Re-Deployment Algorithms for Multiple Service Robots to Optimize Task Response

Armin Sadeghi Stephen L. Smith

Abstract— **This paper focuses on the problem of deploying a set of autonomous robots to efficiently service tasks that arrive sequentially in an environment over time. Each task is serviced when the robot visits the corresponding task location. Robots can then redeploy while waiting for the next task to arrive. The objective is to redeploy the robots taking into account the next N task arrivals. We seek to minimize a linear combination of the expected cost to service tasks and the redeployment cost between task arrivals. In the single robot case, we propose a one-stage greedy algorithm and prove its optimality. For multiple robots, the problem is NP-hard, and we propose two constant-factor approximation algorithm, one for the problem with a horizon of two task arrivals and the other for the infinite horizon when redeployment cost is weighted more heavily than service cost. Finally, we present extensive benchmarking results to characterize both solution quality and runtime.**

I. Introduction

In service applications, robots are tasked with responding to requests for service that arrive periodically over time [1]. For example, in a hospital setting [2] a fleet of robots may be used to provide assistance to patients by traveling to their locations. A key aspect in such applications is where the robots should wait (i.e., their deployment locations) to optimally respond to the next service request. After a request has been serviced, the robots can redeploy in order to re-optimize their positions for future requests. There is an inherent trade off between the expected response time for a service request and the cost incurred to redeploying robots between successive requests.

In this paper, we focus on the problem of deploying a set of robots to service tasks arriving sequentially in an environment. The robots' motion in the environment is captured as a road-map (i.e., graph), and each task arrives at a node of the graph according to a known probability distribution. A group of k robots moves on a common road-map, and tasks arrive at the vertices of the road-map. A task is serviced by a robot traveling to the task location. At each task arrival, we consider minimizing the response time and the redeployment cost (i.e., the cost of transitioning to new deployment locations). The robots choose to redeploy to another location between task arrivals. We consider this objective on a horizon of next N task arrivals.

Related Work: The task allocation problem has been the subject of extensive research [3], [4], [5], [6], [7]. Some of the most related work is [3], [4], where tasks are assigned dynamically to the robots in free space, and the objective is to deploy the robots such that the response time is minimized. In contrast, we focus on dynamically assigning tasks for robots moving on a roadmap, and look to minimize both response time and redeployment cost.

The facility location problem [8], [9] is the problem of installing facilities in a set of locations with a fixed cost of opening a facility. Demands arrive at the different locations and the objective is to minimize the time to respond to the demand and the total cost of opening facilities. A special case of the facility location problem is the k-median problem [9] where the cost of opening facilities is zero.

An extension to the facility location problem is the mobile facility location problem (MFL), introduced in [10]. The objective is to move the facilities while minimizing the total movement cost and the response time. In [11], authors provide a simple local-search algorithm for the MFL with a $3 + o(1)$ approximation ratio. In [12], an extensive set of experiments conducted on the algorithm characterize its performance in solution quality and the run-time. The two main differences between our problem and the MFL are 1) robots service tasks by visiting the task location, thus the configuration changes with each arrival and 2) MFL considers just the next arrival and plans the next waiting configuration for a single-ahead stage. In contrast, we plan the next configuration of the robots for a horizon of N task arrivals.

Online algorithms are presented in the literature for the facility location problem [13], [14]. The results are also extended to MFL with stochastic demands [15], [16]. A related problem is the k-server [17] problem in which tasks arrive sequentially over time, and each task is serviced by visiting the corresponding task location. However, the problem is defined on a metric space, rather than a road-map, and the objective is to minimize the server travel over the worst-case set of task arrivals. In contrast, we consider a known spatial distribution for service requests, capturing scenarios where prior information is available on the frequency of service requests.

Another closely related area of research is dynamic vehicle routing (DVR) [18]. In DVR tasks arrive sequentially over time according to a stochastic process. The most closely related results are on light-load policies, where the arrival rate of tasks is low. However, these problems consider only the service quality as a metric, and look at arrivals in the Euclidean plane rather than roadmaps.

