Optimal Deployment of Multiple Passenger Robots using Sequential Stochastic Assignment

Chris (Yu Hsuan) Lee, Graeme Best, Geoffrey A. Hollinger

***Abstract*— We present an informed passenger robot deployment strategy for marsupial robots. A marsupial robot system consists of a carrier robot (e.g., a ground vehicle), which is highly capable and has a long mission duration, and at least one passenger robot (e.g., a short-duration aerial vehicle) transported by the carrier. Passenger robots can have specialized and complementary capabilities that are tailored to specific tasks during the mission. However, an effective passenger robot deployment strategy is necessary in order to optimize performance of the carrier-passenger system. We propose a deployment algorithm that reasons over uncertainty by exploiting information about the prior probability distribution of features of interest in the environment. Our algorithm is formulated as a solution to a sequential stochastic assignment problem (SSAP). The key feature of the algorithm is a recurrence relationship that defines a set of observation thresholds that are used to decide when to deploy passenger robots. Our algorithm computes the optimal policy in $O(N^2)$ time, where N is the number of deployment decision points. Our results show that our deployment algorithm outperforms other competing algorithms, such as the classic secretary approach and baseline partitioning methods.**

I. Introduction

Exploration of increasingly complex environments demands more flexible robotic capabilities. Developments in heterogeneous multi-robot systems have yielded carrier-passenger robot systems called "marsupial robots", in which highly-capable carrier robots carry and deploy one or more low-capability passenger robots. These marsupial robots can tailor their complementary capabilities to the challenges of exploring complex environments [1]. During exploration, these environments can contain multiple features of interest that the carrier robot may want to observe but are prohibitive or difficult to reach [2], [3]. In marine environments, a large ship may carry and deploy multiple heterogeneous robots to increase the rate of information gathering of features such as seafloor mines, adversarial vessels, and biological hotspots [4], [5], [6]. In the case of urban environments, a team of ground and aerial robots may seek to explore features like ledges, vertical shafts, stairways, or other urban features which ground robots cannot easily reach [7]. During these missions, the carrier robot would ideally deploy the passenger robot at a location which can maximize exploration coverage or information gain and make use of the passenger robot's complementary capabilities (e.g., the ability to fly in 3D terrain).

*This work was supported in part by Office of Naval Research Grant N00014-17-1-2581.

*The authors are with the Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, Corvallis OR, USA. {leeyuh,bestg,geoff.hollinger}@oregonstate.edu

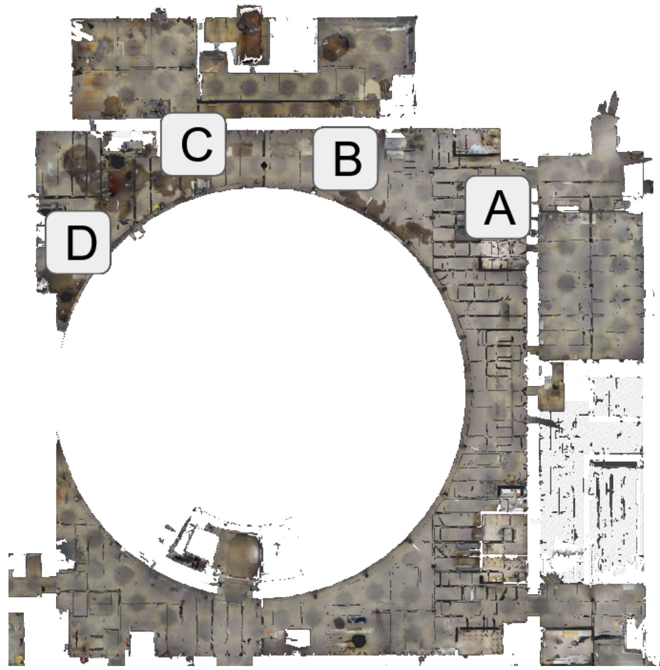

Fig. 1. RGB point cloud map of an urban environment from the DARPA Subterranean Challenge [8] with example deployment locations. Each decision point is revealed sequentially to the robot. Point A has high exploration potential since it leads to another room, but not as much as point C, which leads to multiple rooms and a stairwell only accessible to aerial passenger robots. However, points B and D are poor deployment locations since they lack nearby features to be explored by the passenger robots.

