

Hierarchical Multi-agent Distribution Planning

Tony Allard and Slava Shekh

Command, Control, Communications and Intelligence Division,
Defence Science and Technology Organisation, Adelaide, Australia
{tony.allard,slava.shekh}@dsto.defence.gov.au

Abstract. Logistics and distribution planning are an important element of military operational planning. The vast level of detail available during planning increases the difficulty in forming a complete solution within time constraints. Moreover, the consideration of multiple planning scenarios is paramount to successfully investigate all available contingencies. This paper presents a hierarchical multi-agent approach to distribution planning. The use of a multi-agent system enables a distribution problem to be logically separated, with each part being delegated to an independent agent. The application of hierarchy allows agent communication to be regulated, reducing redundancy in communication, as encountered in a flat multi-agent structure. Through multiple trials, it was found that the application of hierarchy vastly improved the computational performance of the system, without compromise to solution quality.

Keywords: distribution planning, multi-agent systems, agent hierarchy.

1 Introduction

A complete and robust distribution plan underpins a successful military operation. Logistics and distribution planning have become of central importance to operational planning. Logistics with respect to defence is defined as force projection and sustainment of forces [1] and distribution as the act of satisfying those logistical requirements. Therefore the distribution problem is defined as the satisfaction of deployment and redeployment activities (force projection) and the movement of sustainment stock (sustainment). A distribution plan is formed by constructing and assigning a schedule to each transport, collectively providing one possible solution to the distribution problem. Each schedule consists of a mapping of personnel, equipment and sustainment to movements conducted by a specific transport at a given point in time. The aim of a distribution plan is usually to achieve the movements required in the shortest time possible with the lowest cost. Solving the distribution problem is best described as the difficulty of managing cost and uncertainty [2].

The application of intelligent agents in software systems are becoming increasingly attributed as “capable of exhibiting flexible problem-solving behaviour” [3]. When the distribution problem is viewed with the mindset of employing an intelligent agent approach, one cannot help but see the possibility of using multiple agents [4]. This is because the problem can be easily viewed as a set of self-interested agents, some with

tasks that require completion and others with resources to complete those tasks. In this paper we consider each transport as an individual agent with goals to complete distribution tasks in the shortest duration. A single agent handles task announcement and allocation using a bidding process, the details of which will be explained later. Through multi-agent negotiation, agents collaborate to formulate the distribution plan. As the number of agents increase so too does the communication overhead, increasing the time taken to produce a solution to the distribution problem. Since the agents themselves can be described by the transports they represent and the capabilities of those transports, we propose grouping the agents by capability. In this way a single agent can represent a collection of transports and a hierarchical structure of agents is developed. This paper presents a two-tiered hierarchical multi-agent approach to solving the distribution problem, modelling agents:

- at the micro level as individual transportation assets, and
- at the macro level as representations of homogeneous collections of transport capabilities

A hierarchical multi-agent system (MAS) is evaluated for the quality of the solution and the computational time taken to achieve the solution against a set of example scenarios. The distribution problem can be simplified to a Vehicle Routing Problem (VRP), which has been shown to be NP-Hard [5]. As a result, optimal solutions to the real-world scenarios considered in this paper could not be computed. However the solutions produced by the hierarchical MAS were compared to those produced manually and found to be of similar quality.

This paper focuses on the solution to the distribution problem from the point of view of efficient allocation of resources. Therefore force projection and sustainment are modelled as fixed time events. These events represent a quantity of personnel, equipment and sustainment (cargo) to be moved from one location to another by one or more transportation assets with set temporal constraints. The solution space of the distribution problem consists of transports, routes (modes) and bases (nodes) each with varying capabilities, as well as the possibility of unavailability time; such as maintenance. Therefore the logistical planner's job is one of data acquisition and analysis, resource allocation, risk mitigation and optimisation. This paper will discuss a solution to the resource allocation problem and optimisation, for an insight into risk analysis see [6].

