8-2010

Distributed route planning and scheduling via hybrid conflict resolution

Ramesh THANGARAJOO
National University of Singapore

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

DOI: https://doi.org/10.1109/WI-IAT.2010.257

Published in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology WI-IAT 2010: 31 August-3 September 2010, Toronto: Proceedings. Los Alamitos: IEEE Computer Society. pp. 374-378.
http://doi.ieeecomputersociety.org/10.1109/WI-IAT.2010.257

Distributed Route Planning and Scheduling via Hybrid Conflict Resolution

Ramesh THANGARAJOO
The Logistics Institute Asia Pacific
National University of Singapore
Singapore
Email: tlitr@nus.edu.sg

Hoong Chuin LAU
School of Information Systems
Singapore Management University
Singapore
Email: hclau@smu.edu.sg

I. Introduction

This paper is concerned with decentralized planning and scheduling where the information for decision making resides within local agents. When considering a decentralized approach, the goal is not primarily on achieving global optimality. For instance, [Greenstadt et al. 2006] studies the tradeoff in the Distributed Constraint Optimization (DCOP) problem on efficiency, privacy and optimality. In principle, even if the problem size does allow for a centralized approach, there is still a heavy penalty on the excessive sharing of information. This penalty is a combined consequence of issues such as information security/privacy. Furthermore, if response time is critical, the network communication/latency time becomes a limiting factor. The alternative extreme is to have a fully decentralized scheme which may also not be ideal in terms of excessive negotiations (in terms of number or size of messages) needed to obtain global consistency. An interesting research challenge is to derive a reasonable balance between the two extreme approaches which best suits the problem to be tackled.

This paper is primarily concerned with a fairly generic multi-agent route planning and scheduling problem in logistics, where each agent manages its own set of jobs and is responsible for fulfilling them (via defining their routes and schedules). Conflicts may arise since agents need to share common network resources. Each agent seeks to minimize its local performance function and the goal is for agents to jointly derive a conflict-free solution that minimizes the global function, which is the sum of agent objective functions.

When deciding on how agents should interact with one another, an important criterion which is often overlooked in the literature is the relationships (or level of coupling) between two or more agents. This paper seeks to establish an effective measure of the coupling among agents managing jobs that compete for network resources. Depending on the level of coupling between agents, a hybrid conflict resolution method that involves coalition formation and distributed constraint optimization is proposed to attain an effective balance between communication efficiency, privacy and optimality.

Some examples of applications of our methodology includes convoy movement planning, [Chardaire et al. 2005] and movement planning of AGVs [Lou et al. 2009] and other transportation problems that have constraints either imposed by the transportation mode itself such as in rails systems or by the application context due to capacity constraints such as taxi-route planning at airports [Mors et al. 2009]. In this paper, we discuss the application of our proposed solution approach effectively to a real scenario where agents manage the movement of convoys. In this application, agents are assigned sets of convoys to manage, naturally decomposed into sets by means of their missions. The convoy movement problem naturally fits the decentralized optimization problem framework proposed due to the capacity restriction on roads not to have more than one convoy at any one time to avoid congestion.

II. Problem Definition

The unique property of the routing and scheduling problem addressed in this paper is that the underlying transportation network is considered a resource. Being a resource there is a constraint in terms of capacity on the utilization of any part of the network over a specific period of time. This capacity can be defined as an integer value or could be limited to a value of one as in papers [Thangarajoo et al. 2008] and others. This section will present the problem description for this paper.

The network resource is defined as a directed graph $G = (V, E)$ with node set $V = \{v_1, v_2, ..., v_g\}$ and edge set $E = \{e_1, e_2, ..., e_h\} \subset V \times V$. e_q is the arc adjoining v_q^1 and v_q^2. There exists a set of agents $A = \{a_1, a_2, ..., a_n\}$ where N denotes the index set of n agents. Each agent $i \in N$ holds a mutually-exclusive set of q_i jobs, and each job $j \in Q_i$ consists of a start node $s_{ij} \in V$ and a destination node $d_{ij} \in V$. For simplicity, let Q denote the index set for jobs.

