Reliability-Aware Multi-UAV Coverage Path Planning Using Integer Linear Programming

Mickey Li
Bristol Robotics Laboratory
University of Bristol
Bristol, UK
mickey.li@bristol.ac.uk

Arthur Richards
Bristol Robotics Laboratory
Unviersity of Bristol
Bristol, UK
arthur.richards@bristol.ac.uk

Mahesh Sooriyabandara
Bristol Research and Innovation Laboratory
Toshiba Research Europe Ltd
Bristol, UK
mahesh@toshiba-trel.com

***Abstract*—Multi-Agent Systems have the potential advantage of graceful degradation over Single-Agent systems. This is a desirable trait in applications that require area coverage with failure-prone UAVs. This paper uses methods informed by Reliability Engineering to study this formally by using a new stochastic framework which evaluates a strategy's probability of task completion. Based on this analysis, an integer linear programming formulation of the problem is then shown to provide almost optimal strategies at a fraction of the computational cost of brute force methods.**

***Index Terms*—Multi-Robot Systems, Coverage Path Planning, Reliability Analysis**

I. Introduction

Multi-agent systems are well known to provide greater reliability, resilience and fault tolerance to individual and system level failures compared to single-agent systems [1, 2]. This robustness property makes a multi-UAV system especially suited to the problem of area coverage and inspection, as UAVs are particularly prone to failure [3–5]. Whilst [6, 7] consider failure handling, they only assert eventual completion. This work takes a probabilistic planning approach to model failures to provide temporal guarantees of completion. Figure 1 motivates this problem - by what evaluation method is strategy 2 optimal? The aim of this paper is to present (1) an exact reliability evaluation framework, inspired by the field of Reliability Engineering [8], (2) a computationally feasible approximate method, for multi-drone path planning which takes into account the uncertainty in failure of individual drones, such that the *Probability of Mission Completion (PoC)* is maximised. The resultant plan is thus the Pareto-Optimal strategy with respect to reliability and efficiency.

II. Markov Reliability Analysis Framework

The state of the system of n agents $\mathbf{A}$ at a time t is $x^t = (\tau_1^t, \dots, \tau_n^t) \in \mathbb{N}^n = \mathbf{S}$ where τ_i^t is the length of time agent i has survived (i.e. the useful time worked) under the following Markov transition rule:

$$\tau_i^t = \begin{cases} \tau_i^{t-1} + 1 & \text{if agent } i \text{ survives} \\ \tau_i^{t-1} & \text{if agent } i \text{ fails or already failed} \end{cases} \tag{1}$$

This work has been funded by the Engineering and Physical Sciences Research Council (EPSRC) iCASE with Toshiba Research Europe Ltd and FARSCOPE Centre of Doctoral Training at the Bristol Robotics Laboratory

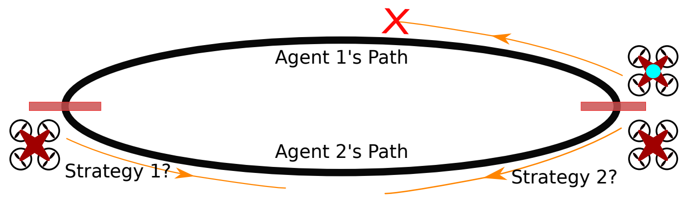

Fig. 1. Consider 2 drones covering a 1D cyclic path. Agent 1 suffers a failure. Should Agent 2 follow strategy 1 or 2 such that the greatest probability of completion is achieved? Intuitively, it must be strategy 2, but why?

A set of m spatially connected, discrete tasks in an environment is defined as $\mathbf{J} = (j_1, \dots, j_m)$. A **strategy** is then defined as a function $\pi_i(x, t) = j$, $x \in \mathbf{S}$ and $j \in \mathbf{J}$ where j is the task performed by agent i at time t given the health states of all agents x.

This framework is described with respect to **static** strategies: pre-allocated fixed paths for each agent with no dependencies on the state $\mathbf{S}$. The strategy matrix $T \in \mathbb{N}^{n \times m}$ can be defined where T_{ij} = the time at which agent i visits task j. T can be mapped onto $\pi_i(x, t)$, for any time t and agent i, by finding the task j such that $T_{ij} = \tau_i$. A task j is **completed** by agent i if agent i survives longer than the time at which the drone is scheduled to visit task j, i.e $\tau_i \geq T_{ij}$.