2 Related Work

Multi-agent techniques have been used for planning and scheduling in a wide range of domains. Andreev, Rzevski et al. [7] used agents to model the drivers and vehicles of a Rent-a-Car company, while [8] applied multi-agent concepts to the newspaper industry. In these cases, a multi-agent system provided problem abstraction and efficiency in generating schedules, both of which would be difficult to achieve with a centralised approach. Agents provide a way of dividing responsibility among multiple entities and negotiation between agents can be used to formulate schedules. One

negotiation technique is auction-based allocation, where tasks are allocated based on the best bids made by agents. [9] used this approach to model auctions with various trucking companies for deliveries in transportation logistics.

Military operations are generally very complex in nature, and modelling them is therefore a difficult task. Multi-agent systems provide an effective way of dealing with this problem. Agents can be used to represent various entities, such as transports, resources and units, including both friendly and hostile forces. The basic behaviours of these entities can then be modelled, which provides a platform for observing interactions between entities and utilising the resultant outputs. This approach has been applied to numerous types of military operations, including military combat [10], crisis management [11] and logistics [12]. The system described by [12], called UltraLog, utilises the interaction and collaboration of over a thousand planning agents to develop a military logistics plan.

Despite the ability of the MAS to provide problem abstraction, modelling a large system can result in a significant communication overhead. As the number of agents increases so does the number of communication channels and hence the effort required to search [13]. One strategy for dealing with this is to incorporate a predetermined hierarchy into the agent structure. [14] presents a hierarchical multi-agent approach for modelling work schedules on a construction site. The hierarchy consists of multiple tiers of agents, which task each other in building a structure, from the low-level agents that lay the bricks to the high-level agents that coordinate the construction. Similarly, the MAS presented in this paper uses a hierarchical structure (albeit not pre-determined) to allocate responsibility; high-level agents are responsible for broad planning through negotiation, while low-level agents perform detailed planning of their respective transportation requirements. The presence of fewer agents at the higher levels of the hierarchy results in less communication overhead.

A hierarchical agent structure was used by [15] to resolve conflicts amongst agents functioning as subject matter experts (SME). The agents were clustered together based on the data they used to make decisions; agents with common data were placed in similar clusters. The MAS was structured as a three-tiered hierarchy with each cluster headed by an agent that facilitated communication within the cluster and between different clusters. The top of the hierarchy was a single agent responsible for making the final decision based on the suggested decisions provided by each cluster. This implementation focused on reducing conflicts between agents and improving the accuracy of the system. While this reduced some communication overhead, if these SME agents are viewed as operating on similar capabilities, then this paper seeks to utilise the hierarchy to episodically filter agents whose capabilities are not suitable for a particular distribution problem.

Initial schedules formulated by agents can be sub-optimal, and improvements become more evident as the schedules are populated. Thus, our system incorporates techniques for schedule optimisation, similar to the transport load consolidation in the system described by [16]. Schedule optimisation allows cargo to be delivered quicker using fewer resources, which is an important factor given the limited transportation assets that are often available during military operations.

3 Problem Definition

The distribution problem defined in this paper is modelled as a finite set of jobs, $J := \{j_1, \dots, j_n\}$, each with an amount of cargo requiring transportation from an origin to destination within fixed temporal boundaries. Let L be the finite set of geographical locations, which include the origin and destination of these jobs, $L := \{l_1, \dots, l_p\}$. Each job, j_i , can then be represented by the tuple $\langle l_{j_i}^{origin}, l_{j_i}^{destination}, t_{j_i}^{start}, t_{j_i}^{finish}, C_{j_i} \rangle$ where, $t_{j_i}^{start}$ and $t_{j_i}^{finish}$ are the earliest time j_i can be started and the latest time j_i can be completed respectively. $l_{j_i}^{origin} \in L$ is the location of j_i 's cargo at time $t_{j_i}^{start}$, while $l_{j_i}^{destination} \in L$ is the desired location of j_i 's cargo at time $t_{j_i}^{finish}$. $C_{j_i} := \{c_{j_i1}, \dots, c_{j_in}\}$ is the finite set of cargo for j_i ; where, each element, c_{j_ik} , is the volume and mass representation of the k^{th} cargo element.

