A robust, distributed task allocation algorithm for time-critical, multi agent systems operating in uncertain environments

A Robust, Distributed Task Allocation Algorithm for Time-Critical, Multi Agent Systems Operating in Uncertain Environments

Amanda Whitbrook, Qinggang Meng and Paul W. H. Chung

a.whitbrook@derby.ac.uk, q.meng@lboro.ac.uk,
p.w.h.chung@lboro.ac.uk

Abstract. The aim of this work is to produce and test a robust, distributed, multi-agent task allocation algorithm, as these are scarce and not well-documented in the literature. The vehicle used to create the robust system is the Performance Impact algorithm (PI), as it has previously shown good performance. Three different variants of PI are designed to improve its robustness, each using Monte Carlo sampling to approximate Gaussian distributions. Variant A uses the expected value of the task completion times, variant B uses the worst-case scenario metric and variant C is a hybrid that implements a combination of these. The paper shows that, in simulated trials, baseline PI does not handle uncertainty well; the task-allocation success rate tends to decrease linearly as degree of uncertainty increases. Variant B demonstrates a worse performance and variant A improves the failure rate only slightly. However, in comparison, the hybrid variant C exhibits a very low failure rate, even under high uncertainty. Furthermore, it demonstrates a significantly better mean objective function value than the baseline.

1 Introduction

The ability to assign tasks well in the light of intrinsic uncertainty is very valuable for multi-agent task allocation systems. However, despite the advantages of distributed systems [1] very few robust algorithms have been developed with this architecture. To date, centralized systems have dominated research focus. This is not surprising since distributed task allocation for multi-agent systems operating in uncertain environments is a challenging problem [2]. One of the main difficulties is that the scheduling system must run independently on each agent but must generate the same schedule in each case. This can be problematic when connectivity between agents is limited or subject to change, when measurements are unreliable, or when information is imprecise or vague. These situations typically arise in Search-and-Rescue (SAR) missions where each agent may record a different location for each survivor because of inaccuracies in sensor readings, and none of the locations may be exact. Furthermore, meas-

urements of the agents' positions and velocities may also be uncertain and different for each agent. SAR missions are time-critical and demand a low probability of failure; it is vital that the assignment is reached quickly and that it represents a robust, conflict-free solution, where every survivor is rescued within the given time-frame. The main aim of this work is to attempt to address some of these challenges by creating a robust, distributed, multi-agent task allocation system with a very low failure rate.

To achieve the aim, the Performance Impact algorithm (PI) is used as the baseline for building a more robust architecture [3]. The PI algorithm has demonstrated better performance than CBBA [4] when solving deterministic model SAR problems, but lacks any mechanism for handling uncertainty. Ponda [5] has developed a robust version of CBBA using stochastic metrics such as the expected value metric [6] and worst-case scenario metric. The approach used here to extend PI is similar, but it takes the technique a step further by using a hybrid combination of expected value and worst-case scenario metric to improve robustness. The hybrid algorithm consistently demonstrates a very low failure rate and a low number of unallocated tasks in model SAR problems. Furthermore, it has a significantly better mean objective function value when compared to the baseline PI algorithm, and uses far fewer samples than Ponda's model [5].

2 Related Work

There is an extensive body of work related to multi-agent task planning, task allocation, and scheduling with many solutions proposed. These include the Contract Net method [7], Markov Random Fields (MRFs) [8], auction-based methods [9], and Distributed Constraint Optimization methods (DCOPs) [10]. In addition, solution methods can be sub-divided into optimization and heuristic types, online and offline types, and centralized and distributed communication architectures. A good review of the different approaches is presented in [11].

Time-critical, multi-agent, task allocation problems are NP-hard [12] and are thus difficult to solve using optimization approaches such as linear programming (LP), mixed integer linear programming (MILP), MRFs, and DCOPS. A MILP solution has been attempted, but the problem is not time-constrained and only eight agents and six targets are tested [13]. Pujol-Gonzalez applies an MRF-based solution to UAV online routing using the max-sum algorithm [8], but the problem is also not time-constrained and empirical tests restrict the number of UAVs to ten surveying a limited area of 100 km^2. In general, when the number of tasks and agents increases sufficiently, the optimization approach becomes intractable because of the exponential number of constraints in the model [14].

Heuristic-based methods provide an alternative as scalability is not such a problem. Popular heuristic methods include Tabu-search [15], genetic algorithms [16], and auction-based techniques [17]. In general, heuristic systems are less complex and demonstrate relatively fast execution times, although the trade-off is that they often provide sub-optimal solutions.

