Sequential Stochastic Multi-Task Assignment for Multi-Robot Deployment Planning

Colin Mitchell, Graeme Best, Geoffrey Hollinger

***Abstract*— Real-time sequential decision making under uncertainty is a challenging task for autonomous robots. Such problems are even more challenging when making decisions involving heterogeneous teams of robots completing multiple tasks. Deploying autonomous taxi cabs and utilizing drones for package delivery represent relevant examples of these types of problems. In this paper, we present an effective solution to a multi-robot multi-task sequential stochastic assignment problem using a simulation-based optimization algorithm (MARP). Our algorithm employs a novel approach that uses Monte Carlo simulation to seek the deployment with the highest probability of being optimal. To demonstrate MARP's performance and robustness, we performed more than 2,000 numerical experiments in two different problem domains, evaluating MARP's performance against three different comparison algorithms. These numerical studies show that MARP significantly outperforms the comparison methods, achieving results within 5% of the maximum possible reward.**

I. INTRODUCTION

Designing robots capable of making complex real-time decisions related to performing multiple tasks, before all pertinent information has been revealed, poses a substantial challenge that limits the extent of viable autonomous operation. Yet many real-world applications require addressing these kinds of problems. Exploration and search and rescue under dangerous operating conditions (e.g. underwater, underground, in space) are characterized by considerable opportunities for preventing human injury or death associated with these essential activities. Another relevant application involves dispatching autonomous vehicles. Prominent examples associated with emerging technology include Amazon's proposed use of autonomous drones for package delivery and fleets of autonomous taxi cabs.

In this paper, we consider a marsupial robot system (a carrier robot with deployable heterogeneous passenger robots) navigating through an unknown environment. At each decision point along the sequence, the carrier must determine the appropriate deployment action when the task rewards/requirements surrounding all future decisions are unknown. Its objective is to maximize the aggregate reward obtained from completing all tasks that manifest as the carrier robot navigates the environment. However, the current, irreversible, action reduces the carrier's available options for subsequent deployments and may decrease the possible total reward. Deploying a highly efficient resource too early risks not being able to satisfy requirements or realize a larger reward later. Deploying the same resource too late risks realizing a lesser reward.

Previously published work relating to this problem can be categorized into three distinct areas: (1) task completion in environments with uncertainty [1], [2], [3], (2) homogeneous teams of robots completing a single task at each decision point in a stochastic decision sequence [4], [5], and (3) heterogeneous teams used for concurrent or collaborative task completion [6], [7]. However, to the best of our knowledge, none of this prior work addresses sequentially deploying robots from a heterogeneous team to perform multiple tasks at each decision point before all task requirements are revealed.

To address these challenges, we developed an online algorithm to maximize the probability of an optimal deployment sequence. Our simulation-based optimization approach utilizes Monte Carlo simulation to facilitate solving a complex stochastic optimization problem characterized by multiple sources of uncertainty (multiple task rewards/requirements at each decision point). In multiple experiments, using both a domain inspired by the DARPA Subterranean Challenge and a domain using actual New York city taxi fare data, our algorithm significantly outperformed two comparison methods and generally yielded solutions whose aggregate reward was within 95% of the maximum reward possible when all tasks are revealed in advance (i.e. the maximum possible reward under perfect information).

The contributions of this paper include: (1) introducing a new class of the sequential stochastic assignment problem involving multiple tasks at each decision point and multi-task robots with defined efficiencies for each task, (2) a solution to this new problem which utilizes Monte Carlo simulation to address uncertainty while maximizing the probability of an optimal aggregate reward, and (3) accommodation of fuzzy constraints by reformulating task requirements as decision variable terms in the objective function.

II. RELATED WORK

To better understand the literature in this area, it is useful to organize the related works into three categories. The first category involves exploratory tasks using robot teams. A subset of this research involves exploratory tasks with no decision making component. Marques et al. offer a critical survey of literature relating to marsupial robotic teams monitoring bodies of water [1]. They specifically note the difficulty of completing tasks in unknown environments.

*This work is funded in part by NSF award IIS-2103817.