A state $x \in \mathbf{C} \subseteq \mathbf{S}$ is a **coverage completion state** iff:

$$F(x) = \min_{j \in [1..m]} E_j(x) = \min_{j \in [1..m]} \max_{i \in [1..n]} \tau_i - T_{ij} \geq 0 \tag{2}$$

as the **time since completion of task** j, $E_j(x) \geq 0$ for all tasks $\mathbf{J}$ and therefore every task has been visited.

Given the failure probability density $f_i(\tau) = p_i(t = \tau)$ for each agent, the **probability of completion** (PoC) for a given strategy T at a particular time t' is the sum of the probabilities of surviving until each of the completion states of t', $\mathbf{C}_T(t')$.

$$\mathbf{C}_T(t') = \{x \in \mathbf{C} | \forall \tau \in x, \tau \leq t'\} \tag{3}$$

$$PoC(T, t') = \sum_{x \in \mathbf{C}_T(t')} \prod_{\tau_i \in x} p_i(\tau_i, t') \tag{4}$$

$$p_i(\tau, t') = \begin{cases} p_i(t = \tau) = f_i(\tau), & \text{if } \tau < t' \\ p_i(t > \tau) = R_i(\tau), & \text{if } \tau \geq t' \end{cases} \tag{5}$$

Equation (5) describes the case where the agent fails before t' and where the agent survives at t'. $R_i(\tau)$ is known as **Reliability**. **Failure Rate** $\lambda_i = \frac{f_i(\tau)}{R_i(\tau)}$ is a system parameter.

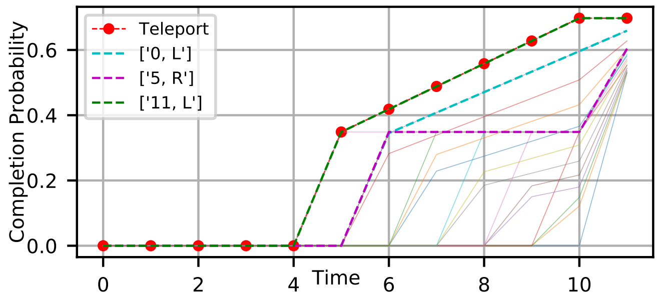

Fig. 2. The PoC of all cyclic strategies for a path of length 12, with 2 agents, $\lambda = 0.1$. Top performing strategies highlighted. Legend describes strategies (initial task, direction) w.r.t one of the agents starting at 0 travelling right.

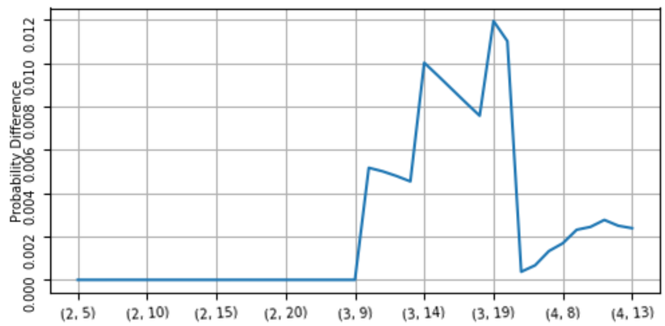

Fig. 3. The difference in PoC between optimal strategies reported by the computationally feasible brute force searches and the ILP. (#drones, #path)

Considering the state as 'work done', instead of explicit locations of agents and completion of tasks, allows the decoupling of the state analysis from the probability analysis.

Complex **non-static** strategies may be defined based on agent failures, but are difficult to formalise. In principle, the static framework can still be applied by running a unique simulation of π for every $x \in \mathbf{S}$ and labelling completion. In general this is non-trivial apart from the 'teleportation' strategy where task spatial constraints are not considered. Teleportation can be proven to be the optimal strategy.

The framework can be used to find the Pareto-Optimal strategy for a fixed time horizon in a brute force manner by evaluating every strategy for a given number of drones and path. Figure 2 shows the optimal strategies found via brute force search for the motivating problem in Figure 1. The strategy '[11,L]' corresponds to Strategy 2 and provides a greater PoC by end time 11 than Strategy 1, '[5, R]', which agrees with intuition.

III. Integer Linear Programming Approach

The brute force method, whilst exact, is only computationally feasible for small scenarios (e.g. 4 agents, 20 path takes a week). An Integer Linear Programming (ILP) formulation is presented to provide an approximation of the optimal strategy, whilst being computationally tractable.