For each agent i, α_{ij} denotes the starting time and β_{ij} denotes the route of job j from its start to destination node. Route β_{ij} is a sequence of edges $(e_{ij_1}, e_{ij_2}, ..., e_{ij_w})$ such that e_{ij_v} and $e_{ij_{v+1}}$ are connected for $1 \leq v \leq w-1$. Let $C(\beta_{ij})$ be time taken to execute the route β_{ij}, i.e.

$$C(\beta_{ij}) = \sum_{v=1}^{w} \frac{L(e_{ij_v})}{\min\{p_{ij}, S(e_{ij_v})\}} + \gamma \sum_{v=1}^{w-1} T(e_{ij_v}, e_{ij_{v+1}}) \quad (1)$$

For the first term of the equation, $L(e_{ij_v})$ is the length of edge e_{ij_v} and p_{ij} is the speed of job j of agent i. $S(e_{ij_v})$ is the maximum speed allowed on edge e_{ij_v}.

The second term of the equation captures the fact that going through short links (edges) 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 and $T(e_{ij_v}, e_{ij_{v+1}})$ indicates if there is a turn between the links e_{ij_v} and $e_{ij_{v+1}}$.

$U(\alpha_{ij}, \beta_{ij}, e')$ is the time that job j reaches the end of an edge e'. Hence, $U(\alpha_{ij}, \beta_{ij}, e') = \alpha_{ij} + C(\beta_{ij}[e_{ij_1}, e'])$.

The start and end times of a job are constrained by the release time r_{ij} and the deadline t_{ij}. That is, a job starts (i.e. a convoy departs) after its release time, i.e.

$$\alpha_{ij} \geq r_{ij} \quad (2)$$

Similarly, the job finishes at (i.e. convoy reaches the end of e" the last edge of the sequence) its destination before its deadline, i.e.

$$U(\alpha_{ij}, \beta_{ij}, e") \leq t_{ij} \quad (3)$$

The non-overlapping time constraint between any two jobs is defined as: for any two agents $g \neq h \in N$ and two respective jobs $u, v \in Q$, $(e_{gu_k}, e', e_{gu_l}) \subseteq \beta_{gu}$ and $(e_{hv_k}, e', e_{hv_l}) \subseteq \beta_{hv}$, we have,

$$\begin{cases} U(\alpha_{gu}, \beta_{gu}, e') + \frac{l_{gu}+\rho}{\min\{p_{gu}, S(e_{gu_l})\}} \geq U(\alpha_{hv}, \beta_{hv}, e_{hv_k}) \\ \quad \text{if } U(\alpha_{gu}, \beta_{gu}, e_{gu_k}) \leq U(\alpha_{hv}, \beta_{hv}, e_{hv_k}) \\ U(\alpha_{hv}, \beta_{hv}, e') + \frac{l_{hv}+\rho}{\min\{p_{hv}, S(e_{hv_l})\}} \geq U(\alpha_{gu}, \beta_{gu}, e_{gu_k}) \\ \quad \text{otherwise.} \end{cases} \quad (4)$$

where ρ is the minimum physical distance between any two jobs, l_{gu} is the length of job u from agent g and $U(\alpha_{gu}, \beta_{gu}, e')$ is the time taken for job u of agent g to reach edge e', $U(\alpha_{gu}, \beta_{gu}, e_{gu_k})$ is the time required for the job g from agent u to reach the end of the edge e_{gu_k}. Therefore, $U(\alpha_{gu}, \beta_{gu}, e_{gu_k}) = \alpha_{gu} + C(\beta_{gu}[e_{gu_l}, e_{gu_k}])$ where $\beta_{gu}[e_{gu_l}, e_{gu_k}])$ in the subsequence of the route $\beta_{gu}[e_{gu_l}, e_{gu_k}]) = (e_{gu_l}, e_{gu_{l+1},...,e_{gu_{k-1}}}, e_{gu_k})$. This constraint in essence limits the occupancy of an arc to one job at any one time. This constraint can be generalized to limit the simultaneous occupancy of an arc to a prescribed maximum capacity.