*C. Mitchell and G. Hollinger are with the Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, Corvallis OR, USA. {mitchcol,geoff.hollinger}@oregonstate.edu

*G. Best is with the School of Mechanical and Mechatronic Engineering, University of Technology Sydney, NSW, Australia. graeme.best@uts.edu.au

Extensive research effort has been invested in working with autonomous systems to sample data, explore, survey, and inspect environments [6], [8], [9]. Moore and Wolfe explore semi-autonomous search and sampling using a heterogeneous team of robots [10]. Hansen et al. and Kalaitzakis et al. have explored a similar search and sampling task in a marine environment [2], [11].

Another subset of this research involves exploratory tasks with non-complex decision making. Using a team of heterogeneous autonomous robots to perform application-specific tasks has been another focus of many published works. Petris and Khattak present a method using a team of marsupial walking and flying robots to explore unknown areas [3]. This combination of robots allows for general widespread exploration of rough terrain areas as well as focused exploration using aerial robots when legged locomotion is insufficient. This is similar to the work presented by Couceiro et al., in which swarm robots are autonomously deployed to support previously deployed agents [12].

The second category focuses on complex decision making involving deployments under uncertainty. Derman presents the Sequential Stochastic Assignment Problem (SSAP) which establishes the basis of most of this category of work [4]. This problem is particularly challenging because deployment decisions must be made sequentially under uncertainty and incomplete information. The algorithm presented provides an optimal solution to the deployment problem described. Lee et al. extend the SSAP algorithm with their Online Passenger Deployment (OPD) algorithm which deploys passenger robots in unknown environments where the number of decision points is greater than or equal to the number of resources [5]. Other extensions of Derman's problem add additional sources of uncertainty to more closely resemble real world scenarios. For example, Khatibi and Jacobson tackle the SSAP problem with the added complication that deployed resources may not successfully complete the task [13]. Other related works consider the uncertain arrival of resources [14], random task deadlines [15], the opportunity to postpone deployment decisions [16], and uncertainty in job value distributions [17].

Papers in the third category address complex decisions more closely related to set covering problems than sequential deployment problems. Liu et al. present work showing the importance of team oriented coverage planning while considering cost constraints [6]. Similarly, Wu et al. explore the impact of heterogeneity when trying to complete complex tasks and avoid associated risks [18]. Their results show that specific combinations of heterogeneous teams outperform homogeneous teams in their disaster relief scenario. Wurm et al. present a method that combines a symbolic planning system with path planning to facilitate coordination of heterogeneous teams [19].

Our work is positioned at the intersection of all three categories of related work, specifically the online sequential deployment of multi-task capable robots for applications involving multiple stochastic tasks. The first of the research categories discussed above involves teams of robots operating in unknown environments, but without online decision making. The research in category two extends this related work with sequential online decision making, but only for single task robots. In contrast, the work outlined in category three addresses stochastic tasks and multi-task capable robots, but all robots in the teams are deployed simultaneously.

III. PROBLEM FORMULATION

We consider a marsupial robot system that must make online decisions regarding when to deploy its heterogeneous passenger robots. At each possible deployment location, the robot must consider whether the reward gained from deployment is expected to be more favorable than proceeding and deploying at a later location. If so, the carrier robot must determine which of its passenger robots to deploy. We formulate this Multi-Robot Multi-Task Sequential Stochastic Deployment Problem (MRMT-SSDP) as a Sequential Stochastic Assignment Problem (SSAP). Multiple deployment decisions are made based on sequentially revealed random tasks and their associated rewards.

We assume that an environment contains a sequence of decision points (DPs) which represent sites where tasks performed by passenger robots may be warranted. At any given DP, there are M tasks that a passenger robot may perform (e.g. sampling, camera feed, communication relay, etc.). The carrier robot houses R passenger robots with efficiencies $\mathbf{e}_i = (e_{i,1}, ..., e_{i,M}), \forall i \in \{1, ..., R\}$ relative to the tasks that need to be performed. At each DP, the carrier robot must make an irreversible decision to deploy or not to deploy a passenger robot. If a decision is made to deploy, the carrier must also decide which of its passengers to deploy. There are assumed to be a total of $N \geq R$ decision points, where N is known to the carrier robot. Along the sequence of decision points, the independent observations of the rewards associated with the tasks are denoted $(\mathbf{X}_1, ..., \mathbf{X}_N)$, where $\mathbf{X}_j = (X_{j,1}, ..., X_{j,M})$ and $X_{j,k}$, a random variable, represents the reward associated with completing the k^{th} task at DP j. Furthermore, the algorithm relies on knowing the prior distributions of random variables $X_{j,k}$, denoted as $f_{j,k}(x)$. By the end of the decision sequence, all robots must be deployed.