Auction-based heuristic algorithms, a subset of market-based approaches [18] have been widely used for solving these problems [19]. The method is easily adapted to a decentralized architecture, although this can increase complexity and communication overheads [20]. However, auction-based approaches have many benefits including high efficiency, good scalability [21], and robustness when implemented within a decentralized paradigm.

Two particular combinatorial auction-based algorithms lend themselves to the solution of the problems of interest in this paper - CBBA [4] and the PI algorithm [3]. Both algorithms use combinatorial auctions, where bundles of tasks are formed. These combinatorial methods have exhibited superior performance to single-item auctions and have generated good results when compared to optimal centralized approaches [22]. It has been shown empirically that the baseline PI algorithm performs better than the baseline CBBA algorithm [3], [23], with PI demonstrating a much better success rate with different numbers of tasks and agents, and different network topologies. However, the papers mentioned do not examine PI's handling of uncertainty.

2.1 Incorporating Uncertainty into Task Allocation Algorithms

To account for uncertainty, many researchers use Monte Carlo sampling techniques to allow the approximation of complex distributions. Undurti and How formulate the problem as a Constrained Markov Decision Process (C-MDP) [24]. Their method allows risk, defined as the probability of violating constraints, to be kept below a threshold value whilst optimizing the reward. Simulation results showed that the algorithm performed well, but the experiments were limited to only two agents. An online MDP method is used in [25], but results are inferior to basic reactive approaches and testing is based on a much simpler problem.

Maheswaran et al. enable users to encode their intuition as guidance for the system [26]. This approach simplifies a scheduling problem by decomposing it into simpler problems that can be solved in a centralized fashion. The work of Ramchurn et al. follows a similar approach, where human decisions are encoded as additional constraints for the optimization [27]. However, in work presented here attention is restricted to solutions that do not involve human intervention.

Lui and Shell postulate an alternative method that assesses the robustness of any given solution to uncertainty given a measure of it [28]. They propose the Interval Hungarian Algorithm that provides a tolerance-to-uncertainty metric for a given allocation. In particular, they compute a set of inputs that yield the same output schedule, providing a reliable method for assessing the tolerance of the allocation to uncertainties.

Creation of a solution that can hedge against uncertainty is an alternative technique to those already listed. Ponda implements this by adding probabilistic models of uncertain variables, Monte Carlo sampling, and stochastic metrics (such as the expected-value and worst-case scenario) to baseline CBBA to improve its robustness [5]. Simulation results showed improved performance over the baseline, achieving results similar to centralized approaches. However, the experiments involved only six agents and 10,000 samples were required. In addition, beyond about twelve tasks the robust algo-

rithms began to fail. This suggests that further research in this area is needed to address the problems.

3 Methodology

3.1 Problem Definition

The problem of interest is documented fully in [3], [23], [29] and [30]. It is the optimal, conflict-free assignment of a set of n heterogeneous agents $\mathbf{V} = [v_1, \ldots, v_n]^T$ to an ordered sequence of heterogeneous tasks from an m-sized set $\mathbf{T} = [t_1, \ldots, t_m]^T$. Each task has a fixed location and a maximum (latest) start time g, i.e., the problem is time-critical. Each task requires only one agent to service it, and each agent can complete only one task at a time, although it can complete other tasks afterwards, if there is time. The objective function is the minimization of mean individual task cost over all tasks rather than mean completion time for each agent, as the former takes into account the number of tasks that benefit from the time saving. The constraints are that the number of tasks assigned to a particular agent must be less than or equal to the total number of tasks, all tasks must be assigned to an agent, each ordered sequence of allocations is a subset of the whole set of tasks, tasks cannot be assigned to multiple agents, and an agent must complete a task before its latest start time.

3.2 The PI Algorithm

The PI algorithm is a distributed, multi-agent task allocation system that runs simultaneously on each agent. As in CBBA, the tasks are grouped into bundles that are continuously updated as the auction proceeds. In CBBA, the agents bid on the bundles rather than individual tasks and the bundles are formed by logically grouping similar tasks. In contrast, the PI algorithm uses a novel concept called performance impact to score and organize the task bundles. These are incrementally built and updated by systematically swapping tasks between agents, and then measuring the benefit over all tasks using special metrics. The removal performance impact (RPI) measures the benefits of removing a task from a bundle and the inclusion performance impact (IPI) measures the benefits of adding a task. Full details of the metrics and the PI algorithm are presented in [3], [23, [29] and [30]. The details are not reproduced here because of space limitations. In addition, for the purposes of this paper, the reader only needs to know that the RPI and the IPI are calculated using the time costs $c_{i,k}$ associated with each agent i and task k. In addition, creation of a robust scheduler in this way means that the methodology can be applied to other task allocation algorithms; it is not limited to the PI algorithm.