Contribution: Our first contribution is to formulate the redeployment problem as a multi-stage optimization. We then cast the problem as a dynamic program (DP), and show that in the single robot case, a simple greedy policy is optimal.

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (a6sadegh@uwaterloo.ca; stephen.smith@uwaterloo.ca)

Then we propose two simple policies for the multiple robot problem with theoretical performance guarantees. In the first policy, built on the existent α-approximation algorithm [11] considering only one arrival, we provide a 5α-approximation algorithm for the two-stage problem. In the second policy, the planning horizon is extended to the infinity. We show that if robots are initiated in the k-median, then the policy is a 2α-approximation in a setting prioritizing the expected service cost. Finally, we simulate the two policies on real-world scenarios, namely high arrival rates and unknown probability distribution of tasks, and provide performance of the policies on both random environments and a work place floor plan.

The paper is organized as follows. In Section II, we formulate the problem as a dynamic program. In Section III, we provide a policy for the single robot problem. Section IV consists of upper and lower bounds for the optimal value and two policies for the multi-robot problem. Finally, in Section V, we provide benchmarking results.

II. Problem Formulation

Consider a set of k robots and a set of locations $\mathcal{V}$ for the robots to wait and tasks to arrive. Let $G = (\mathcal{V}, \mathcal{E}, c)$ be a metric graph on vertices $\mathcal{V}$, let $\mathcal{E}$ be the edge set and $c : \mathcal{E} \to \mathbb{R}_+$ be the cost function defined on the edge set satisfying the triangle inequality. A vertex of the graph represents a state of the robots, e.g. position and heading for the Dubins car, and cost of an edge represents the time robot takes to travel between the states.

A subset of $\mathcal{V}$ occupied by the robots at a time stage is called a configuration and it is denoted by $Q \in \mathcal{Q}$ where $\mathcal{Q}$ is the set of all the possible configurations, i.e., all subsets of $\mathcal{V}$ with size k.

The probability that a task arrives on vertex $v \in \mathcal{V}$ is $p_v \geq 0$, where $\sum_{v \in \mathcal{V}} p_v = 1$. Task arrivals occur sequentially, with the time between arrivals sufficient for the robots to reconfigure between arrivals. We explore the effect of relaxing this assumption in Section V. This is analogous to light load in DVR [18]. The task locations are independent and identically distributed. A robot is assigned to the task after the task arrival and services by visiting its location.

We consider a multi-objective problem of minimizing a linear combination of the response time and the redeployment cost, where the latter part captures energy consumption, or additional travel. The redeployment cost between two configurations is denoted by $\text{Assgn}(Q_1, Q_2)$ and is given by the minimum cost for robots to transition from Q_1 to Q_2, i.e., assignment cost. The minimum cost redeployment between two configurations is the minimum cost assignment of the vertices of two sets Q_1 and Q_2. Let $d(Q, u)$ be the closest distance from the vertices of Q to u, then we define the response time of a configuration as the expected distance to the next arrival, i.e., $\sum_{u \in \mathcal{V}} p_u d(Q, u)$. The configuration after sending the closest robot in Q to u is denoted by Q_u.

Figure 1 illustrates the two stages of the problem. Robots after servicing task $i-1$ at v are at configuration $Q_{i-1,v}$ (Figure 1a), i.e, closest robot in Q_{i-1} moved to v. Figure 1b illustrates the redeployment to configuration Q_i waiting for the next task to arrive. Task i arrives at u and the closest robot moves to service the task (Figure 1c), and finally the robots redeploy to Q_{i+1} (Figure 1d).

The cost incurred at each stage i is linear combination of the service time and the redeployment cost, i.e., $\text{Assgn}(Q_{i-1}, Q_i) + \beta \sum_{u \in \mathcal{V}} p_u d(Q_i, u)$, where β is a user-defined variable. The large values of β corresponds to a scenario where minimizing the response time is the priority and the small values of β prioritize the relocation cost.