At stage $j \in \{1, ..., N\}$, the carrier robot reaches a decision point, and the outcomes of all random variables $(X_{j,1}, ..., X_{j,M})$, denoted $\mathbf{x}_j = (x_{j,1}, ..., x_{j,M})$, are known to the robot. If the carrier robot decides to deploy, it assigns one passenger robot, i, to the deployment location and realizes reward $\pi(\mathbf{x}_j, \mathbf{e}_i)$, where $\pi(\mathbf{x}, \mathbf{e})$ calculates the total reward based on the individual task rewards and the assigned robot's efficiencies. 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 at most one deployment per stage.

We define the set of deployments as $D = \{d_1, ..., d_R\}$, where $d_r = (i, j)$ represents a robot-task pair assigning robot i to decision point j. The goal of the carrier robot

is to maximize the sum of the rewards associated with the deployment pairs; i.e., find the optimal deployment sequence:

$$D^* = \underset{D}{\operatorname{argmax}} \sum_{(i,j)\in D} \pi(\mathbf{x}_j, \mathbf{e}_i). \quad (1)$$

IV. ALGORITHM TO MAXIMIZE AGGREGATE REWARD PROBABILITY

To address all of the points listed above, we introduce the Maximize Aggregate Reward Probability (MARP) algorithm which employs a simulation based optimization approach to maximize the probability of an optimal deployment at every decision point along the decision sequence. This approach is shown empirically to achieve results comparable to an Oracle (with respect to aggregate reward). Additionally, it allows us to fully incorporate the task probability distributions as well as consider the deployment constraints on each resource. In the following sections, we begin by defining the proposed algorithm in a general setting, then provide a domain specific implementation for each of our example problems.

A. General Algorithm

Our algorithm requires four inputs to make a deployment decision at a decision point: (1) revealed task values for the current decision point, (2) probability distributions to model each task, (3) the remaining number of decision points, and (4) the number of simulated trials to use in constructing the optimal deployment probability distribution. For simplification, we assume the carrier knows exactly how many decision points there are in the decision sequence. Using these inputs, MARP executes the following steps at each decision point:

1) Using simulated future DP task values and the currently revealed task values, generate a matrix whose elements are the reward, $\pi(\mathbf{x_j}, \mathbf{e_i})$, for all combinations of task values and resources.
2) Determine the optimal deployment sequence[1] for trial t, $D'_t = \{d_j, d_{j+1}, ..., d_N\}$, through the final decision point, using the reward matrix generated in step 1.
3) Repeat steps 1 and 2 for T trials per decision point.
4) Using the current DP deployment from all $D' \in \{D'_1, ..., D'_T\}$, construct the optimal deployment probability mass function, $m(d)$, for the current decision point.
5) Deploy resource with the maximum probability of being the optimal deployment, $d^*_j = \operatorname{argmax}_d m(d)$.

MARP repeats these steps, in sequence, until one of two termination conditions occurs: (1) the algorithm runs out of resources to deploy, or (2) the carrier is at the end of the decision sequence and no more decisions must be made. Fig. 1 illustrates the use of simulated trials in determining a deployment with the highest probability of being optimal. To simplify the implementation, we add $N-R$ dummy resources with zero efficiencies. We therefore deploy a resource at all N decision points. However, $N-R$ of these deployments are dummy resources, effectively representing decisions to not deploy.

[1]We used an implementation of the Hungarian algorithm, $O(n^3)$ complexity, but any appropriate combinatorial optimization algorithm could be used.

B. Domain Specific Implementations

1) Known Distributions: We first consider a general environment with randomly generated task values using multiple combinations of task distributions, for example a mine, cave, or urban environment (similar to those used in the 2021 DARPA Subterranean Challenge[2]). We use known distributions to generate our task values for every decision point in the decision sequence; we use the same distributions for MARP's simulations at each decision point.