In order to optimize exploration efficiency, a marsupial robot system must decide when and where to deploy its passenger robots. In Fig. 1, an example of an urban environment is illustrated with sequential decision locations with different exploration potential. If a carrier robot deploys too early, it risks missing out on the potentially more valuable later decision points. Each of these sequential decision locations may have different reward values which are only revealed when they are observed along the robot's path. Due to the unknown nature of an unexplored environment, the carrier robot is required to make online deployment decisions while reasoning over the possible discovered deployment value at each potential deployment location. Furthermore, the value and number of these features are not known in advance. Multi-robot systems require coordination beyond naïve partitioning of deployment between the passenger robots to ensure that the maximum expected number of features are captured [9]. Prior works have explored marsupial robot coordination and planning [3], [10] but, to our knowledge, do not explicitly solve for passenger robot deployment or deploys in a naïve manner [11]. To address these challenges, we develop an online passenger deployment strategy that reasons over the predicted future reward of deploying at each decision point.

We formulate the deployment strategy as a sequential stochastic assignment problem (SSAP) [12] where a set of passenger robots are assigned to deployment locations. We assume the probability distribution of the features in the environment is known a priori. The deployment algorithm exploits this probabilistic prior information of the distribution of features throughout the environment. The key component of the algorithm is a recurrence relationship that defines a set of observation thresholds. These thresholds are used to decide when to deploy passenger robots by comparing to the current observation of deployment reward. Furthermore, the thresholds only depend on the number of decision points remaining in the mission and the prior distribution of features. The algorithm is guaranteed to find the optimal online deployment policy under the assumption of a known prior distribution with independent observations. Also, the algorithm is efficient in that it has $O(N^2)$ runtime, where N is the number of decision points.

Our contributions in this paper are:

1) A multi-robot passenger deployment problem formulated as a sequential stochastic assignment problem
2) An informed passenger deployment strategy for this problem, which builds on [12], that maximizes the expected sum of the deployment rewards

The solution to the problem formulation is guaranteed to find the optimal online policy in quadratic time. We show empirically that our algorithm is competitive with an offline oracle solution that has full access to the rewards in advance and greatly outperforms comparable deployment algorithms, including a partitioned solution to the classic secretary problem [13], and naïve baseline methods. On average, the sequential assignment deployment algorithm performed within 96% of the oracle.

II. Related Work

Heterogeneous robotic teams are a subset of multi-robot systems that are composed of different types of robots with varying, often complementary, capabilities. There have been works which evaluate distributed coordination of these heterogeneous-robot systems [14], [15]. In most heterogeneous robot systems, the motion of the robots are loosely-coupled without physical constraints between robots. In marsupial robots, tightly-coupled constraints are physically imposed on the deployments of the passenger robots by the carrier robots. These marsupial robots are a special type of heterogeneous robots, comprising of a carrier "mother" robot and potentially multiple passenger "child" robots [1]. The complementary nature of marsupial robotics enables these combinations of robots to traverse a wider type of terrain and approach a larger set of tasks during a mission.

The increased complexity of marsupial robots requires new planners which are able to handle tasks like coordination, deployment, retrieval, and manipulation. A temporal symbolic planner was created by Wurm et al. [10] in order to organize and coordinate between multiple marsupial robotic systems. The authors are able to utilize the framework to handle exploratory navigation actions, as well as plan for debris clearing actions. However, the temporal planner does not explicitly decide an optimal deployment location and relies on frontier exploration algorithms to generate deployment targets. Mathew et al. [14] addresses the task scheduling and path planning for package delivery using marsupial robots. They consider urban environments where deployment locations and potential routes are known beforehand. In [16], the authors consider marsupial robot path planning and passenger robot deployment in order to minimize return signals by an adversary and to maximize the amount of information gained from a target of interest. The authors simplify the passenger deployment criteria by having the passenger deploy immediately and do not consider multiple passenger deployments. Hansen et al. [17] employs an iterative planning approach to deploy sensors for the highest expected information gain. Their deployment system consists of an autonomous surface vehicle which deploys passive drifters following water currents in a marine environment.