In general, the optimal configuration at each stage depends on the previous arrivals, therefore, we have to account for the future tasks arrival at each stage. Therefore, we consider problem of minimizing the discounted stage costs over N next task arrivals. Let $V_i(Q)$ be the expected cost incurred during stages $i, \ldots, N$ at current configuration Q, then we can write the deployment problem as follows:

$$V_i(Q_{i-1}) = \text{Assgn}(Q_{i-1}, Q_i) + \beta \sum_{u \in \mathcal{V}} p_u d(Q_i, u) + \gamma \sum_{u \in \mathcal{V}} p_u V_{i+1}(Q_{i,u}). \tag{1}$$

Note that if N approaches infinity, then the problem becomes a discounted factor infinite horizon problem with $\gamma \in [0, 1)$.

For small values of β and γ, the problem corresponds to the case where minimizing the transition cost between the configurations is dominant. For instance, with $\beta = 0$ the optimal policy at each stage is $Q_i = Q_{i-1}$. In the case of large β and $\gamma = 0$, the optimal policy at stage i approaches the k-median solution [9], i.e., $\min_{Q_i \in \mathcal{Q}} \sum_{u \in \mathcal{V}} p_u d(Q_i, u)$.

The well-known k-median problem is a special case of the single-stage problem, therefore, the single-stage problem is NP-hard [11], which in turn implies that Problem (1) is NP-hard.

In the next section, we provide a policy for the single robot case, and in Section IV we discuss multiple robots problem.

III. Optimal Policy for Single Robot

In this section, we provide the optimal policy for the single robot case. The greedy approach to the problem for a single robot is to plan for a single stage without considering future arrivals. Let $\Pi_1(Q_{i-1})$ be the greedy policy at configuration Q_{i-1}, which returns the minimizer of the single stage, i.e.,

$$\Pi(Q_{i-1}) = \text{argmin}_{Q_i \in \mathcal{Q}} \text{Assgn}(Q_{i-1}, Q_i) + \beta \sum_{u \in \mathcal{V}} p_u d(Q_i, u). \tag{2}$$

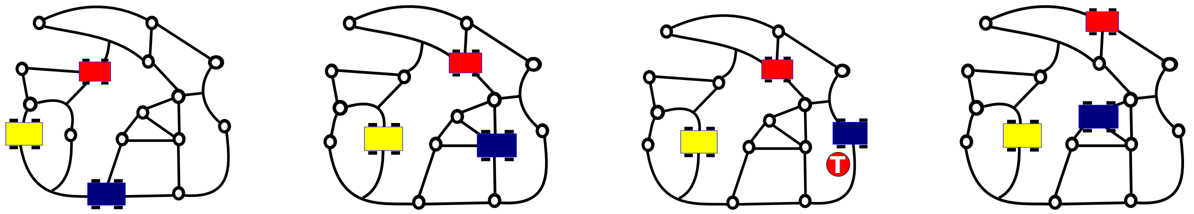
In Lemma III.1, we show that the greedy policy is optimal.

Lemma III.1. *Given a single robot, the greedy policy in* (2) *is optimal.*

Proof. Observe that the $Q_{i,u}$ of Problem 1 in the single robot case is $\{u\}$ regardless of the choice of Q_i, therefore, $V_{i+1}(Q_{i,u})$ is independent of Q_i at step i and only depends on the probability distribution. Therefore, the optimal policy for Problem (1) with single robot is the minimizer of

$$\text{argmin}_{Q_i \in \mathcal{Q}} \text{Assgn}(Q_{i-1}, Q_i) + \beta \sum_{u \in \mathcal{V}} p_u d(Q_i, u).$$

This is the greedy policy in (2). $\square$



(a) Configuration $Q_{i-1,v}$, serviced task $i-1$ at v and transitioning to the new waiting configuration.

(b) Configuration Q_i, waiting for task arrival.

(c) Task i arrives at u, transition to configuration $Q_{i,u}$.