The local agent subproblem is defined as follows: for each agent i , given $\alpha_i = \{\alpha_{i1}, ..., \alpha_{in}\}$, $\beta_i = \{\beta_{i1}, ..., \beta_{in}\}$, the goal is to find vectors α_i and β_i minimizing:

$$F_1(\alpha_i, \beta_i) = \max_{j \in Q} (\alpha_{ij} + C(\beta_{ij})) - \min_{j \in Q} \alpha_{ij}. \quad (5)$$

The goal of the problem is to minimize the global objective function, which is the sum of the individual local agent objectives, i.e. to find vectors α_{ij} and β_{ij} minimizing:

$$F_2(\alpha_{ij}, \beta_{ij}) = \sum_{i=1}^{N} (\max_{j \in Q} (\alpha_{ij} + C(\beta_{ij})) - \min_{j \in Q} \alpha_{ij}). \quad (6)$$

III. Related Work

Coalition formation provides a natural option to resolve the conflicts between agents. Coalition formation works well when agents are tightly coupled, since conflicts are resolved in a centralized fashion through the formed coalition. This also means that agents need to send their information to a central server, thereby compromising on information privacy. While coalition serves to resolve conflict, unchecked or excessive coalitions will lead to the loss of the benefits of solving the problem in a decentralised manner in the first place. Taken to the extreme, when all agents form a single coalition, we in fact get the fully centralized approach. The study by [Kutanoglu & Wu 2007] analyzes the size, type, and timing of coalitions in a multi-agent production scheduling problem. The possibility of achieving a high-quality schedule with a reasonable number of iterations is investigated by controlling the coalitions performed.

An alternative to coalition formation in conflict resolution is for agents to negotiate and compromise on local objectives. Negotiation is less efficient when the level of coupling between agents is tight, since an excessive amount of information may need to be sent across agents to resolve conflicts. [Cox and Durfee 2003] for instance deals with identifying the synergy between plans of hierarchical planning agents.

Coordination and agent planning are intertwined in [Mailler & Lesser 2004] to provide a general method to solve a DCOP, called OptAPO (Optimal asynchronous partial overlay). It involves improving the value of the sub-problem owned by each agent by mediation.

IV. Solution Approach

As discussed above, when individual agent plans are conflicting, there are generally two categories of methods to resolve conflict - coalition formation and what is generally categorized as non-coalition formation methods. While both coalition and non-coalition methods have their own strengths and weaknesses; in general coalition formation methods perform better when the relationship between agents are stronger. In our problem context, this relationship is in terms of overlapping network resource requirements and coinciding schedules. Ideally thus the level of this form of inter-relationship between two agents should determine the choice of method used.

Our proposed solution approach is a hybrid method involving the combination of coalition formation and a non-coalition method (or more precisely, a DCOP algorithm) based on agent inter-relationship information. We introduce the notion of arc criticality in subsection IV-A, and measure the inter-relationship (or the level of coupling) between agents as a function of arc criticality, given in subsection IV-B. Subsection IV-C describes the logic of decision making within each agent (that seeks to maximize its own utility). Subsections IV-D and IV-E describe the coalition formation and DCOP procedures respectively, and finally subsection IV-F defines our proposed hybrid framework that combines coalition formation and DCOP methods to solve the decentralized route planning and scheduling problem.

A. Arc Criticality

Define $C(\beta_{ij})$ to be the cost of executing the route β_{ij}. If this is the lowest cost route let it be $C'(\beta_{ij})$ which can also be represented by the start and end nodes $C'(\beta(v_x, v_y)$ where v_x is the start node and v_y is the end node. For the edge in concern e_r, which is the arc adjoining nodes v_r^1 and v_r^2, $C'(\beta(s_{ij}, v_r^1) + C'(e_r) + C'(\beta(v_r^2, f_{ij})$ is the lowest cost route from the start to the finish node via e_r.

Before delving into the measurement of the agent inter-relationships, it is important to define the notion of the criticality of an arc with respect to an agent, which gives us an indication of the probability that it is a bottleneck (for the movement of convoys). More precisely, the criticality with respect to each job on an arc would be 1 if the arc falls on the shortest path for that job (moving from its start to destination). Otherwise, this value would be a fraction measured by the ratio of the shortest path distance over the shortest path distance traveled via the specific arc. Hence, the closer this value tends to one, the higher the probability that this arc would be chosen as an arc in the agent solution for performing that job.