The deployment problem is closely related to optimal stopping theory [13], which considers problems that require deciding online the right time to carry out a particular action. A key example of an optimal stopping problem is the classic secretary problem, but does not leverage prior information regarding future rewards [13]. However, the Cayley-Moser problem [18] reasons over predicted future rewards by considering the prior probability distribution. This has been applied to robotics problems by Lindhé and Johansson [19], who consider the problem of when to communicate by utilizing multi-path fading communication model to make predictions about future stopping decisions. Our method is a generalization of the Cayley-Moser algorithm, adapted for the context of the deployment problem. Additionally, we formulate a policy that considers all future deployments, rather than just the single next decision point.

Optimal stopping variants have been generalized to multiple decision points. Das et al. [4] applies multi-choice optimal stopping theory for AUVs to collect multiple plankton samples which minimizes the cumulative regret of the samples. They explore two applicable algorithms, the multi-choice hiring algorithm [20] and the submodular secretary algorithm [21]. Both of these algorithms seeks to select the best observations and do not consider information such as the distribution of the observations. A generalization of the Cayley-Moser algorithm, sequential stochastic assignment problems (SSAP) [12], seeks to find an optimal policy to maximize the expected sum of rewards from multiple assignments of agents to tasks. SSAP has been applied to other fields [22] but, to our knowledge, has not been utilized in robotics. A closely related body of work focuses on multi-robot task assignment problems [23]; however, these problems generally assume that task values are known a priori and are therefore not directly applicable to online problems, such as ours.

III. Problem Formulation

We consider a marsupial robot system that must continually make online decisions regarding when to deploy its passenger robots. At each deployment location, the robot must consider if the reward gained from deployment is expected to be more favorable than continuing onwards and deploying at a later location. We formulate the multi-robot deployment decision as a sequential stochastic assignment problem (SSAP). Multiple deployment decisions are made based on sequentially revealed random variables. We formalize this problem as follows.

A carrier robot moves through an environment and makes decisions to deploy or not deploy a passenger robot. There are assumed to be a total of N decision points, where N is known to the robot. The carrier robot houses R passenger robots of equal capability. At each decision point, the carrier robot may choose to deploy one passenger robot.

We consider an environment that contains a set of point features, which may represent points of interest that are ideal for additional exploration by passenger robots, but are ill-suited for the carrier robot. We assume these features are distributed as a Poisson point process with a known rate Λ.

At decision point j, the carrier robot detects features within a circular sensing area centered at the current location, with sensing radius ρ. The number of features, x_j, that are detected within a sensing area is drawn from a Poisson distribution, which has a probability mass function

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{for } x \in \{0, 1, ...\}. \tag{1}$$

The rate of this Poisson distribution λ is proportional to Λ as $\lambda = \Lambda A$, where A is the ratio between the sensing area and the environment area. Along the sequence of decision points, the observations of the number of features in the sensing area are denoted $(X_1, ..., X_N)$. We assume that the sensing areas do not overlap, and therefore the observations are independent.

At stage $j \in \{1, ..., N\}$, the carrier robot reaches a decision point, and the outcomes of all random variables $(X_1, ..., X_j)$, denoted $(x_1, ..., x_j)$, are known to the robot. Along the path, the robot must make an irreversible decision at each deploy location to deploy or continue on to deploy later. If the carrier robot decides to deploy, it assigns one passenger robot to the deployment location and returns a reward of x_j. If the carrier robot decides to continue, no reward is claimed at this stage.

This process continues for the N stages. All passenger robots must be deployed by stage N, with the constraint of one deployment per stage. We define the set of stages where passenger robots were deployed as $(d_1, ..., d_R)$. The goal of the carrier robot is to maximize the expected sum of the reward returned from the deployment locations; i.e., find the optimal deployment sequence:

$$(d_1, ..., d_R)^* = \underset{(d_1, ..., d_R)}{\operatorname{argmax}} \mathrm{E}\Big[\sum_{r=1}^{R} x_{d_r}\Big]. \tag{2}$$

Here, x_{d_r} denotes the reward from passenger robot r.