(d) New waiting configuration Q_{i+1}

Fig. 1: Demonstrating a single stage of the problem. (a) Configuration of the robots after servicing task t_{i-1} at v (b) New waiting location of the robots for the new task arrival (c) Task arrival and the closest robot deployed to service the task (d) New waiting location of the robots for the next arrival.

(a) (b)

Fig. 2: Example of different solutions of single-stage optimal and infinite horizon optimal solutions. The dark vertices are occupied with the robots. (a) shows the initial configuration and the optimal solution of single-stage. (b) shows the optimal policy for the given initial configuration in infinite horizon.

Complexity of greedy policy: The greedy policy considers all the vertices in V. The $\mathrm{Assgn}(Q_{i-1}, Q_i)$ function is $O(1)$ and $\sum_{u\in\mathcal{V}} p_u d(Q_i, u)$ is $O(|V|)$. Therefore, the total runtime of the greedy policy at each stage is $O(|V|^2)$.

IV. Multiple Robots

In this section, we investigate the optimal policy for the multiple robots. In Section III, the optimal policy at each arrival is independent of the previous choices of the configurations, and thus the greedy policy is optimal. However, this is not the case for the multiple robots.

Example IV.1. Consider an instance of the problem with two robots on a graph with vertices located at $\{(0,0),(0,2),(2,0),(2,2),(1,1)\}$ in the Euclidean space (see Figure 2a). Let $\beta = 5$, $\gamma = 0.9$ and equal probability on the vertices, then for any configuration that does not contain the mid-vertex, the single stage optimal policy is to stay at the configuration, however, the optimal policy for infinite horizon is to move one of the robots to the mid-point (see Figure 2b).

The infinite horizon problem, namely Problem (1) is a stationary problem, i.e. optimal policy at each configuration is equal for every stage i. Thus, the problem can be written as a dynamic program as follows:

$$V(Q) = \min_{Q'\in\mathcal{Q}} \{\mathrm{Assgn}(Q, Q') + \beta \sum_{u\in\mathcal{V}} p_u d(Q', u) + \gamma \sum_{u\in\mathcal{V}} p_u V(Q'_u)\} \quad \forall Q \in \mathcal{Q}. \tag{3}$$

The dynamic program (DP) is optimally solved via several well-known approaches such as policy iteration, value iteration, or linear programming. Observe that for an instance of k robots and n tasks locations, there exist $\binom{n}{k}$ configurations – exponential in the number of robots. The main drawback of the dynamic programming approaches is that finding the optimal policy for a single configuration requires solving DP (3) for all the configurations, which is not computationally tractable for large instances.

In this section, we propose two simple policies and we provide some results on the performance of them. First we provide bounds on the optimal value of V, which is essential in the performance analysis of the two policies.

A. Bounding the value function

In this section, we provide bounds on the optimal value function V of configurations for the infinite horizon problem. These bounds are essential in our analysis of the simple policies. For simplicity we define the following notations:

$$D(Q) := \sum_{u\in\mathcal{V}} p_u d(Q, u); \text{ and}$$
$$c(Q_1, Q_2) := \mathrm{Assgn}(Q_1, Q_2) + \beta D(Q_2).$$

The term $D(Q)$ is the expected response time of the robots at configuration Q, and $c(Q_1, Q_2)$ is the stage cost for redeployment from Q_1 to Q_2.

The following results provide an upper-bound on the optimal value for a given configuration.