Definition 1 (Job Criticality) For each agent $i \in N$ and job $j \in Q_i$, define the job criticality $\sigma_{e_r}(i, j)$ as

$$\sigma_{e_r}(i,j) = \frac{C'(\beta(s_{ij}, f_{ij})}{C'(\beta(s_{ij}, v_r^1) + C'(e_r) + C'(\beta(v_r^2, f_{ij})}$$

Definition 2 (Arc Criticality) For each agent $i \in N$, define the arc criticality $\sigma_{e_r}(i)$ as the sum of the job criticalities:

$$\sigma_{e_r}(i) = \sum_{j=1}^{q_i} \frac{C'(\beta(s_{ij}, f_{ij})}{C'(\beta(s_{ij}, v_r^1) + C'(e_r) + C'(\beta(v_r^2, f_{ij})}$$

B. Level of Coupling

The following steps define how the inter-relationships between agents can be computed.

1) For each arc e_r and for each agent i, compute the arc criticality $\sigma_{e_r}(i)$.
2) For each arc e_r and for each pair of agents x, y, let $\sigma_{e_r}(x, y)$ denote the joint arc criticality between x and y, which is computed as $\sigma_{e_r}(x) \times \sigma_{e_r}(y)$
3) For each pair of agents x, y, compute the level of coupling between x and y, which is the highest value $max\ \sigma_{e_r}(x, y)$ among all arcs e_r.

C. Local Solution for Agents

The following pseudo-code gives the logic of decision making within each agent. It consists basically of 2 modules (components): (1) Routing module and (2) Scheduling module. Details of this algorithm is given in [Thangarajoo et al. 2008]. The routing component utilizes a standard shortest path algorithm (such as Dijkstra). Let the function SHORTEST-PATH(v_i, v_j, D), $v_i, v_j \in V$, $D \subseteq E$, compute the shortest path from node v_i to node v_j without using the edges in D (which are the bottleneck edges to be avoided).

procedure AGENT:
$\forall j \in Q$, $D_j \leftarrow \{\}$
$\forall j \in Q$, $\beta_j \leftarrow$ SHORTEST-PATH(s_j, d_j, D_j)
$\alpha \leftarrow$ SCHEDULE-ROUTES$(\beta, r_1, ..., r_q)$
$obj \leftarrow F_1(\alpha, \beta)$
while $iter < max_I terations$ **do**
    $D \leftarrow$ UPDATE-LINKS(α, β)
    $\forall j \in Q$, $\beta'_j \leftarrow$ SHORTEST-PATH(s_j, d_j, D_j)
    $\alpha' \leftarrow$ SCHEDULE-ROUTES$(\beta', r_1, ..., r_q)$
    $obj' \leftarrow F_1(\alpha', \beta')$
    if $obj' > obj$ **then do**
        $\alpha \leftarrow \alpha'$
        $\beta \leftarrow \beta'$
    end if
    $iter \leftarrow iter + 1$
end while
output α, β
end procedure

D. Coalition Formation Procedure

The general idea of our proposed hybrid approach is to form coalitions among agents whose levels of coupling are high, and to perform conflict-resolution otherwise. In this paper, we propose a very simple coalition formation algorithm which involves combining agents whenever the level of coupling between these agents exceeds a certain threshold value. Due to the additive nature of arc criticality and level of coupling defined above, we arrive at the following proposition.

Proposition 1. When a pair of agents form a coalition, the level of coupling of the coalition with the adjoining agents cannot decrease.

This justifies performing coalitions for all pairs of tightly-coupled agents in any pairwise order. However, after a coalition, the level of coupling of the new agent with other agents proportionally increases, based on the larger number of tasks in the new agent. To reflect the accurate relationship between agents, ω_{xy} is defined as

$$\omega_{xy} = \frac{\overline{q}^2}{q_x \times q_y}\sigma_{xy}$$

Hence, our proposed coalition formation algorithm is given as follows:

1) Set γ, a threshold level for the inter-relationship level between a pair of agents.
2) Compute $\omega_{xy}\ \forall a_x, a_y \in A, x \neq y$, which indicates the inter-relationship level between all pairs of agents.
3) While there exist a pair of agents x, y such that $\sigma_{xy}\omega_{xy} \geq \gamma$ perform the following steps:
    a) Form coalition between x and y, creating a new agent.
    b) Remove x and y from set A, and replace with the new coalition agent.
    c) Re-compute $\omega_{xy}\ \forall a_x, a_y \in A, x \neq y$

E. DCOP Procedure

Our proposed method for solving DCOP is OptAPO [Mailler & Lesser 2004] (which could be replaced with other methods such as ADOPT). OptAPO has been customized to suit the routing and scheduling problem defined locally for an agent. The output of the DCOP algorithm is a conflict-free solution among agents (if one exists). The proposed algorithm can be divided into three parts, namely initialization, negotiation and reiteration.