IV. Online Passenger Deployment Algorithm

We present the online passenger deployment algorithm, which finds the optimal deployment policy in quadratic time. The algorithm is computed via dynamic programming, using a technique that stems from sequential stochastic assignment [12]. We begin by considering the general case of R passenger robots to deploy, then formulate a specialized efficient solution for a single passenger robot. Finally, we provide an analysis of the optimality and runtime complexity of the algorithm.

A. Multi-Robot Deployment

Our deployment algorithm precomputes a set of thresholds, which are dependent on the distribution of the observations and the number of remaining stages. At stage j, the current observed value x_j is compared to the threshold values and informs the carrier robot whether or not to deploy now.

For convenience, we define a sequence of values $(p_1, ..., p_N)$, such that $p_i = 1$ for $N - R < i \leq N$, which represent the R passenger robots, and $p_i = 0$ for $1 \leq i \leq N - R$. From this definition, the problem can then equivalently be thought of as assigning these p_i values to each x_j.

Specifically, the optimal policy for the passenger robot assignment is to assign p_i to the observed deployment value x_j, if the observation value x_j falls into an ith non-overlapping interval comprising the real line [12]. These non-overlapping intervals are separated by a set of thresholds, denoted as $a_{i,n}$, where n is the number of remaining stages. Each threshold $a_{i,n}$ is computed recursively and depends only on the pdf $f(x)$, as well as number of stages N. For stage $j = N - n + 1$, there are a set of thresholds, such that

$$-\infty = a_{0,n} \leq a_{1,n} \leq a_{2,n} \leq ... \leq a_{n,n} = \infty. \tag{3}$$

At stage j, the optimal choice is to use p_i if the realization x_j of the random variable X_j is contained in the interval $(a_{i-1,n}, a_{i,n}]$.

Threshold $a_{i,n+1}$ is defined as the expected value, if there are n stages remaining, of the quantity to which the ith smallest p is assigned [12]. This formulates the recurrence relationship

$$\begin{aligned}
a_{i,n+1} = {} & \Pr(x_n < a_{i-1,n}) a_{i-1,n} \\
& + \Pr(a_{i-1,n} < x_n < a_{i,n}) \\
& \quad \times \mathrm{E}(x_n | a_{i-1,n} < x_n < a_{i,n}) \\
& + \Pr(x_n > a_{i,n}) a_{i,n} \qquad (4) \\
= {} & a_{i-1,n} \int_{-\infty}^{a_{i-1,n}} f(x) dx + \int_{a_{i-1,n}}^{a_{i,n}} x f(x) dx \\
& + a_{i,n} \int_{a_{i,n}}^{\infty} f(x) dx, \qquad (5)
\end{aligned}$$

where $-\infty \cdot 0 = 0$ and $\infty \cdot 0 = 0$. In both (4) and (5), the second term is for the case where x_n lies within the ith interval, and therefore p_i receives the value of x_n. The first term is for the case where x_n is below the ith interval, meaning that x_n is assigned to a $p_k < p_i$, and thus p_i is

5	-	-	-	-	∞
4	-	-	-	∞	$a_{4,5}$
3	-	-	∞	$a_{3,4}$	$a_{3,5}$
2	-	∞	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
1	∞	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
i / n	1	2	3	4	5

Fig. 2. The method for the recurrence threshold calculation is shown and the relationship between the terms in (4). $a_{i,n}$ is the threshold for the ith non-overlapping interval when there are n stages left to go. For example, the term $a_{2,3}$ is calculated from the $a_{i-1,n}$ term, which is $a_{1,2}$, and the $a_{i,n}$ term, which is $a_{2,2} = \infty$.

assigned in a later stage with an expected value of $a_{i-1,n}$. Similarly, the third term is for the case where x_n is above the interval and p_i has an expected future assignment of $a_{i,n}$.