Lemma IV.2 (Upper-bound)**.** *The cost of the optimal policy for $Q \in \mathcal{Q}$ is upper-bounded by*

$$V(Q) \le \min_{Q'\in\mathcal{Q}} \{c(Q, Q') + \gamma \frac{\beta + 1}{1-\gamma} D(Q')\}.$$

Proof. For any $Q' \in \mathcal{Q}$ we have,

$$V(Q) \le c(Q, Q') + \gamma \sum_{v\in\mathcal{V}} p_v V(Q'_v). \tag{4}$$

Therefore, we need to upper-bound $V(Q'_v)$ for all $v \in \mathcal{V}$. Note that $V(Q'_v)$ is the cost of the optimal policy for the configuration Q'_v. Then we have,

$$V(Q'_v) \le c(Q'_v, Q') + \gamma \sum_{u\in\mathcal{V}} p_u V(Q'_u).$$

Note that $\text{Assgn}(Q'_v, Q) = d(Q, v)$, thus we have,

$$V(Q'_v) \leq d(Q', v) + \beta \sum_{u \in \mathcal{V}} p_u d(Q', v) + \gamma \sum_{u \in \mathcal{V}} p_u V(Q'_u).$$

By taking expectation on $V(Q'_v)$, we obtain

$$\sum_{v \in \mathcal{V}} p_v V(Q'_v) \leq (\beta + 1) \sum_{v \in \mathcal{V}} p_v d(Q', v) + \gamma \sum_{v \in \mathcal{V}} p_u V(Q'_u).$$

Finally we have,

$$\sum_{v \in \mathcal{V}} p_v V(Q'_v) \leq \frac{\beta + 1}{1 - \gamma} \sum_{v \in \mathcal{V}} p_v d(Q', v) \tag{5}$$

The result follows directly from Equations (4) and (5). □

Let $\Pi_1^* : \mathcal{Q} \to \mathcal{Q}$ be the optimal policy for the single stage problem, i.e.,

$$\Pi_1^*(Q) = \arg\min_{H \in \mathcal{Q}} \text{Assgn}(Q, H) + \beta \sum_{u \in \mathcal{V}} p_u d(H, u) \tag{6}$$

and let J be the minimizer of the k-median problem, i.e, $J = \arg\min_{J \in \mathcal{Q}} \sum_{u \in \mathcal{V}} p_u d(J, u)$.

Let Q^* be the configuration associated with the optimal policy at Q, i.e., $Q^* = \Pi^*(Q)$, then we establish following lower-bound on the optimal value.

Lemma IV.3 (Lower-bound)**.** *The cost of the optimal policy for $Q \in \mathcal{Q}$ is lower-bounded by*

$$V(Q) \geq c(Q, \Pi_1^*(Q)) + \gamma \frac{\beta}{1 - \gamma} \sum_{u \in \mathcal{V}} p_u d(J, u).$$

Proof. Since moving to Q^* is the optimal action in Q, then we have,

$$V(Q) = c(Q, Q^*) + \gamma \sum_{v \in \mathcal{V}} p_v V(Q^*_v). \tag{7}$$

By the definition of Π_1^*, we have, $c(Q, \Pi_1^*(Q)) \leq c(Q, Q^*)$. Then to prove the result, it suffices to show that

$$\sum_{v \in \mathcal{V}} p_v V(Q^*_v) \geq \frac{\beta}{1 - \gamma} \sum_{u \in \mathcal{V}} p_u d(J, u).$$

Let $\Pi^*(Q)$ be the function takes a configuration as input and returns the optimal configuration to visit. Then $\Pi^*(Q)_u$ is configuration where the closest robot at $\Pi^*(Q)$ to u has moved to u. Thus we have,

$$V(Q^*_v) \geq \beta \sum_{u \in \mathcal{V}} p_u d(J, u) + \gamma \sum_{u \in \mathcal{V}} p_u V(\Pi^*(Q^*_v)_u).$$

For the simplicity of the notation, let $M = \Pi^*(Q^*_v)$, then we can write similar inequalities for each M_u as follows:

$$V(M_u) \geq \beta \sum_{w \in V} p_u d(J, w) + \gamma \sum_{w \in V} p_w V(\Pi^*(M_u)_w).$$

Therefore, we have,

$$\begin{aligned} V(Q^*_v) \geq \beta \sum_{u \in \mathcal{V}} p_u d(J, u) + \gamma\beta \sum_{w \in V} p_u d(J, w) \\ + \gamma^2 \sum_{u \in \mathcal{V}} p_u \sum_{w \in V} p_w V(\Pi^*(M_u)_w). \end{aligned}$$