During initialization, each agent first constructs an optimal assignment to its variables using the local search algorithm presented in subsection IV-C above and communicates the necessary information to a central server. According to relationships and possible conflicts among agents, the central server divides the agents into mediation groups [Mailler & Lesser 2004]. In our routing and scheduling problem, a possible conflict occurs when two or more agents share a common link for an overlapping time period.

During the negotiation stage, the agents in each mediation group resolve their conflicts via a mediation leader. The mediation leader is the agent with the highest autonomy in the group. The autonomy of an agent is given as a relative number to the other agents' autonomy. [Barber et al. 2000] and [Scerri et al. 2002] explore the concept of adjustable autonomy and its relationship to the behavior of agents. As described in [Lau et al. 2008], the autonomy of an agent refers to the priority of the agent's objective over another when a compromise has to be reached during conflict resolution.

1) Agents receive the information of the member agents in their mediation group, and their autonomy values.
2) Within each group, agents send their information to their mediation leader, which includes its schedules and routes, its identification and autonomy level, the release times of its convoys, and its intention to mediate.
3) The agents which have sent their information to their mediation leader will set their mediation flag as active.
4) The mediation leader, upon receiving all information from the group, will perform the mediation to resolve conflicts between the agents. The starting time of the convoys are defined by scheduling the convoys onto their previously defined routes firstly in order of the autonomy of the agent and secondly by the individual convoy release time. If any conflicts are detected during scheduling a particular convoy (with the set of scheduled convoys), the release time of the convoy is delayed to avoid the conflict.
5) Similar to the agent's local solution algorithm, the mediation leader's improvement algorithm will focus on minimizing the objective function. Hence, it will randomly choose an agent, from the list of related agents to improve on. The randomly chosen convoy to re-route could be a convoy from another agent where the conflict was detected.
6) The final results are sent to the agents.

The reiteration stage involves the agents updating their respective revised local solutions to the central server who checks for further possible conflicts and repeating the negotiation stage if necessary:

1) Agents change their mediation status to passive. This allows them to be involved in another mediation if necessary.
2) Agents will consolidate the Initialization Information as defined in the initialization phase and send them to the central server.
3) If new relationships are formed the process will reiterate the negotiation phase

F. Hybrid Framework

Finally, with the above discussion, we present our proposed hybrid framework, which is a 2-stage method that combines the strengths of the coalition and DCOP method. In the first stage, we form coalitions among agents based on their coupling level. This is followed by stage two that performs conflict-resolution via solving a DCOP.

V. Results

In this section, we show the results of comparison on large-scale random instances of the convoy routing and scheduling problem.

A network graph with 1000 nodes was generated with 3000 links. Each agent solves a 5 or 10 convoy routing and scheduling problem, generated with the coupling for the problem measured by the mean ($\overline{\omega}$) and standard deviation ($STD(\omega)$).

We compare a spectrum of hybrid approaches by varying the value of γ from 0.9 $\overline{\omega}$ to 1.2$\overline{\omega}$. $\gamma = 0$ indicates a centralized approach while γ=2$\overline{\omega}$ will indicate a pure distributed approach. We show results of comparison on the quality of solutions (measured by objective F_2), efficiency (measured by CPU time for mediation and local agent search algorithm), as well as communication efficiency (measured by the number of messages between agents). From the standpoint of privacy, the centralized approach represents the worst approach since all agent information need to be sent to a central server; while the pure distributed approach represents the best approach.