Each geographical location is connected to other locations through zero or more routes. Let $R := \{r_1, \dots, r_h\}$ be the finite set of routes for the distribution problem. Each of these routes can support a single mode of transportation within the fixed set $M := \{road, rail, air, sea\}$. Each route, r_b , is defined by the tuple $\langle l_{r_b}^{origin}, l_{r_b}^{destination}, m_{r_b}, d_{r_b} \rangle$ where, $l_{r_b}^{origin} \in L$ and $l_{r_b}^{destination} \in L$ are the start and end geographical locations of r_b ; $m_{r_b} \in M$ is the mode of r_b and d_{r_b} is the distance of r_b . Two locations can be connected by a sequence of routes, where each route can potentially involve a different mode of transportation. Hence, a single job may involve traversing routes of multiple modes.

The distribution problem also models transportation assets as the fixed set $V := \{v_1, \dots, v_m\}$. Each transport v_x is represented by the tuple $\langle \omega_{v_x}, d_{v_x}, m_{v_x}, c_{v_x} \rangle$ where, ω_{v_x} and d_{v_x} are the maximum speed and distance v_x is capable of travelling respectively; $m_{v_x} \in M$ is the mode of transportation v_x can travel and c_{v_x} is the volume and mass representation of the amount of cargo v_x can carry.

Distribution planning is the allocation of transport assets to specific jobs. A job, j_1 , is completed when its cargo C_{j_1} is transported from $l_{j_1}^{origin}$ to $l_{j_1}^{destination}$. Transportation of C_{j_1} cannot begin prior to $t_{j_1}^{start}$ and for j_1 to be successful all cargo must arrive at $l_{j_1}^{destination}$ no later than $t_{j_1}^{finish}$. A transport, v_1 , can only participate in the completion of j_1 if there is some item in C_{j_1} that fits within the cargo capacity of v_1 . When moving between two locations v_1 can only travel on the set of routes $R_\mu \subseteq R$, where $\forall y \in R_\mu, (m_y = m_{v_1}) \wedge (d_y \leq d_{v_1})$. v_1 can transport j_1 's cargo directly to the $l_{j_1}^{destination}$ or partway to a new location, l_k . If v_1 transports the cargo partway, another transport, v_h , is required to transport the cargo the remaining distance (l_k to $l_{j_1}^{destination}$) or to another intermediary location (l_k to l_{k+1}). This cooperative style of transportation may be required if $l_{j_1}^{origin}$ and $l_{j_1}^{destination}$ are separated by routes of varying modes.

The distribution problem is solved when all jobs are completed successfully. It may not be possible to complete a job within a single trip, so a transport may need to perform multiple trips, or more than one transport may need to be used. Transports can also participate in multiple jobs at once.

4 Multi-agent Negotiation

We begin by looking at a non-hierarchical (flat) agent structure, which models agents as individual transportation assets. Let $A := \{a_1, \dots, a_n\}$ be the fixed set of agents in the system. Each agent is aware of its capabilities as a transport, v_{a_i} ; its current schedule, Δ_{a_i} ; the locations it can travel, L_{a_i} and the other agents in the system with which it can communicate, A_{a_i} . Therefore an agent a_i can be represented by the tuple $\langle v_{a_i}, L_{a_i}, \Delta_{a_i}, A_{a_i} \rangle$. In this flat structure each individual agent is capable of communicating with all other agents in the system, meaning $A_{a_i} = \{a_i\}^c \cap A$.

The process of solving a distribution problem is coordinated by the Bid Controller (BC), with the goal of formulating an efficient allocation of transport resources. The BC has access to the set of jobs, J , and processes J in order of $t_{j_i}^{start}$, announcing each job to all agents in the system. The agents form their best bid for each job and submit that bid to the BC. This bidding process is illustrated in Fig. 1, and discussed in more detail below.