Observe that we can repeatedly write the similar inequalities for each stage, then we have,

$$\begin{aligned} V(Q^*_v) &\geq \beta \sum_{u \in \mathcal{V}} p_u d(J, u)(1 + \gamma + \gamma^2 + \ldots) \\ &= \frac{\beta}{1 - \gamma} \sum_{u \in \mathcal{V}} p_u d(J, u). \end{aligned} \tag{8}$$

Finally, the result is immediate by Equations (7) and (8). □

In the rest of this section, we provide our two simple policies for deployment of a system of multiple robots.

B. Policy I: Approximation Algorithm for TWO-STAGE-HORIZON

First, from the problem definition in Section II, given a configuration Q_{i-1}. the two-stage problem is to minimize:

$$\min_{Q_i \in \mathcal{Q}} c(Q_{i-1}, Q_i) + \gamma \sum_{u \in \mathcal{V}} p_u c(Q_{i,u}, \Pi_1(Q_{i,u})) \tag{9}$$

where Π_1 is a policy defined on $\Pi_1 : \mathcal{Q} \to \mathcal{Q}$. With a simple observation, we can show that Π_1 is Π_1^*. Our algorithm for the two-stage problem is built on the existent algorithms for the single-stage problem with multiple robots. We provide the algorithm for TWO-STAGE-HORIZON in Algorithm 1, and we prove approximation results in Theorem IV.4.

Built on the approximation algorithms for the single-stage problem in [11], our algorithm for the two-stage problem consists of three single stage-problems. The algorithm evaluates the solutions to the single-stage problems in Line 7 and returns the better solution. Function SINGLE-STAGE(β, Q) in Algorithm 1 returns the solution to the single-stage problem, i.e., Equation (6). The run-time of the algorithm is dictated by the existent single-stage algorithm [11].

In Section V, we show on an extensive set of experiments that the TWO-STAGE-HORIZON policy outperforms the SINGLE-STAGE policy in a wide range of β and γ values.

Algorithm 1

```
1: function TWO-STAGE-HORIZON(G, β, γ, Q)
2:     if β < 1 then
3:         return SINGLE-STAGE(2γβ, Q)
4:     else
5:         H_1 ← SINGLE-STAGE((β+γ)/(1+γ), Q)
6:         H_2 ← SINGLE-STAGE(β + γ + βγ, Q)
7:         Evaluate H_1 and H_2 for Problem (9) and return
           the minimally-valued configuration.
8:     end if
9: end function
```

Now we establish the following result on the problem.

Theorem IV.4. *Suppose there exists an α-approximation algorithm for the single-stage problem, then Algorithm 1 is a 5α-approximation algorithm for the two-stage problem.*

To prove the theorem, we are required four intermediate results, and the proof of each is in the Appendix.

We divide our analysis into two cases i) $\beta < 1$ and ii) $\beta \geq 1$. First we establish the following observation on the single-stage problem with $\beta < 1$.

Observation IV.5. If $\beta < 1$, then $\Pi_1^*(Q) = Q$ for all $Q \in \mathcal{Q}$.

According to Observation IV.5, the optimal policy for the single-stage problem with $\beta < 1$ is to stay at the configuration and wait for the next task arrival.

Under Observation IV.5, the two stage problem becomes

$$\min_{Q_i \in \mathcal{Q}} c(Q_{i-1}, Q_i) + \gamma\beta \sum_{v \in \mathcal{V}} p_v D(Q_{i,v}). \tag{10}$$

Now consider the following single-stage problem:

$$\min_{Q_i \in \mathcal{Q}} c(Q_{i-1}, Q_i) + 2\gamma\beta D(Q_i) \tag{11}$$

Observe that Problem (11) is the single-stage problem in Line 3 of Algorithm 1. Now we provide the following result on the quality of the solution obtained from Problem (11) as a solution for Problem (10).