3.3 The Robust PI Variants

Each robust PI variant creates robust time cost values $r_{i,k}$ by sampling uncertain variables from a Gaussian distribution N times. A Gaussian distribution is selected because

physical quantities that are the sum of many independent processes (for example measurement errors) often have distributions that are nearly normal.

In variant A, the expected values of the actual time costs are taken as the robust time costs, where the expected value of an uncertain variable ζ is calculated as follows:

$$\mathrm{E}(\zeta) = \sum_{s=1}^{N} \zeta_s \left\| p_s \right\|. \tag{1}$$

In (1), p_s is the probability of the sample value ζ_s, and $\|p_s\|$ is the normalized value of this given by:

$$\left\| p_s \right\| = \frac{p_s}{\sum_{l=1}^{N} p_l}. \tag{2}$$

Thus, in variant A, the robust time costs are given by:

$$r_{i,k} = \mathrm{E}(c_{i,k}) = \sum_{s=1}^{N} c_{i,k} \left\| p_s \right\|. \tag{3}$$

In (3), the probability p_s of sample s is taken as the combined probability of the component uncertain variables. Variant B makes use of the worst-case scenario metric to calculate the robust time costs so that:

$$r_{i,k} = \max_{s=1}^{N} (c_{i,k}). \tag{4}$$

Variant C uses a hybrid technique that places a buffer value ψ on the difference between the expected time cost and the maximum (latest) start time of the task g_k. If

$$g_k - \mathrm{E}(c_{i,k}) < \psi \tag{5}$$

is true then the maximum time cost (4) is used for $r_{i,k}$; the deadline for the task is tight so it pays to be pessimistic. In other words, the algorithm is simply more cautious about accounting for uncertainty in its allocation. However, if (5) is false then the deadline is more flexible and the expected time cost can be used for $r_{i,k}$ as in (3).

It is important to note that each variant uses the same objective function as the baseline, but substitutes the robust time costs $r_{i,k}$ for the measured ones $c_{i,k}$. Apart from sampling and calculating the robust time costs, the procedural details for the robust PI algorithms follow the same pattern as the baseline.

4 Experimental Design

4.1 Scenario

The algorithms are tested on a scenario based on the rescue aspect of a SAR mission. The agents are UAVs carrying food and helicopters carrying medicine, and their mission is to rescue disaster survivors by delivering the supplies to them. Each survivor requires either food or medicine and their delivery constitutes the completion of a task. The start locations of the UAVs and helicopters are known in advance, as are the 3-dimensional locations and requirements of the survivors. The world x and y coordinates range from -2500m to 2500m and the z coordinates range from 0m to 1000m. The helicopters travel at 30m/s and the UAVs at 50m/s. The mission time limit is set at 5000s, the earliest start time for each task is 0s, and the latest start time is a random fraction of 5000s that cannot be less than 1500s. The task durations are fixed at 300s and 350s for medicine and food delivery respectively. Forty tasks and five vehicles are tested in each case.

4.2 Uncertainty Models

Three levels of uncertainty (low, medium and high) are considered, which vary according to prescribed errors in the key uncertain variables - task location, vehicle velocity and task duration. These variables are modelled as Gaussian distributions centered on a known mean with standard deviation equal to the estimated error. Monte Carlo sampling is used to allocate a value from the distribution to each uncertain variable and this is carried out separately for each vehicle. Relatively large errors are modelled for the task location as information relies upon intelligence from mixed external sources. UAVs generally use airspeed indicators to measure their velocity [31], but these can demonstrate instrument errors of up to about 7m/s [32]. Task durations are the most uncertain parameters since many sources contribute to them. For this reason, relatively large errors are modelled, but the uncertain values are not allowed to fall below their real values by more than 50s as it is assumed that there is a minimum time for each. For the low uncertainty case, the errors are 50m for task location, 5% for vehicle velocity and 10% for task duration. For the medium case they are 100m, 10% and 25% respectively, and for the high case they are 200m, 20% and 50% respectively.