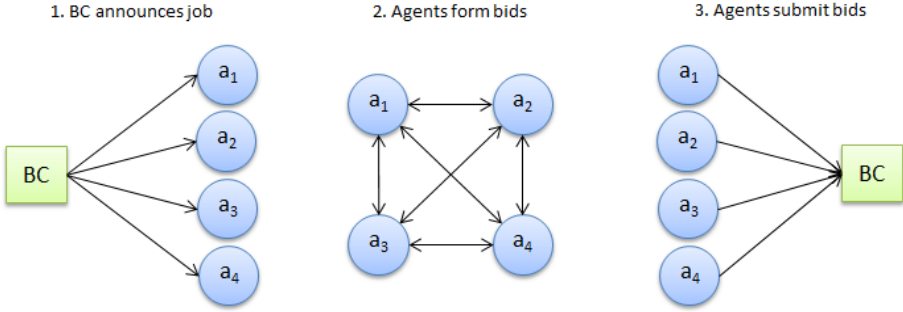


Fig. 1. The bidding process in the flat agent structure

When the BC announces a new job, j_1 , the goal of each agent is to transport as much of C_{j_1} as possible from $l_{j_1}^{origin}$ to $l_{j_1}^{destination}$ before $t_{j_1}^{finish}$. To achieve this each agent computes a series of bids for j_1 and selects its most efficient bid to respond to the announcement.

A bid is a proposed movement or set of movements to complete j_1 , either in part or in full. a_1 can form a bid to transport C_{j_1} directly from $l_{j_1}^{origin}$ to $l_{j_1}^{destination}$, or if it is unable to reach $l_{j_1}^{destination}$, it may transport C_{j_1} to an intermediate location l_k and subcontract the remainder of the journey. Once a_1 decides to subcontract, it will send a new sub-job announcement to A_{a_1} outlining the requirement to transport C_{j_1} from l_k to $l_{j_1}^{destination}$. The agents that receive the announcement will then undergo the same bidding process as described above with a_1 . Concretely, each agent a_i in A_{a_1} will consider completing the sub-job themselves, or subcontracting further, depending on what is likely to result in the most efficient bid.

A bid that contains more than one movement is termed a composite bid. Composite bids are limited in size to no more than six movements or the number of transports in the problem, whichever is smaller. This means that for a bid to be valid it must complete the transportation of C_{j_1} from $l_{j_1}^{origin}$ to $l_{j_1}^{destination}$ utilising no more than six transportation assets. The imposition of a subcontract limit on the MAS limits the search space, preventing agents from pursuing intractable solutions. A subcontract limit of six was found to be a good compromise between computation time and solution quality.

Each agent, a_1 , creates bids by iterating over L_{a_1} and A_{a_1} and attempts to move the maximum amount of cargo it can carry to each location, l_k , initiating subcontracting when $l_k \neq l_{j_1}^{destination}$. This follows a heuristic search which prunes branches of the search tree that can be seen to offer less efficient results.

Once all bids have been generated, a_1 responds to the job announcement with its most efficient bid. Efficiency is determined by the expected completion time of the job, according to the bid. If two bids have the same expected completion time, the agent will use the bid with the lowest transport utilisation time. Bids that involve fewer transports for shorter periods of time will have lower utilisation times and are considered more desirable. The system itself maintains a set of bids received by agents and selects the best bid that satisfies the same efficiency criteria. All agents involved in the best bid will then have their schedules updated. If b_y does not transport all of C_{j_1} from $l_{j_1}^{origin}$ to $l_{j_1}^{destination}$, the system will create a new job announcement to transport the remaining cargo, at which point the bidding process begins again.

5 Benefits of Hierarchy

The MAS described above enables each agent to communicate or subcontract to any other agent in the system. This is termed a flat communication structure (FCS); meaning that for each agent in the system $A_{a_i} = \{a_i\}^c \cap A$. Therefore if N_a is the number of agents in the MAS, then each agent has $N_a - 1$ communication links. If ρ is the subcontract limit, then the worst case number of iterations of subcontracting is $\prod_{\varepsilon=1}^{\rho-1} (N_a - \varepsilon)$. If the MAS iterates over the set of jobs, J , and each job requires, at worst case, each agent to iterate over all locations, L , then the worst case computational complexity of the problem is $J \times L \times N_a \times \prod_{\varepsilon=1}^{\rho-1} (N_a - \varepsilon)$. Using this equation, two of the sets can be held constant while the other is increased. The effect of each set on the computational complexity is shown in Fig. 2. Computational complexity is modelled in computational units; an abstract unit representing a single iteration over any of the three sets. Fig. 2 shows that an increase in the number of jobs or locations in the system results in a linear increase in computational complexity. However as the number of agents in the system increases, the computational complexity increases exponentially. This exponential increase is not surprising, since N_a agents could have up to $N_a - 1$ communication links.