For this experiment, we assume any available resource may be deployed at any decision point. We evaluate the reward obtained by any resource at a decision point as the dot product of the efficiency vector and the task vector. In other words for robot i at decision point j:

$$\pi(\mathbf{x}_j, \mathbf{e}_i) = \mathbf{x}_j \cdot \mathbf{e}_i = \sum_{t=1}^{M} x_{j,t} e_{i,t} \quad (2)$$

2) N.Y. Taxi Data: Our next domain demonstrates the flexibility of MARP by applying it to dispatch available autonomous taxi cabs. For these experiments we used New York City yellow and pink taxi cab fare data[3][4] obtained from Kaggle (an open source data repository). Conceptually, passengers request a cab and specify three requirements as well as their pick-up and drop-off locations. Specific data used from the Kaggle repository includes:

- Passenger count: Number of passengers for the fare, treated as the minimum vehicle passenger capacity requirement.
- Payment type: Credit or cash payment, treated as a requirement to accept cash or handle either.
- Rate code: Used to identify an airport destination or pick-up, treated as a requirement for larger luggage capacity.
- Distance: Distance of the trip in miles, used to calculate the fare profit

The depot in this example assumes the role of the carrier robot and the taxis (assumed to be autonomous) assume the passenger role. We use all four of the fare attributes outlined above for scoring an assignment by first evaluating whether or not the taxi meets all three requirements, then use the vehicle's profit per mile value and the distance task value to calculate the reward (in this case profit) obtained by the taxi-task pairing.

We treat the three requirements as fuzzy constraints; the objective function is modified to assess a penalty if one or more of the constraints is unmet. This penalty is equal to the negative of the reward value associated with the fare. However, when calculating the aggregate reward after

[2]https://www.subtchallenge.com/

[3]https://www.kaggle.com/datasets/neilclack/nyc-taxi-trip-data-google-public-data

[4]https://www.kaggle.com/datasets/pavandas/city-taxi-trip-pricing-and-distances

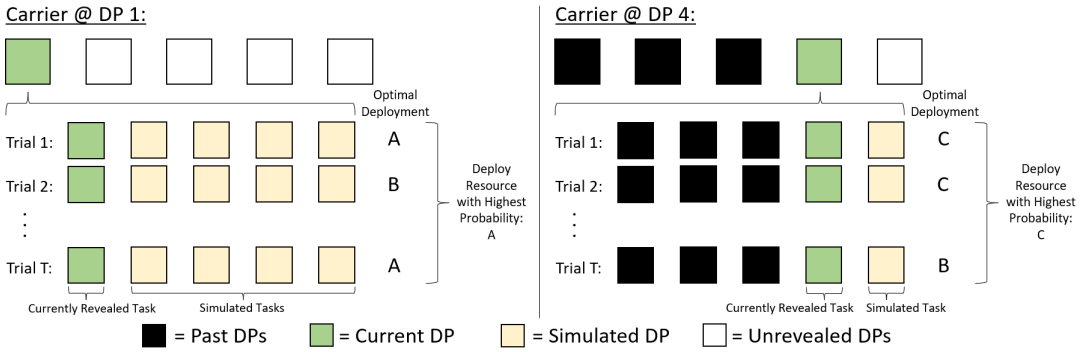


Fig. 1: Illustration of the MARP algorithm: (left) showing the algorithm at decision point (DP) 1 with the first DP revealed and subsequent DPs simulated. (right) shows the currently revealed DP (4), a single subsequent simulated DP (5), and past DPs (1-3) no longer relevant for the current decision. Reward maximizing deployments depicted represent optimal deployments for the trials; the actual deployment for the current DP will be the deployment that occurs with the highest probability.

all assignments are made, a zero reward value is realized for any deployment with one or more unmet requirements. Because fare distance is the only task value not representing a requirement, the reward calculation simplifies to:

$$\pi(\mathbf{x}_j, \mathbf{e}_i) = x_{j,k_d} \cdot e_{i,k_d} \tag{3}$$

where k_d is the index of the distance task such that x_{j,k_d} is the distance for ride request j, and e_{i,k_d} is the profit per mile for taxi i.