4.3 Parameter Setting and Metrics

Preliminary trials with values between 5s and 40s confirmed that the best value to use for ψ in variant C is about 20s, although the algorithm does not appear to be very sensitive to the parameter. The size N of the samples is maintained at 100 throughout all the experiments, which are conducted using a randomly generated network topology where half of all possible connection pairs are set as communicable. This represents a realistic structure as it is not fully connected and is not as simplistic as a row or circular structure.

Each algorithm is run 100 times to obtain a percentage failure rate. Success is measured by selecting random real values for the uncertain parameters from the known probability distributions and calculating the actual number of tasks allocated to vehicles using the robust solution. If any tasks are unassigned then the run is counted as a failure; a run is only successful when all tasks are allocated. The total number of unassigned tasks is also recorded for each algorithm and uncertainty case, as are the mean objective function values.

5 Results

Figure 1 shows the total number of unassigned tasks across all 100 runs and Figure 2 shows the percentage failures for each algorithm and each uncertainty case.

Fig. 1. Unallocated tasks for each algorithm and each uncertainty case

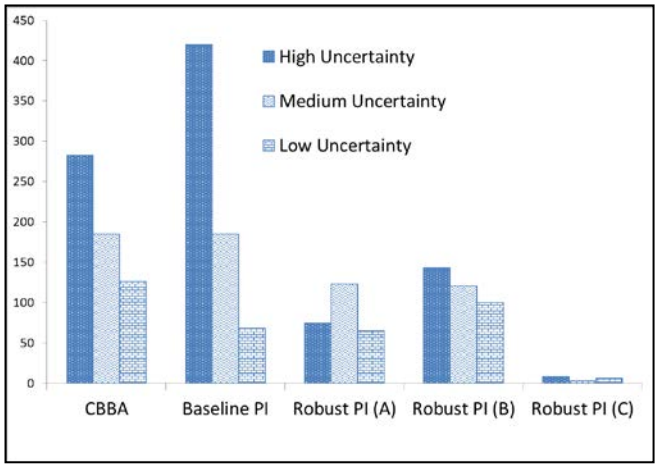
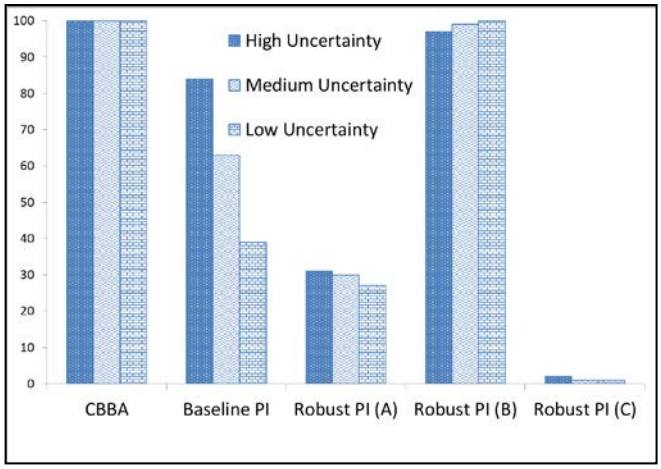


Fig. 2. Percentage failures for each algorithm and each uncertainty case



Failure to allocate tasks is a serious drawback for CBBA and baseline PI when uncertainty is modelled. For PI, the problem scales linearly with the level of uncertainty in terms of both the failure rate and the number of unassigned tasks. CBBA shows 100% failure rate for all uncertainty cases, and similar results to PI for the unassigned tasks, although it performs slightly better for high uncertainty in this respect.

Robust PI (A) improves on the baseline performance with less difference between the uncertainty cases, but the algorithm does not produce an acceptable failure rate or a low enough number of unassigned tasks. Robust PI (B)'s performance is comparable to that of CBBA in terms of the failure rate, and to Robust PI (A) in terms of the unassigned tasks. It proves 100% capable of predicting its performance but, in most cases, just predicts its own failure.

Robust PI (C) demonstrates a very low failure rate and low total number of unassigned tasks, which can be attributed to the more 'cautious' design of the algorithm. When the time margin for task completion is tight it acts pessimistically, selecting the worst-case metric to calculate the robust task costs. In addition, t-tests comparing the mean objective function for baseline PI and variant C reveal that the latter is significantly faster in all of the uncertainty cases (low: 1316s vs 1321s, medium: 1322s vs 1340s, high: 1340s vs 1378s).