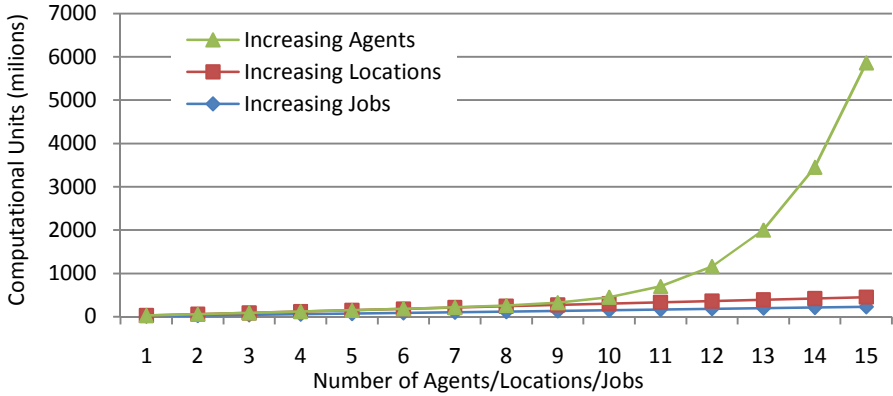


Fig. 2. Effect of system elements on computational complexity

It can therefore be said that the number agents in the MAS, or indeed the number of communication links between them, is the dominant factor affecting computational complexity. The most effective method for reducing computational complexity is to reduce agent communication or the communication overhead. This paper suggests logical grouping of agents to isolate communication between groups and the development of these groups into a hierarchical communication structure (HCS) to better direct communication. Agents were grouped based on the capabilities of the transport asset they were modelling, with similar capabilities grouped together. A new type of agent, transport agent T , models the head of an agent group and directs communication down the hierarchy. This hierarchy is illustrated in Fig. 3, providing an alternative communication structure to step 2 of Fig. 1.

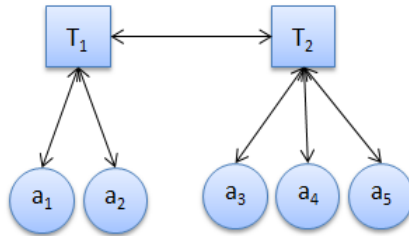


Fig. 3. The hierarchical agent communication structure

Bidding, communication and hence subcontracting are performed at the transport agent level. Transport agents select the most suitable agent in their group to participate in a bid. The original equation for worst case iterations of subcontracting thus becomes $N_t \times \prod_{\varepsilon=1}^{\rho-1} (N_t - \varepsilon)$ where N_t is the number of transport agents, and $N_t \ll N_a$. Each transport agent can be thought of as a different capability group within the MAS. The exponential growth of the computational complexity can now be limited by the ability to divide agents into capability groups. As will be shown in the next section the HCS results in a large reduction in computational complexity and hence computation time when compared with the FCS.

The HCS solution quality depends on the degree of similarity between transports within capability groups. When capability groups are characterised by specific platform type (e.g. C-130H aircraft, C-17 aircraft, etc.) the solutions of both MAS structures should be equivalent since the Transport Agent exactly represents its constituents. However when capability definitions are generalised (e.g. aircraft, ships, trucks, etc.) the solutions produced by the HCS tend to reduce in quality when compared to those produced by the FCS. As capability definitions become more abstract the quality of the HCS will continue to degrade with a corresponding increase in computational performance. This degradation in quality may require the use of an automated plan repair functionality to correct the errors introduced. This paper considers only specific transport capability groups, while abstract capability representations and plan repair techniques are areas for future work.

6 Results

Testing was completed in two parts; one assessing computational complexity and the other testing the quality of the solution.