The profit per mile efficiency depends on the type of vehicle used. Statistically, the need for a van in New York City is far less than for a traditional sedan. Due to the low van usage frequency, we assume higher van usage cost (e.g. lower gas mileage, parking, purchase/financing costs) as compared to sedans. This was observed in the profit per mile data obtained and is reflected in the taxi pools used for our testing.

For this domain, we generated probability distributions using the data obtained from Kaggle to model the fare distance and requirements. Our experiments utilized two different methods for generating distributions:

- Using the entire data set - use all of the data to fit the distributions and use them to generate random fares for testing.
- Using training and testing subsets of the data set - select a random subset of the data to fit the distributions, then use remaining data to represent fares for testing.

Finally, we fit distributions to model the task values in two ways: 1) utilizing the full N.Y.C. fare data set (*complete*) and 2) using separate randomly sampled subsets for fitting task distributions and testing algorithm performance (*subset*). For *complete* experiments, test tasks are randomly generated using the resulting distributions.

V. RESULTS AND ANALYSIS

A. Comparison methods

To evaluate MARP's effectiveness, we ran[5] two types of experiments, as discussed in section IV-B, one with randomly generated tasks using a variety of known distributions, and the other with New York City taxi data. Furthermore, we evaluate MARP's performance against multiple baselines produced using two comparison methods described below; due to the novelty of our problem, we cannot directly turn to the literature for existing comparison methods. Additionally, we used an earlier heuristic method (see Future-Current Heuristic below) we developed for the MRMT-SSDP.

The ideal evaluation requires comparing an algorithm's aggregate reward performance, $\Pi^\eta = \sum_{(i,j)\in D} \pi(\mathbf{x}_j, \mathbf{e}_i)$ calculated for method η, with respect to a truly optimal policy (i.e. when all task values are known in advance) representing the exact upper bound in our solution space determined using a completely revealed decision sequence. We refer to this solution as the Oracle. Similar to MARP, the Oracle implementation uses the Hungarian algorithm to find the optimal deployment sequence to maximize the total reward.

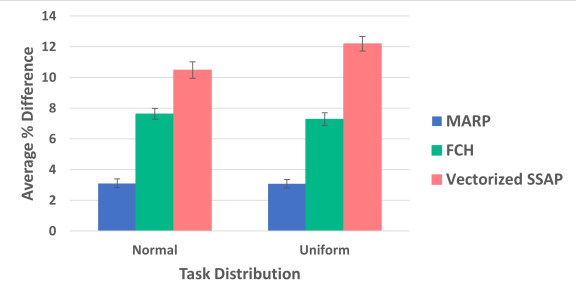
We calculate the percentage difference from Oracle, for MARP and the two comparison methods, to provide directly comparable performance metrics:

$$\%\,\text{diff} = \frac{\Pi^O - \Pi^\eta}{\Pi^O}$$

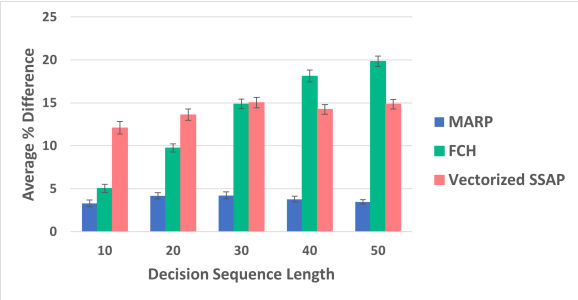
where Π^O represents the maximum possible aggregate reward produced by Oracle. This allows us to establish MARP's absolute performance compared to the optimal, as well as its performance with respect to the two comparison methods:

- **Vectorized SSAP:** In Lee's [5] original implementation, the carrier robot holds a team of passenger robots, each of which has a different efficiency for the same task. This allows resources to be ranked by their efficiencies and deployment decisions are made using a threshold table of deployment reward expected values in a manner similar to Derman [4]. However, this approach cannot be directly used to solve the MRMT-SSDP problem because multiple tasks and heterogeneous passengers preclude any meaningful ranking of the resources.
  Generating the OPD threshold table requires a single task value probability distribution. To adapt the OPD algorithm to the MRMT-SSDP problem, we must com-

[5]All experiments performed on a standard desktop computer with Windows 10 using AMD Ryzen 5 3600X 6-core processor (3.8GHz) and 32 GB ram.