In our case, the observation values $(X_1, X_2, ..., X_j)$ are Poisson distributed, with the pmf shown in (1). Substituting (1) into (4), yields

$$a_{i,n+1} = a_{i-1,n} \sum_{x=0}^{\lfloor a_{i-1,n} \rfloor} \frac{\lambda^x e^{-\lambda}}{x!} + \sum_{x=\lceil a_{i-1,n} \rceil}^{\lfloor a_{i,n} \rfloor} x \frac{\lambda^x e^{-\lambda}}{x!} + a_{i,n} \left[1 - \sum_{x=0}^{\lfloor a_{i,n} \rfloor} \frac{\lambda^x e^{-\lambda}}{x!} \right]. \quad (6)$$

The above recurrence relationship can be computed in quadratic time by iterating over n and i. This process is illustrated in Fig. 2, where the cells in the table can be computed from left to right by reusing the previous values, as indicated by the arrows.

B. Single-Robot Deployment

A single passenger robot presents a single decision for the carrier robot to decide to deploy at a location. In the single passenger robot case, the sequential stochastic assignment algorithm only considers a single $p_N = 1$. Now, the only thresholds relevant for deployment are the $a_{n-1,n}$ elements. These elements are shown as the cells one below the diagonal in Fig. 2. With only one set of thresholds to calculate, the $a_{n-1,n}$ terms are identical to the thresholds used in the Cayley-Moser optimal stopping problem [18]. This has the benefit of only requiring linear, rather than quadratic, computation time in N.

C. Analysis

The dynamic programming proceeds by iteratively solving optimal subproblems for $a_{i,n+1}$ using the recurrence relationship in (6), and thus computes the optimal set of thresholds $a_{i,n}, \forall i, n$. The full proof of the results that these subproblems are optimal, and that these thresholds lead to an optimal online assignment policy, is presented in [12]. The proof proceeds by induction, and relies on Hardy's Theorem [24], which states that the optimal assignment between two sets with a sum-product objective is to pair the smallest values in each set, then the next smallest, and continued until the largest values are paired.

As illustrated in Fig. 2, there are $O(N^2)$ subproblems to be computed, where N is the total number of stages. The integral (5) is computed once per subproblem, thus the computation time is $O(N^2F)$, where F is the time to compute (5). For our problem, (5) simplifies to (6), which can be computed in time $O(A)$, where A is the maximum value of any threshold $a_{i,n}, i < n$. As discussed in Sec. IV-B, for a single-robot deployment, the number of subproblems is reduced to $O(N)$, thus yields a computation time of $O(NF)$.

V. Experiments

In order to evaluate the effectiveness of our algorithm for passenger robot deployment, we conducted simulation tests comparing our algorithm with various other deployment strategies. On average, our algorithm performed within 96% to the oracle.

A. Experimental Setup

A 2D simulated world containing randomly distributed features of interests was created, as seen in Fig. 3. The features of interests were modeled as points and would only be detected within the observation range of the carrier robot. Certain deployment algorithms described below divides the carrier robot path into R equal partitions in the case of multi-robot deployment. The algorithms treat each partition independently from each other. Different deployment strategies were employed, aiming to maximize the amount of features detected:

- *Sequential Assignment*: Performs as described in the multi-robot deployment approach and considers the entire path as a singular deployment, without partitions and deployment constraints.
- *Oracle*: Selects the top R deployment location across all the partitions, with perfect knowledge of the world in advance.
- *Partition Oracle*: Selects the best deployment location within each partition, with perfect knowledge of the world in advance.
- *Cayley-Moser*: Performs as described in the single robot deployment approach in each partition.
- *Classic Secretary*: An optimal stopping variant that only observes for the first N/Re of decision points and then selects the next value that is higher than the max value observed in the observation phase [13]. The algorithm runs within each partition.
- *Random*: Selects a decision point randomly within each partition.
- *First*: Selects the first decision point within each partition.
- *Last*: Selects the last decision point within each partition.