Computational complexity was assessed using two distribution problem scenarios that were similar to circumstances faced by military planners. The two scenarios differed in their route network; the first was designed with a fully connected route network allowing each individual transport type to complete a problem in full. The second was designed with a partially connected route network where agents were required to subcontract to achieve their goals. Specifically, in the partially connected network, each location could only be reached, on average, by two routes. For example, Location C might only be reachable by sea from A, or by air from B. This created a need for agents to cooperate in the partially connected network, because most jobs could not be completed alone. Conversely, the fully connected network had no such restriction and each agent could operate independent of other agents.

Each distribution problem consisted of a set of fixed time events requiring the movement of supplies from multiple geographical locations. Three different transport asset capabilities were initialised in groups at random locations. The number of transports of each type was varied from one to 25 during testing with each amount tested 100 times. Both the HCS and FCS attempted to solve each of these distributions problems. When a solution was found, each communication structure produced the same solution however the time required to reach such a solution varied, especially when dealing with increasing instances of each transport capability. Solutions produced by both structures were of equal quality due to the use of specific transport capability groups in the HCS as described in Section 5.

The computation time for various problems is shown in Fig. 4 and Fig. 5 for fully connected and partially connected networks respectively. These graphs use a logarithmic time scale due to the rapid growth in computation time of the FCS. The vertical bars show the standard deviation of each result. Due to time constraints, the FCS was unable to complete instances of the problem comprising of more than four transports in each capability for a fully connected route network, and seven transports in each capability for a partially connected route network. These instances are therefore not shown in the graphs.

The quality of the solution was also tested against a centralised approach for distribution planning, known as the Greedy Distribution Algorithm (GDA) [17]. Instead of using a multi-agent system, the GDA contains a central entity that considers the constraints of the planning scenario and formulates a distribution plan by applying heuristics. The use of heuristics is driven by the inability of exact methods, such as integer programming, to scale effectively to larger problems [18].

The solutions produced by the HCS were compared to those produced by the GDA algorithm for the same problem set, as shown in Table 1. These problems are based on real world distribution scenarios faced by military planners, and for this paper have been labelled S1-S10. The quality of the solutions is measured by the makespan of the schedule generated by the algorithm. The makespan is the time difference between the start of the first movement and the end of the last movement. Shorter makespans are desired for a given distribution problem as it means the movements were completed in a shorter duration.

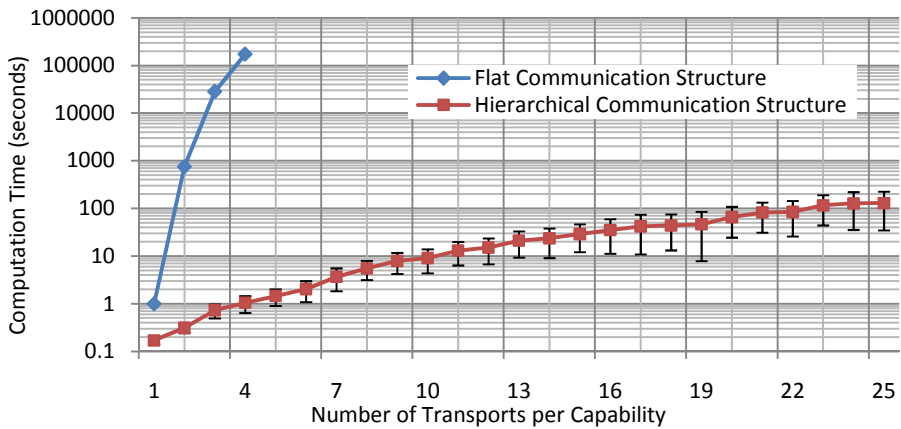


Fig. 4. Distribution problem computation time for a fully connected route network

