
3	1	559	2129	9612	8.26	11
4	2	956	3710	17179	38.29	12
5	1	1459	5727	26918	236	14
5	2	1459	5727	26922	828	15
5	3	1459	5727	26924	339	15
6	6	2068	8180	38849	802	15
7	7	2783	11069	52954	616	16
8	8	3604	14394	69239	2758	16
9	9	4531	18155	87705	52159	17
10	10	5564	22352	108348	78877	17

Table 2: MP Comparison

Random Instances and Results

For larger convoy problem instances, the performance of our hybrid method was measured with various random problems generated. For a better assessment, a network graphs with 1000 nodes were generated with varying number of links from 1500 to 3000. Randomly generated convoy test problems with 100 to 450 convoys were used to test the hybrid solution model.

Measuring Problem Complexity In transportation problems, the complexity is usually dependant on the network complexity measured by the features of the network such as its structure, connectivity and clustering (Huapu & Ye 2007). However the impact of the actual departure points

and destination points of the convoy has an impact on the problem complexity where complexity refers to the difficulty of the problem.

For the purpose of defining the problem complexity, we define the *criticality* of each arc as its accessibility to each convoy's start-destination pair. More precisely, the criticality with respect to each convoy's route would be 1 if it falls on the shortest path. Otherwise, this value would be a fraction measured by ratio of the shortest path distance over the distance traveled with the shortest path via the specific arc. We define the overall problem complexity measure based on the mean and standard deviation (STD) of the link criticalities.

Results Figure 1 gives a summary of our results.

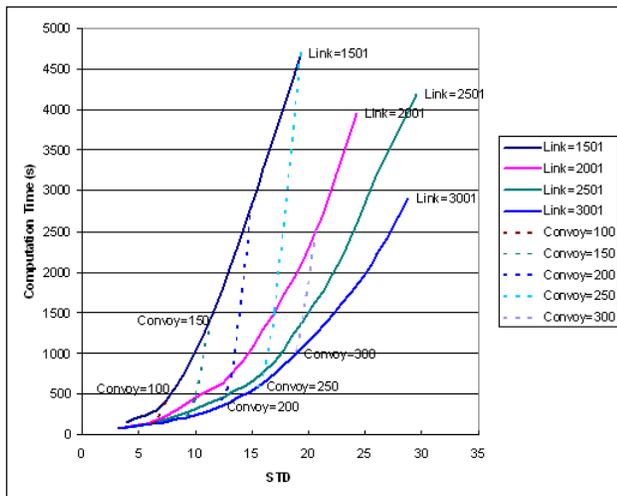


Figure 1: Graph of Computation Time

We made the following observations:

Both increasing the number of convoys and a reduction in the number of links in the graph (which indicates connectivity) results in a congested network with a high convoy/arcs ratio, thereby resulting in an increase in computation time.

The computation time increases polynomially with increase in the problem complexity (both mean and STD). In the worst case, as more convoys share the arcs in the shorter paths, we see a polynomial growth (of order ≤ 4) in computation time as the complexity (mean and STD) increases.

Keeping the number of convoys constant and reducing the number of links results in only marginal changes in the complexity mean but shows a significant increase in the complexity STD.

Measured over the parameter complexity STD, the increase in the number of convoys can be seen to have a greater impact on the computation time. This increase is however accentuated with a reduction in the links in the network graph used.

Conclusion

In this paper, we have provided a scalable and effective solution to a real convoy movement problem. The final de-

ployed system is capable of managing the movement of a large number (up to hundreds) of convoys within an urban city. Due to the low computation time of the algorithm it is also able to dynamically adapt to changes by re-running the algorithm with the incorporated problem changes.

References

- Chardaie, P.; McKeown, G.P.; Verity-Harrison, S.A.; Richardson, S.B. 2005. Solving a Time-Space Network Formulation for the Convoy Movement Problem. *Operations Research*, 53(2):219-230.
- Chih K.; Bodden M.P.; Hornung M.A.; and Kornhauser A.L. 1990. Routing and inventory logistics system: a heuristic model for optimally managing intermodal double-stack trains. *J Transp Res Forum* 31:56-62.
- Florian M.; Bushell G.; and Ferland J. 1976. The engine scheduling problem in a rail network. *INFOR* 14:121-138.
- Huapu, L.; Ye, S. (2007) 'Complexity of public transport networks' (Inst. of Transp. Eng., Tsinghua Univ., Beijing, China); Source: Tsinghua Science and Technology, 12(2):204-13.
- Iakovou, E.; Douligeris, C.; Li, H.; and Yudhbir, L. 1999. A maritime global route planning model for hazardous materials. *Transportation Science* 33(1):34-48.
- Karp R. 1972. Reducibility among combinatorial problems. *Complexity of Computer Computations*.
- Kwon, O.K.; Martland, C.D.; and Sussman, J.M. 1998. Routing and scheduling temporal and heterogeneous freightcar traffic on rail networks. *Transp Res E Logist Transp Rev* 34(2):101-115.
- Lee, Y.N.; McKeown, G.P.; and Rayward-Smith V.J. 1994. The Convoy Movement problem. Technical Report SYS-C94-01, University of East Anglia, Norwich.
- Lee, Y.N.; McKeown, G.P.; and Rayward-Smith V.J. 1994b Solving the Convoy Movement Problem using branch-and-bound on a network of transputers. In A. De Gloria M.R.J. and Marini D. (eds) *Transputer Applications and Systems* 1994, pages 786-796. IOS press.
- Lee, Y.N.; McKeown, G.P.; Rayward-Smith, V.J. 1996. The Convoy Movement Problem with initial delays. V.J. Rayward-Smith, C.R. Reeves, G.D. Smith, eds. *Modern Heuristic Search Methods*. John Wiley, Chichester, UK, 215-236.
- Qiu, L.; Hsu, W.J.; Huang, S.Y.; and Wang, H. 2002. Scheduling and routing algorithms for AGVs: a survey, *International Journal of Production Research*, 40(3):745-760.
- Nozick, L. K.; List, G.F.; Turnquist, M.A.; and Erkut, E. 1997. Integrated routing and scheduling in hazardous materials transportation. *Transportation science*. 31(3):200-215.
- Potvin, J.Y.; Ying, X.; Benyahia, I. 2006. Vehicle routing and scheduling with dynamic travel times, *Computers and Operations Research*, 33(4):1129-1137.
- Tuson, A.L.; and Harrison, S.A. 2005. Problem difficulty of real instances of convoy planning. *Journal of the Operational Research Society* 56:763-775.