There is a trade-off between a fast run-time and a robust solution; the robust variants (A and C) that use expected time-costs take about 50 times longer to run than the baseline (for example, a 1s run-time for the baseline in the high uncertainty case compares to 47s for variant C). In the baseline, the IPI-calculation dominates, taking up about 85% of execution time. Examination of the individual run times for each part of PI variants A and C shows that about 78% of run-time is devoted to the MATLAB statistical functions associated with determining the expected time costs and the probabilities of the uncertain variables. This is the key factor underlying the longer run-times. However, the IPI calculation still dominates the remaining routines, taking around 20% of total run-time, with actual time spent in the IPI routine longer than for equivalent problems solved with the baseline. Around 8 seconds are added to run-time, which may be attributed to the increase in complexity associated with computing the expected time costs and probabilities of the uncertain variables.

6 Conclusions

CBBA and baseline PI do not handle uncertainty well; a high percentage of the solutions fail to allocate all of the tasks, and the number of unallocated tasks is relatively high. Taking the expected value of the time costs reduces the failure rate and the numbers of unallocated tasks, but the method is still not reliable enough for time-critical problems. Using the worst-case scenario metric demonstrates poor performance, especially for high uncertainty. However, when a combination of the expected value and the worst-case scenario metric is used, the results are greatly improved, in terms of both a more robust solution (a 1% to 2% failure rate and low number of unallocated tasks) and a significantly smaller objective function value for all the uncertainty cases tested. In addition, only 100 samples are required to achieve this low

failure rate, which compares very favourably to the 10,000 samples used by Ponda, [5]. However, despite the small sample size, scalability in the numbers of agents and tasks is still a problem, with variant C displaying a higher run time compared to the baseline. For the model problems tested, one run still completes in a relatively short time compared to the mission length, but there is a limit to usability in terms of the numbers of tasks and agents because of the increased computation time. Future work will aim to improve the efficiency of variant C and then provide estimates of how this scales as the number of samples, tasks and agents increases.

The study of distributed robust optimization remains wide open. Most methods designed for solution of the types of problem discussed in this paper do not consider uncertainty in their solutions. Thus, the main contribution here is the successful design and implementation of a robust distributed system for solving such problems.

7 Acknowledgments

This work was supported by EPSRC (grant number EP/J011525/1) with BAE Systems as the leading industrial partner.

8 References

1. Zhang, K., Collins, E. G. Jr. & Shi, D.: Centralized and Distributed Task Allocation in Multi-Robot Team via a Stochastic Clustering Auction. ACM Trans. Autonom. Adapt. Syst, 7 (2), Article 21 (2012)
2. Horling, B. & Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. Knowl. Eng. Rev. 19, 281-316 (2004)
3. Zhao, W., Meng, Q. & Chung, P. W. H.: A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and its Application to Search and Rescue Scenario. IEEE Transactions on Cybernetics, 46 (4), 902-915 (2015)
4. Choi, H-L., Brunet, J. & How, J. P.: Consensus-based Decentralization Auctions for Robust Task Allocation. IEEE Transactions on Robotics, 25 (4), 912-926 (2009).
5. Ponda, S. S.: Robust Distributed Planning Strategies for Autonomous Multi-Agent Teams. PhD dissertation, Mass. Inst. Technol. (2012).
6. Grinstead C. M. & Snell, D. A.: Expected Value and Variance. In: Introduction to Probability, 2nd ed. Rhode Island: American Mathematical Society, pp. 936-953 (1998)
7. Smith, R. G.: The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers, C-19 (12), 1104-1113 (1998)
8. Pujol-Gonzalez, M., Cerquides, J., Meseguer, P., Rodriguez-Aguilar, J. A., & Tambe, M.: Engineering the Decentralized Coordination of UAVs with Limited Communication Range. Lecture Notes in Computer Science, 8109, pp. 199-208 (2013)
9. Lagoudakis, M., Markakis, E., Kempe, D., Keskinocak, P., KJleywegt, A., Koenig, S., Tovey, C., Meyerson, A. & Jain, S.: Auction-based Multi-Robot Routing. Robotics: Science and Systems, 5, MIT Press (2005)
10. Fave, F. M. D., Farinelli, A., Rogers, A. & Jennings, N.: A Methodology for Deploying the Max-sum Algorithm and a Case Study on Unmanned Aerial Vehicles. In: Proceedings of IAAI-2012, pp. 2275-2280 (2012)