(a) Comparisons using Normal and Uniform task distributions with means and coefficients of variation held constant for all tasks.



(b) Comparisons using Normal tasks distributions held constant for different decision sequence lengths.

Fig. 2: Comparison of percent difference from optimal aggregate reward using MARP (proposed algorithm), FCH, and Vectorized SSAP. In both cases, MARP significantly outperforms both of the comparison methods. (a) MARP performs consistently for both Normal and Uniform distributions; MARP's percentage differences from optimal aggregate reward are more than 50% less than those produced by the second best method (FCH). (b) MARP's performance is consistent for all decisions lengths tested; FCH and Vectorized SSAP show declining performance for decision sequences of increasing lengths. Error bars for both plots are one SEM.

bine multiple task distributions into the single task value distribution required by ODP. Creating such a convolution for table generation could be difficult. While it doesn't strictly apply, we use a heuristic based on the Central Limit Theorem to combine the distributions into one; we assume the resulting distribution is normally distributed with mean $\mu = \sum_k \mu_k$ and variance $\sigma^2 = \sum_k \sigma_k^2$, where μ_k and σ_k represent the mean and variance of task distribution k. Similarly, it is necessary to combine the passenger robots efficiencies, which are stated in reward units per unit of task value; we calculate the aggregate efficiency as the sum of the individual task efficiencies.

Because Vectorized SSAP makes no provision for evaluating the constraints we are unable to use it as a comparison method in the N.Y. taxi data experiments; the algorithm does not consider individual resources when generating the threshold table.

- **Future-Current Heuristic:** This heuristic algorithm is based on the concept of Derman's SSAP algorithm [4] discussed above. Rewards are calculated using Eqn. 2 and "future" rewards are calculated using expected values. At each decision point:
  1) Calculate current reward $\rho_{C,i}$ for available passengers using revealed task values
  2) Calculate expected reward $\rho_{E,i}$ for available passengers using task distributions
  3) Deploy the passenger with maximum $\rho_{C,i} - \rho_{E,i}$

B. Known Distribution Results

We tested our algorithm on a set of resources, each with different efficiencies, by changing the following parameters:

- Number of simulations per decision point
- Combinations of task probability distributions
- Decision sequence length

In all of our experiments to date, MARP outperformed both Vectorized SSAP and FCH when using various combinations of Normal, Uniform, Exponential, and Poisson distributions.

We found that in this domain, the number of trials per decision point had a relatively minor effect on the aggregate reward performance; the range of average percent differences is only 0.02% when varying the number of trials per DP from 100 to 10,000. (see Fig. 3).

Varying distributions did not appear to significantly affect aggregate reward performance relative to oracle (see Fig. 2a). Moreover, when testing with uniform distributions to model all tasks, MARP still performed well with respect to the oracle's average optimal score.

Similarly, altering the resource efficiencies, while affecting the aggregate reward values, did not affect MARP aggregate reward with respect to oracle. Additionally, we observe that MARP's performance is relatively unaffected by the number of decision points, whereas FCH demonstrated pronounced deterioration as the number of decision points increases; Vectorized SSAP also shows deterioration, but less pronounced than was observed for FCH (see Fig. 2b).

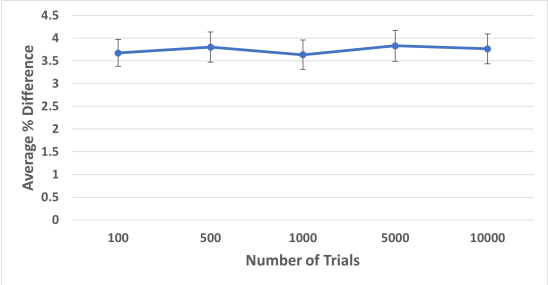


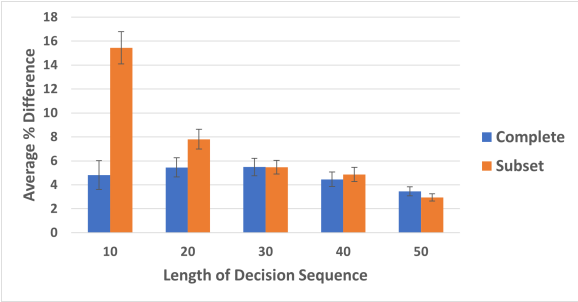
Fig. 3: Effect of number of trials per decision point on average percent difference from the optimal aggregate reward. Independent identically distributed (IID) tasks normally distributed with $\mu = 100$ and $\sigma = 60$. Percentage difference is stable even with smaller numbers of trials. Error bars are one SEM.