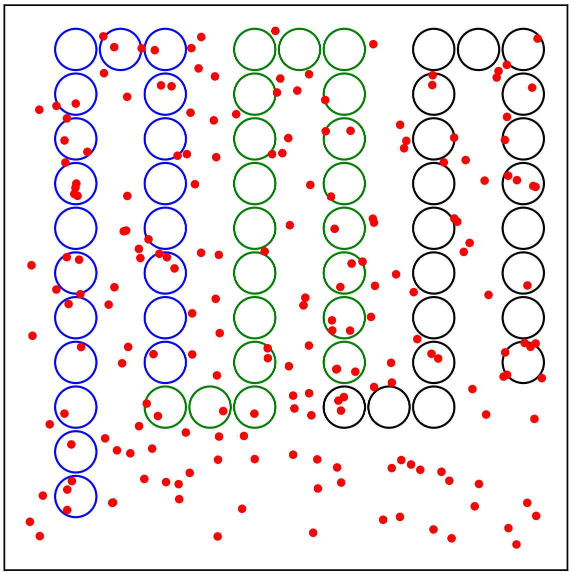

Fig. 3. Lawnmower path with three deployment partitions. 200 sparse features (red dots) shown here for visual fidelity. Circle outlines denote the robot's observation area at each deployment decision point.

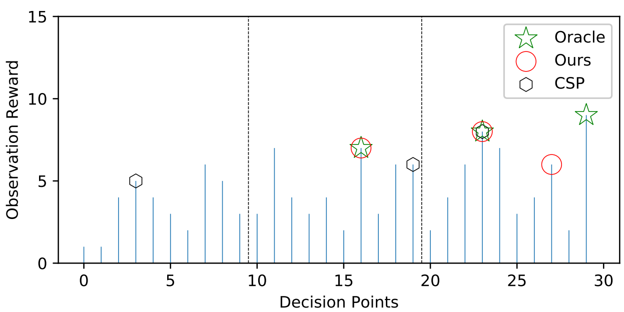

Fig. 4. Illustrative example of the deployments for three passenger robots to be deployed over 30 decision points. Our method (reward: 21) performs similar to the *Oracle* (reward: 24) that has full knowledge of the observation sequence in advance. The *Classic Secretary* (reward: 19) algorithm performed relatively poorly, in part due to being constrained by having one deployment in each of the three partitions (dotted lines).

B. Results

An example of a trial of the algorithm selection process is shown in Fig. 4. The *Oracle* algorithm optimally selects the top $R = 3$ rewards. Our algorithm, *Sequential Assignment*, decided to deploy and select the reward in an optimal fashion in the first two decision cases. The algorithm does not select early deployment locations because it expects to encounter higher values in the future and the observed values were not higher than the required thresholds. Lastly, the algorithm's last decision was not optimal. As the end stage is approaching, the thresholds begin to lower and relax since there are less stages for higher future expected values. The last few thresholds approaches the expectation of a single observation and prompts the algorithm's selection before the last stage. The *Classic Secretary* algorithm does not reason over possible future observations and locally selects a decision point in each partition. In the second partition, the algorithm passes over a valuable decision location within the first N/Re decision points and is forced to accept the last decision point in that partition. Lastly, the algorithm selects a high deployment value location in the third partition but not the best possible deployment location.

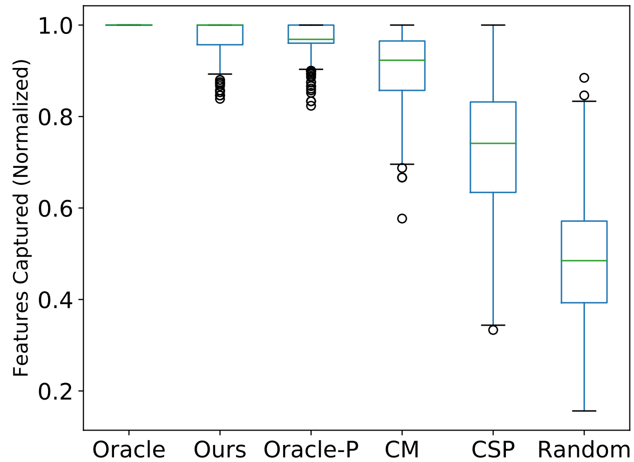

Fig. 5. Algorithm comparisons with three passenger robots over 250 test iterations. *Partition Oracle* is denoted as Oracle-P. CM is *Cayley-Moser* and CSP is *Classic Secretary*. The *First* and *Last* algorithms performed similarly to the *Random* algorithm and are excluded for clarity.