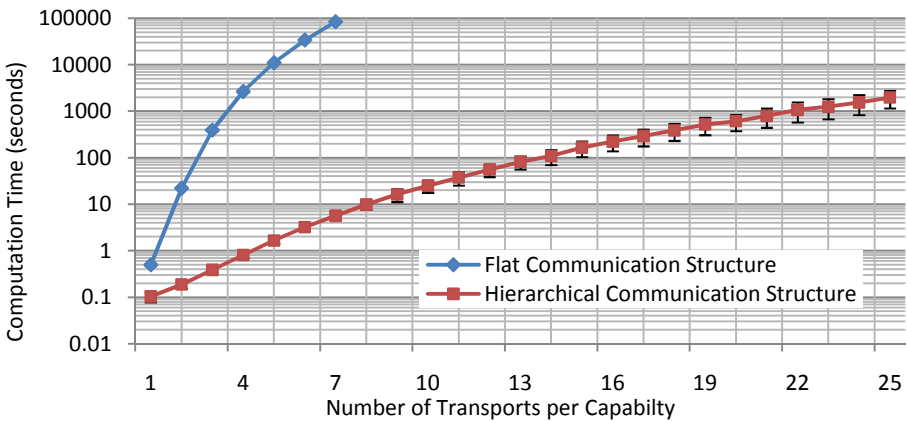


Fig. 5. Distribution problem computation time for a partially connected route network

Table 1. Distribution plan makespans for the HCS and GDA

Scenario	Hierarchical Communication Structure	Greedy Distribution Algorithm
S1	52 days, 14 hours, 41 minutes	68 days, 6 hours, 34 minutes
S2	21 days, 11 hours, 9 minutes	23 days, 21 hours, 32 minutes
S3	182 days, 5 hours, 51 minutes	199 days, 21 hours, 51 minutes
S4	10 days, 7 hours, 43 minutes	17 days 15 hours 40 minutes
S5	37 days, 4 hours, 12 minutes	37 days 12 hours 6 minutes
S6	121 days, 17 hours, 15 minutes	140 days 22 hours 9 minutes
S7	31 days, 8 hours, 9 minutes	31 days 12 hours 24 minutes
S8	70 days, 9 hours, 22 minutes	80 days 20 hours 25 minutes
S9	48 days, 8 hours, 46 minutes	50 days 11 hours 42 minutes
S10	21 days, 5 hours, 41 minutes	21 days 12 hours 36 minutes

7 Discussion

The results in Fig. 4 show a comparison of computation times for both communication structures in a fully connected route network. Fig. 5 shows a similar comparison for a partially connected route network. In both network configurations, the FCS performed poorly, growing at a near exponential rate. Due to this exponential growth, calculating a result became infeasible for transport quantities above four in fully connected networks, and seven in partially connected networks. For example, a fully connected network with four transports per capability using the FCS took approximately 48 hours to compute one solution. In contrast, the HCS demonstrated improved scalability, suggesting that it is better suited for handling the complexity of real-world scenarios. A solution could be found for route networks with up to 25 transports per capability in an acceptable timeframe, faster than the FCS could compute a solution for four transports per capability.

The poor computational performance of the FCS can be attributed to the large number communication links. Every agent can potentially communicate with every other agent, regardless of previous communications between agents of a similar capability. This leads to redundant communication that costs computation time without improving the quality of the solution. The HCS exhibits a distinct improvement in computational performance. This is due to the structure's logical isolation of similarly capable agents, thereby reducing redundant communication and allowing negotiation between agents to be better focused on improving the global solution. This is evident in Fig. 4 and Fig. 5 where the HCS consistently outperformed the FCS, despite both approaches producing solutions of equal makespan.

Fig. 4 displays a non-uniform computation time slope for the HCS, becoming increasingly evident as the number of agents per capability increases. This is caused by the lack of restriction in the route network. The ability of any transport to travel to any destination lends itself to many different solutions for the same problem. This requires a large number of samples for a statistically significant result. While the gathered data still reflects the scalability of the algorithm, a larger sample set could improve the uniformity of the computation time slope. Gathering more results and further investigating the progression of the computation time slope for the MAS implementation is an area for future research and is beyond the scope of this paper.

Table 1 shows the comparative performance of the HCS and GDA. The GDA utilises a greedy algorithm designed to produce solutions in a timely manner at the expense of solution quality. In contrast the HCS employs a more comprehensive search of the solution space enabling it to provide better quality solutions at the expense of computational performance. To overcome performance issues the HCS utilises hierarchy to logically group agents and reduce communication overhead. As a result the HCS was able to produce better quality solutions when compared with the GDA, while remaining comparable in computational performance.