11. Khamis, A., Hussein, A. & Elmogy A.: Multi-Robot Task Allocation: A Review of the State-of-the-art. In: A. Koubaa and J. R. Martinez-de Dios (eds.), Cooperative Robots and Sensor Networks 2015: Studies in Computational Intelligence, 604, pp. 31-51 (2015)
12. Bruno, J. L., Coffman, E. G. & Sethi, R.: Scheduling Independent Tasks to Reduce Mean Finishing Time. Communications of the ACM, 17 (7), 382–387 (1974).
13. Atay, N. & Bayazit, B.: Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem. Technical Report 2006-54, Washington University, St. Louis (2006)
14. Gerkey, B. P. & Matarić, M. J. M.: A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. Intl. J. of Robotics Research, 23 (9), 939–954 (2004)
15. Glover, F. & Marti, R. Tabu Search. In: E. Alba and R. Marti (eds.), Metaheuristic Procedures for Training Neural Networks, pp. 53-69. (2006).
16. Shima, T., Rasmussen, S. J., Sparks, A. G. & Passino, K. M.: Multiple Task Assignments for Cooperating Uninhabited Aerial Vehicles using Genetic Algorithms. Computers and Operations Research, 33 (11), 3252-3269 (2006)
17. Oliver, G. & Guerrero, J.: Auction and Swarm Multi-Robot Task Allocation Algorithms in Real Time Scenarios. In: T. Yasuda (ed.), Multi-Robot Systems, Trends and Development, pp. 437–456 (2011)
18. Dias, M. B. & Stentz, A.: Opportunistic Optimization for Market-based Multi-Robot Control. In: Proceedings of IROS-2002, pp. 2714–2720 (2002)
19. Bertsekas, D. P.: The Auction Algorithm for Assignment and Other Network Flow Problems. Technical Report, Mass. Inst. Technol., Cambridge, MA. (1989)
20. Dias, M., Zlot, R., Kalra, N. & Stentz, A.: Market-based Multirobot Coordination: A Survey and Analysis. In: Proceedings of the IEEE, 94 (7), pp. 1257–1270 (2006)
21. Coltin, B. & Veloso, M.: Mobile Robot Task Allocation in Hybrid Wireless Sensor Networks. In: Proceedings of IROS-2010, pp. 2932–2937 (2010)
22. Cramton, P., Shoham, Y. & Steinberg, R.: An Overview of Combinatorial Auction, ACM SIGecom Exchanges, 7 (1), 3-14 (2007)
23. Whitbrook, A., Meng, Q. & Chung, P. W. H.: A Novel Distributed Scheduling Algorithm for Time-critical, Multi-Agent Systems. In: Proceedings of IROS-15, pp. 6451-6458 (2015)
24. Undurti A. & How, J. P.: A Decentralized Approach to Multi-agent Planning in the Presence of Constraints and Uncertainty. In: Proceedings of ICRA-2011, pp. 2534-2539 (2011)
25. Musliner, D. J., Durfee, E. H., Wu, J., Dolgov, D. A., Goldman, R. P. & Boddy, M. S.: Coordinated Plan Management using Multiagent MDPs. In: Proceedings of SSDPSM (2006)
26. Maheswaran, R. T., Rogers, C. M., Sanchez R. & Szekely, P.: Realtime Multi-agent Planning and Scheduling in Dynamic Uncertain Domains. In Proceedings of ICAPS (2010.
27. Ramchurn, S.D., Fischer, J. E., Ikuno, Y., Wu, F., Flann, J & Waldock, A.: A Study of Human-Agent Collaboration for Multi-UAV Task Allocation in Dynamic Environments. In: Proceedings of IJCAI (2015)
28. Liu, L. & Shell, D. A.: Assessing Optimal Assignment under Uncertainty: An Interval-based Algorithm. International Journal of Robotics Research, 30 (7), 936-953 (2011)
29. Turner, J., Meng, Q. & Schaeffer, G.: Increasing Allocated Tasks with a Time Minimization Algorithm for a Search and Rescue Scenario. In Proceedings of ICRA-2015, pp. 340-3407 (2015)
30. Whitbrook, A., Meng, Q. & Chung, P. W. H.: Reliable, Distributed Scheduling and Rescheduling for Time-Critical, Multi-Agent Systems. IEEE Transactions on Automation Science and Engineering, to appear (2017)
31. Collinson, R. G. P.: Introduction to Avionic Systems, 3rd ed., London: Springer.
32. Huston, W. B.: Accuracy of Airspeed Measurements and Flight Calibration Procedures. Technical Report No. 919, NACA, Langley Memorial Aeronautical Laboratory (1948)