C. N.Y. Taxi Data Results

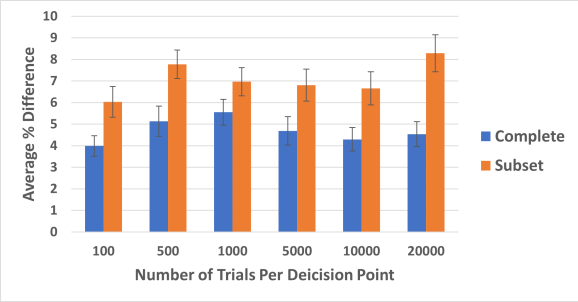
For this problem domain we ran multiple tests altering the following parameters:

- Number of trials per decision point
- Task probability distributions (entire dataset vs. a randomly sampled subset)
- Decision sequence length

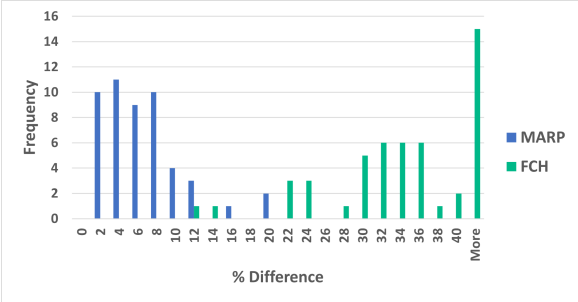
MARP performed very well in the New York City taxi assignment domain even when using relatively few simulated



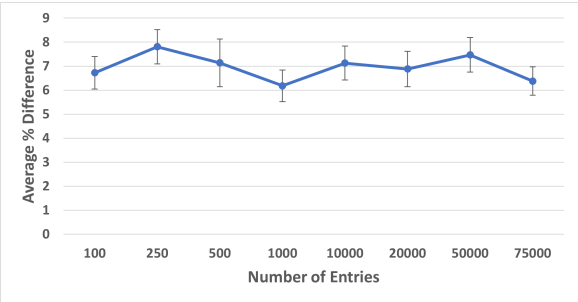
(a) Effect of decision sequence length on MARP's aggregate reward performance.



(b) Effect of trial count on MARP's aggregate reward performance.



(c) Comparison of MARP and FCH for a single test with 1000 trials per decision point.



(d) Effect of distribution fitting sample size on MARP's aggregate reward performance.

Fig. 4: N.Y.C. Taxi testing results using two different distribution fitting methods. *Complete* experiments fit task distributions using the entire data set. *Subset* experiments fit task distributions used randomly sampled observations. (a) MARP generally performs well for all decision sequence lengths. Additionally, performance advantages observed in *complete* experiments effectively disappear for sequences of 30, or more, DPs. (b) In this domain, MARP performs well very small trial count. The aggregate reward distributions in (c) clearly demonstrate MARP's significantly performance gains over FCH. (d) shows that distribution fitting sample size has relatively little impact on aggregate reward performance. (a), (b), and (c) used 20,000 observations for distribution fitting. Error bars for plots (a), (b), and (d) are one SEM.

trials (see Fig. 4b) and small training data sets. This is a direct result of the nature of the probability distributions in the problem domain. For example, the probability of a passenger requirement exceeding eight people is very small (1% or less). Failing to realize such an outlier in a small number of simulated trials does not affect the results significantly. Similarly, while failing to include such unlikely outliers in a small training data set, the resulting truncated distribution still effectively represents the population and does not materially impact MARP's performance.