8 Conclusion

This paper presents a hierarchical MAS for solving logistic distribution problems. From trials conducted against real-world distribution scenarios it was shown that the HCS dominates the FCS in its ability to produce efficient solutions in considerably less computation time. As the scenario size increases the FCS becomes intractable, while the HCS remains computationally efficient, without loss of solution quality. The efficiency of the HCS is the result of its ability to isolate similarly capable agents and better direct communication.

The hierarchical MAS presented was able to provide better quality solutions to distribution scenarios than algorithms developed with similar intent. When solutions from both the HCS and GDA were compared it was found that the HCS was able to produce on average better quality solutions with similar computational performance. This supports the claim that HCS can produce quality solutions to complex real-world problems within acceptable timeframes.

Future research is directed to further increasing the computational performance of the MAS, either through improved agent communication or applied heuristics. Another area of interest is the application of the MAS in dynamic replanning and plan repair as a computer-based decision aide for human planning.

Acknowledgements. The authors would like to acknowledge Don Gossink, Luke Marsh and Hing-Wah Kwok for their support, guidance and contributions to this research project.

References

1. Tuttle, W.G.T.: Defense logistics for the 21st century. Naval Institute Press, Annapolis (2005)
2. Simchi-Levi, D., Bramel, J., Chen, X.: The logic of logistics: theory algorithms and applications for logistics and supply chain management. Springer, New York (2005)
3. Jennings, N.R.: An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM 44 (2001)
4. Perugini, D., Wark, S., Zschorn, A., Lambert, D., Sterling, L., Pearce, A.: Agents in Logistics Planning - Experiences with the Coalition Agents Experiment Project. In: 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (2003)
5. Lenstra, J.K., Rinnooy Kan, A.H.G.: Complexity of Vehicle Routing and Scheduling Problems. Networks 11, 221–227 (1981)
6. Thiagarajan, R., Kwok, H., Calbert, G., Gossink, D., Shekh, S., Allard, T.: A simulation-based risk analysis technique to determine critical assets in a plan. In: 19th International Congress on Modelling and Simulation, pp. 503–509 (2011)
7. Andreev, S., Rzevski, G., Shviekin, P., Skobelev, P., Yankov, I.: A Multi-agent Scheduler for Rent-a-Car Companies. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 305–314. Springer, Heidelberg (2009)
8. Böhnlein, D., Schweiger, K., Tuma, A.: Multi-agent-based transport planning in the newspaper industry. International Journal of Production Economics 131, 146–157 (2011)
9. Robu, V., Noot, H., La Poutre, H., Van Schijndel, W.J.: A multi-agent platform for auction-based allocation of loads in transportation logistics. Expert Systems with Applications 38, 3483–3491 (2011)
10. Malhotra, A.: Agent-Based Modeling in Defence. DRDO Science Spectrum, 60–65 (March 2009)
11. Kaddoussi, A., Zoghalmi, N., Hammadi, S., Zgaya, H.: An agent-based distributed scheduling for military logistics. In: 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 59–64 (2011)
12. Carrico, T., Greaves, M.: Agent Applications in Defense Logistics. In: Defense Industry Applications of Autonomous Agents and Multi-Agent Systems, pp. 51–72 (2008)
13. Perugini, D., Jarvis, D., Reschke, S., Gossink, D.: Distributed Deliberative Planning with Partial Observability: Heuristic Approaches. In: International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS), pp. 407–412 (2007)
14. Molinero, C., Núñez, M.: Planning of work schedules through the use of a hierarchical multi-agent system. Automation in Construction 20, 1227–1241 (2011)
15. Wakulicz-Deja, A., Przybyla-Kasperek, M.: Hierarchical Multi-Agent System. In: Recent Advances in Intelligent Information Systems, pp. 615–628 (2009)
16. Baykasoglu, A., Kaplanoglu, V.: A multi-agent approach to load consolidation in transportation. Advances in Engineering Software 42, 477–490 (2011)
17. Marsh, L., Gossink, D.: Exploiting Symmetries in Logistics Distribution Planning. In: 19th International Congress on Modelling and Simulation, pp. 482–488 (2011)
18. Marsh, L.: A Study on Optimisation Techniques for Logistics Distribution. School of Mathematical Sciences. The University of Adelaide, Adelaide (2008)