A strategy is represented by the binary variable X where $X_{ij}(t) = 1$ iff agent i visits task j at t. Instead of maximising PoC directly, the ILP minimises the maximum log-probability of any single task j being missed $\ln p(m_j)$. $p(m_j)$ is then the probability of all agents independently failing before their scheduled visits of j.

$$\underset{X}{\text{minimize}} \quad \max_j \ln \prod_i \prod_t F(t)^{X_{ij}(t)} \tag{6}$$

$$= \max_j \sum_i \sum_t X_{ij}(t) \ln F(t) \tag{7}$$

$$\text{subject to} \quad \sum_j X_{ij}(t) = 1 \qquad \forall i, t \tag{8}$$

$$\sum_i X_{ij}(t) \leq 1 \qquad \forall j, t \tag{9}$$

$$\sum_i \sum_k X_{ij}(t) \geq 1 \qquad \forall j \tag{10}$$

$$X_{ij}(t) \leq \sum_{j' \in \mathcal{N}(j)} X_{ij'}(t-1) \qquad \forall i, j, t \tag{11}$$

$$\sum_t X_{ij}(t) \leq 1 \qquad \forall i, j \tag{12}$$

Where (8) ensures every agent can only do 1 task at each time. (9) ensures every task is visited at some time and (10) ensures every task is visited at some time by some agent. (11) constrains agent movement to neighbouring tasks. Finally (12) stops an agent from visiting a task more than once.

By minimising $\max_j p(m_j)$, the ILP makes it unlikely that any given task is missed. Intuitively, this makes completion more likely, and it can be shown that it maximises a lower bound on PoC. It is not equivalent to a direct optimisation of PoC as it does not capture task inter-dependencies. Nevertheless, experiments suggest that the ILP selects strategies whose PoC comes close to the optimal PoC, found by exhaustive evaluation.

The PoC of the ILP's chosen strategy is calculated by converting X into a static T matrix for the Markov Framework. To evaluate the accuracy, the ILP is applied to cyclical paths of various lengths and numbers of drones using the Gurobi LP solver [9]. Figure 3 shows the difference between the PoC of the strategy obtained through brute force evaluation compared to the outputted strategy of the ILP. The ILP appears to provide a very close PoC with its strategy choice, only a small deviation for longer paths. This is promising as close to exact accuracy can be obtained from a method which takes a fraction of the time (e.g. 4 agents, 20 path takes minutes).

IV. Conclusion

This paper proposes that for applications where knowing all tasks are completed are crucial, but the hardware is prone to failure, reliability may be favoured over efficient plans. Inspired by Reliability Engineering, a probabilistic evaluation framework is presented which reports on the probability of completion of a mission given a strategy. The Pareto-Optimal strategy can be found through brute force evaluation of all strategies, but this is computationally infeasible. An Integer Linear Programming approach is shown to provide almost optimal results for a fraction of the computation time. Future work will include exploring non-linear optimisation approaches as well as considering more complex drone failure models such as localisation and communication failures.

REFERENCES

[1] G. Weiss, *Multiagent systems, Second Edition*. MIT Press, 2013. ISBN 9780262018890. [Online]. Available: https://mitpress.mit.edu/books/multiagent-systems-second-edition

[2] H. Hamann, *Swarm Robotics: A Formal Approach*. Cham: Springer International Publishing, 2018. ISBN 978-3-319-74526-8. [Online]. Available: http://link.springer.com/10.1007/978-3-319-74528-2

[3] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 9 2012. doi: 10.1177/0278364912455954. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364912455954

[4] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor Unmanned Aerial Vehicles," in *2012 Proceedings of IEEE Southeastcon*. IEEE, 3 2012. doi: 10.1109/SECon.2012.6196930. ISBN 978-1-4673-1375-9 pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6196930/

[5] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A Survey on Aerial Swarm Robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 8 2018. doi: 10.1109/TRO.2018.2857475. [Online]. Available: https://ieeexplore.ieee.org/document/8424838/

[6] N. Hazon and G. A. Kaminka, "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1102–1114, 12 2008. doi: 10.1016/J.ROBOT.2008.01.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008000122

[7] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 10 2010. doi: 10.1109/IROS.2010.5651321. ISBN 978-1-4244-6674-0 pp. 5577–5582. [Online]. Available: http://ieeexplore.ieee.org/document/5651321/

[8] P. D. T. O'Connor and A. Kleyner, *Practical reliability engineering*. Wiley, 2012. ISBN 9780470979815

[9] Gurobi Optimization LLC, "Gurobi Optimizer Reference Manual," 2020. [Online]. Available: http://www.gurobi.com