The results of the *Sequential Assignment* algorithm through repeated tests are extremely positive. Trials with three passenger robot deployments and $n = 60$ stages are shown in Fig. 5. During each trial, the algorithms' results are compared to the max possible number of features captured accomplished by the *Oracle*. On average, the *Sequential Assignment* algorithm performed within 96% of the *Oracle*. Also, the *Partition Oracle* is not as perfect as the *Oracle* but performs similarly when compared to the *Sequential Assignment* algorithm. The *Partition Oracle* captures each of the local maximum, which provides excellent results, but does not consider a global purview and is unable to capture the global maximum. The partitioned *Cayley-Moser* algorithm has a lower performance than the *Sequential Assignment* algorithm since the calculated thresholds are only local to each partition and is constrained to one deployment location per partition. The *Sequential Assignment* algorithm is able to account for the expected future values over the total number of stages, whereas the partitioned *Cayley-Moser* locally calculated the thresholds only for the number of stages in a partition.

The *Sequential Assignment* algorithm generalizes to R passenger robot deployments. Multiple test scenarios were conducted with a different number of passenger robots in order to examine the behavior of the algorithms as a function of the R deployment decisions. The average utility, as a percentage of the *Oracle*, is shown for the algorithms in Fig. 6. The performance of *Sequential Assignment* handily outperforms other algorithms. *Cayley Moser* is mathematically equivalent to *Sequential Assignment* in the case where $R = 1$ and yields identical results, as discussed in the single-robot deployment section.

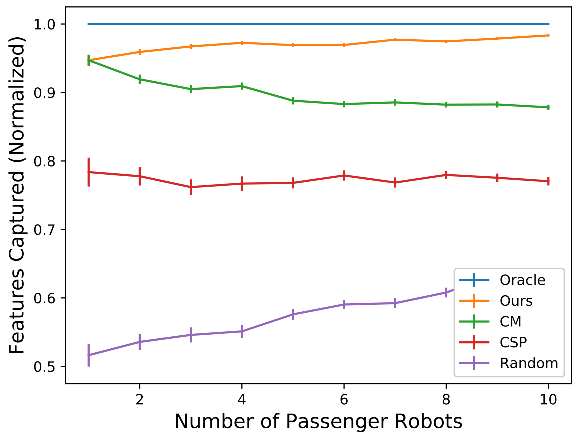

Fig. 6. Algorithm comparisons with R passenger robots and 150 test iterations. The *First* and *Last* algorithms performed similarly to *Random* and are excluded for clarity. SEM error bars are displayed.

VI. Future Work

We show a formulation of passenger robot deployment for marsupial robots and demonstrate the feasibility of the sequential assignment optimal policy to passenger robot deployments. However, there is still a need to address various aspects and assumptions of our deployment algorithm in order to increase the real-world robustness of the algorithm. In order to address one of the algorithm's main assumption, it would be interesting to study the case where the prior feature distribution is not known and to learn the distribution online [25]. Additionally, to more closely represent real-world robot exploration, the deployment algorithm should be simulated in a more realistic environment that incorporates obstacles such as real-world maps and data. Furthermore, practical deployment of passenger robots may impose constraints between subsequent deployments which warrants further research. An example of a constraint is preventing back-to-back deployments between consecutive decision points for robot collision safety. Also, it is interesting to consider a dependent relationship between consecutive observations. Overlapping observations more closely represent the case where the robot continuously observes new information along the path as it is travelling. Lastly, the sequential stochastic assignment problem handles the case where multiple passenger robots are to be deployed but does not extend to the case with multiple carrier robots. We aim to extend the deployment problem formulation to multiple carrier robots as well.

References

[1] R. R. Murphy, M. Ausmus, M. Bugajska, T. Ellis, T. Johnson, N. Kelley, J. Kiefer, and L. Pollock, "Marsupial-like mobile robot societies," *Proc. Int. Conf. on Autonomous Agents*, pp. 364–365, 1999.