Problem Details.	Setting	Objective	CPU Time/s		Mess. Sent
	γ	$F_2(\alpha, \beta)$	Med.	Local Search	
Convoys	0	124297	0	152.5	0
/Agent=5	0.9$\overline{\omega}$	121184	495.91	61	16
	$\overline{\omega}$	139252	524.38	76.3	20
$\overline{\omega} = 12.75$	1.1$\overline{\omega}$	121435	467.88	76.3	28
$STD(\omega)$	1.2$\overline{\omega}$	103470	519.94	61	40
=1.95	2$\overline{\omega}$	85271	583.11	30.5	58
Convoys	0	239189	0	304	0
/Agent=10	0.9$\overline{\omega}$	226644	625.86	273.6	10
	$\overline{\omega}$	209782	1718.38	212.8	22
$\overline{\omega} = 51.67$	1.1$\overline{\omega}$	191428	1581.77	91.2	40
$STD(\omega)$	1.2$\overline{\omega}$	169763	1566.56	60.8	52
=5.18	2$\overline{\omega}$	167301	1804.31	30.4	58

Table I
HYBRID COMPARISON

From the solution quality standpoint, the pure distributed approach performs best. This is because when two agents are combined in a coalition, there is a natural loss in the individual agent objective. Since search is performed in a distributed fashion, the relevant computation time is dependent on the size of the largest agent. Hence, local search CPU time tends to decrease with an increase in γ values.

We also note a superlinear increase in the number of messages sent (Mess. Sent) as we increase the value of γ. The results show that from the standpoint of balancing communication efficiency and solution quality, the ideal γ value varies between 1.1$\overline{\omega}$ and 1.2$\overline{\omega}$.

VI. IMPLICATION OF RESULTS

The results above show that decision support is required for the convoy routing and scheduling problem to balance the tradeoff between privacy, communication efficiency and optimality well. Each mission of convoys (represented by agents, with their own local objective) only share information with other agents to avoid conflicts in their convoys' movement plan. This is due to the cost of transferred information between agents. This cost is measured by the importance given to the proactive enforcement of communication silence in periods of tension. Maintaining privacy however has an impact on communication efficiency. By limiting the information being transferred, more messages may be required between agents to maintain an equivalent level of optimality. However, constraints due to network limitations also common in a wartime scenario, may eventually lead to optimality being compromised.

Acknowledgment. We thank K. M. Ng for comments and J. Zhou for coding the system.

REFERENCES

[Barber et al. 2000] Barber K.S., Goel A., and Martin C.E.,2000. Dynamic Adaptive Autonomy in Multi-Agent Systems. *Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Autonomy Control Software* 12(2):129-147.

[Chardaire et al. 2005] P. Chardaire et al. (2005). Solving a Time-Space Network Formulation for the Convoy Movement Problem. *Operations Research*, 53(2):219-230.

[Cox and Durfee 2003] J. S. Cox and E. H. Durfee (2003). Discovering and Exploiting Synergy Between Hierarchical Planning Agents. In Proceedings of *AAMAS*, Melbourne, Australia, 281-288.

[Greenstadt et al. 2006] R. Greenstadt, J. P. Pearce and M. Tambe (2006). Analysis of Privacy Loss in Distributed Constraint Optimization, In Proceedings of *AAAI*, Boston, Massachusetts, 2006.

[Kutanoglu & Wu 2007] E. Kutanoglu and S.D. Wu (2007). Coalitions in coordinated multi-agent production scheduling: A computational study. *Journal of Manufacturing Systems*, 26, 1221.

[Lau et al. 2008] H. C. Lau, L. Agussurja and R. Thangarajoo (2008). Real-time supply chain control via multi-agent adjustable autonomy. *Computers & OR*, 35(11): 3452-3464.

[Lou et al. 2009] P. Lou, X. Wu, and J. Wang (2009) Path Planning and Control for Multiple AGVs Based on Improved Two-Stage Traffic Scheduling. *Int. J. of Automation Technology*, 3(2):157-164.

[Mailler & Lesser 2004] R. Mailler, V. R. Lesser (2004). Solving Distributed Constraint Optimization Problems Using Cooperative Mediation. In Proceedings of *AAMAS*, 438-445.

[Mors et al. 2009] A. W. ter Mors and J. van Belle and C. Witteveen (2009). Context-Aware Multi-Stage Routing. In Proceedings of *AAMAS*, Budapest, Hungary, 49-56.

[Scerri et al. 2002] P. Scerri, D. V. Pynadath and M. Tambe (2002). Towards Adjustable Autonomy for the Real World. *Journal of AI Research*, 17, 171-228.

[Thangarajoo et al. 2008] R. Thangarajoo, L. Agussurja and H. C. Lau (2008). A Hybrid Approach To Convoy Movement Planning in an Urban City. In Proceedings of *AAAI/IAAI*, 1738-1744.