We found that varying the number of observations between 100 and 75,000 yielded generally acceptable results with relatively small performance penalties associated with sample sizes that are too small or too large (see Fig. 4d). There are obvious runtime benefits associated with using smaller training data sets and fewer trials. The ideal training data set size and simulated trial count will be domain specific, depending on the nature of the tasks and requirements. Additionally, with small decision sequences there is a clear advantage to using larger sample sizes. However, with longer decision sequences, there appears to be no compelling evidence that large samples outperform smaller ones (see Fig. 4a).

To better understand MARP's performance, we looked at the distribution of reward percent differences for MARP and FCH (see Fig. 4c). MARP's percent differences are fairly tightly clustered around a much smaller mean than observed for FCH. This significantly smaller average and reduced variability suggests that MARP will produce consistently effective solutions.

VI. CONCLUSION

In this work, we formulated the Multi-Robot-Multi-Task Stochastic Sequential Deployment Problem in which a carrier robot must decide which, if any, of its passenger robots to deploy at each decision point along the decision sequence. We introduced our simulation-based optimization method, MARP, which maximizes the probability of making an optimal deployment at each decision point. This in turn seeks to maximize the probability that we will obtain the optimal aggregate reward. We then presented the results from two sets of experiments, for different problem domains, which showed that MARP scored close to Oracle and performs better than previously developed algorithms. Execution benchmarks show that MARP consistently averages 200 ms per decision point, a very reasonable computation time for likely application areas (e.g. exploration, search and rescue, data collection).

ACKNOWLEDGEMENT

We thank Mr. Joshua Fan for assistance with testing MARP in both the known distribution and N.Y. Taxi domains.

REFERENCES

[1] 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 *OCEANS MTS/IEEE Genova*, 2015, pp. 1–6.

[2] 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 *OCEANS MTS/IEEE Charleston*, 2018, pp. 1–7.

[3] P. De Petris, S. Khattak, M. Dharmadhikari, G. Waibel, H. Nguyen, M. Montenegro, N. Khedekar, K. Alexis, and M. Hutter, "Marsupial walking-and-flying robotic deployment for collaborative exploration of unknown environments," *arXiv preprint arXiv:2205.05477*, 2022.

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

[5] C. Y. H. Lee, G. Best, and G. A. Hollinger, "Optimal sequential stochastic deployment of multiple passenger robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8934–8940.

[6] B. Liu, X. Xiao, and P. Stone, "Team orienteering coverage planning with uncertain reward," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9728–9733.

[7] A. Smith, G. Best, J. Yu, and G. Hollinger, "Real-time distributed non-myopic task selection for heterogeneous robotic teams," *Autonomous Robots*, vol. 43, 03 2019.

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

[9] 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," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1435–1452, 2015.

[10] 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 *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 002 279–002 286.

[11] M. Kalaitzakis, B. Cain, N. Vitzilaios, I. Rekleitis, and J. Moulton, "A marsupial robotic system for surveying and inspection of freshwater ecosystems," *Journal of Field Robotics*, vol. 38, no. 1, pp. 121–138, 2021.

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

[13] A. Khatibi and S. Jacobson, "Doubly stochastic sequential assignment problem," *Naval Research Logistics (NRL)*, vol. 63, pp. n/a–n/a, 03 2016.

[14] R. Righter, "Stochastic sequential assignment problem with arrivals," *Probability in the Engineering and Informational Sciences*, vol. 25, no. 4, p. 477–485, 2011.

[15] ——, "The stochastic sequential assignment problem with random deadlines," *Probability in the Engineering and Informational Sciences*, vol. 1, no. 2, pp. 189–202, 1987.

[16] T. Feng and J. C. Hartman, "The sequential stochastic assignment problem with postponement options," *Probability in the Engineering and Informational Sciences*, vol. 27, no. 1, p. 25–51, 2013.

[17] A. Lee and S. Jacobson, "Sequential stochastic assignment under uncertainty: Estimation and convergence," *Statistical Inference for Stochastic Processes*, vol. 14, pp. 21–46, 02 2011.

[18] H. Wu, A. Ghadami, A. E. Bayrak, J. M. Smereka, and B. I. Epureanu, "Impact of heterogeneity and risk aversion on task allocation in multi-agent teams," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7065–7072, 2021.

[19] 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, pp. 277–294, 2013.