[2] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[3] J. Moore, K. C. Wolfe, M. S. Johannes, K. D. Katyal, M. P. Para, R. J. Murphy, J. Hatch, C. J. Taylor, R. J. Bamberger, and E. Tunstel, "Nested marsupial robotic system for search and sampling in increasingly constrained environments," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 002 279–002 286.

[4] J. Das, F. Py, J. B. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *Int. J. Robotics Research*, vol. 34, no. 12, pp. 1435–1452, 2015.

[5] F. Marques, A. Lourenço, R. Mendonça, E. Pinto, P. Rodrigues, P. Santana, and J. Barata, "A critical survey on marsupial robotic teams for environmental monitoring of water bodies," in *Proc. IEEE OCEANS*, 2015.

[6] M. Kalaitzakis, B. Cain, N. Vitzilaios, I. Rekleitis, and J. Moulton, "A marsupial robotic system for surveying and inspection of freshwater ecosystems," *J. Field Robotics*, 2020, DOI: 10.1002/rob.21957.

[7] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanskỳ, D. Heřt, M. Petrlík, T. Báča, V. Spurnỳ *et al.*, "DARPA subterranean challenge: Multi-robotic exploration of underground environments," in *Proc. Int. Conf. on Modelling and Simulation for Autonomous Systems*. Springer, 2019, pp. 274–290.

[8] DARPA, "Explore alpha course in 3D." [Online]. Available: https://my.matterport.com/show/?m=dP4NytELbJB

[9] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 1846–1852.

[10] K. M. Wurm, C. Dornhege, B. Nebel, W. Burgard, and C. Stachniss, "Coordinating heterogeneous teams of robots using temporal symbolic planning," *Autonomous Robots*, vol. 34, no. 4, pp. 277–294, 2013.

[11] M. S. Couceiro, D. Portugal, R. P. Rocha, and N. M. Ferreira, "Marsupial teams of robots: Deployment of miniature robots for swarm exploration under communication constraints," *Robotica*, vol. 32, no. 7, pp. 1017–1038, 2014.

[12] C. Derman, G. J. Lieberman, and S. M. Ross, "A sequential stochastic assignment problem," *Management Science*, vol. 18, no. 7, pp. 349–355, 1972.

[13] T. S. Ferguson, *Optimal Stopping and Applications*. UCLA, 2008, ch. Finite Horizon Problems, p. 2.1–2.15.

[14] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Trans. on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[15] G. Best, W. Martens, and R. Fitch, "Path planning with spatiotemporal optimal stopping for stochastic mission monitoring," *IEEE Trans. on Robotics*, vol. 33, no. 3, pp. 629–646, 2017.

[16] J. C. Las Fargeas, P. T. Kabamba, and A. R. Girard, "Path planning for information acquisition and evasion using marsupial vehicles," in *Proc. American Control Conf. (ACC)*. IEEE, 2015, pp. 3734–3739.

[17] J. Hansen, S. Manjanna, A. Q. Li, I. Rekleitis, and G. Dudek, "Autonomous marine sampling enhanced by strategically deployed drifters in marine flow fields," in *Proc. MTS/IEEE OCEANS*, 2018.

[18] L. Moser, "On a problem of Cayley," *Scripta Math*, vol. 22, pp. 289–292, 1956.

[19] M. Lindhé and K. H. Johansson, "Exploiting multipath fading with a mobile robot," *Int. J. Robotics Research*, vol. 32, no. 12, pp. 1363–1380, 2013.

[20] Y. Girdhar and G. Dudek, "Optimal online data sampling or how to hire the best secretaries," in *Proc. Canadian Conf. on Computer and Robot Vision*. IEEE, 2009, pp. 292–298.

[21] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2010, pp. 39–52.

[22] V. Shestak, E. K. Chong, A. A. Maciejewski, and H. J. Siegel, "Probabilistic resource allocation in heterogeneous distributed systems with random failures," *J. of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1186–1194, 2012.

[23] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *Int. J. Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[24] G. Hardy, J. Littlewood, and G. Polya, *Inequalities*. Cambridge University Press, 1934.

[25] S. C. Albright, "A Bayesian approach to a generalized house selling problem," *Management Science*, vol. 24, no. 4, pp. 432–440, 1977.