Lemma IV.6. *Suppose there exists an α-approximation algorithm for the single-stage problem, then the solution obtained from Problem (11) provides an $\alpha(1+2\gamma)$-approximation for Problem (10).*

Hence, Algorithm 1 is a 3α-approximation for the two-stage problem with $\beta < 1$. Consider the following problem:

$$\min_{Q_i, Q_{i+1} \in \mathcal{Q}} c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma c(Q_i, \Pi_1(Q_i)). \tag{12}$$

Now we establish the following result on Problem (12).

Lemma IV.7. *Suppose there exists an η-approximation algorithm for Problem (12), then there exist $(1 + 2\gamma/\beta)\eta$-approximation algorithm for Problem (9).*

Note that Problem (12) is not a single-stage problem, and similar to the previous analysis, we will define two alternative problems to Problem (12), and then we will evaluate the quality of the solutions of the problems. Consider the following two problems:

$$\min_{Q_i \in \mathcal{Q}} \text{Assgn}(Q_{i-1}, Q_i) + (\beta + \gamma + \beta\gamma) D(Q_i) \tag{13}$$

$$\min_{Q_i \in \mathcal{Q}} \text{Assgn}(Q_{i-1}, Q_i) + \frac{\beta + \gamma}{1 + \gamma} D(Q_i) + \frac{\gamma}{1+\gamma} c(Q_{i-1}, \Pi_1(Q_{i-1})) \tag{14}$$

Observe that in Problem (14), the first two terms and the third term are two independent single-stage problems, therefore, it can be approximately solved under assumption of Theorem IV.4 with two independent calls to the single-stage approximation algorithm.

Lemma IV.8. *Suppose there exists an α-approximation algorithm for single-stage problem, then*

(i) *a solution to Problem* (13) *provides an $\alpha\beta$-approximation for Problem* (12)*; and*

(ii) *a solution to Problem* (14) *provides an $\alpha(1 + 2\gamma)$-approximation for Problem* (12).

With these results we now prove Theorem IV.4.

Proof of Theorem IV.4. Algorithm 1 evaluates the solution for the two problems in Line 7 and returns the better solution. An immediate result of Lemmas IV.7 and IV.8, is that the approximation factor for Algorithm 1 with $\beta \geq 1$ is

$$\min\{\alpha\beta, \alpha(1+2\gamma)\}(1 + 2\frac{\gamma}{\beta}) \leq \alpha(1 + 4\gamma) \leq 5\alpha. \quad \square$$

A $3+o(1)$- approximation algorithm for single-stage problem and k-median is provided in [11]. Hence, Algorithm 1 is a $9 + o(1)$ approximation algorithm for the TWO-STAGE-HORIZON with $\beta < 1$ and $15 + o(1)$ with $\beta > 1$.

C. Policy II: Move to k-median

An intuitive policy for Problem (1) is to move the robots to the k-median solution in between the arrivals. In fact it is shown to be the optimal policy in the light load if the transition cost between the configurations is negligible compared to the service cost [18]. Let Π_{median} denote this policy, and $V_{\text{median}}(Q)$ be the value function under this policy. Then we establish the following result.

Lemma IV.9. *Suppose an α-approximation algorithm for k-median returns Q, then the value function under Π_{median} is at most $\alpha(1 + \gamma/\beta)$ times the optimal, i.e.,*

$$V_{\text{median}}(Q) \leq \alpha(1 + \frac{\gamma}{\beta}) V(Q).$$

Proof. First observe that the value function under this policy, namely $V_{\text{median}}(Q)$, is as follows:

$$V_{\text{median}}(Q) = c(Q, Q) + \gamma \sum_{u \in \mathcal{V}} p_u V_{\text{median}}(Q_u).$$

Also notice that the $V_{\text{median}}(Q_u)$ is bounded as follows, where the equality is ensured if Q is the only k-median.

$$V_{\text{median}}(Q_u) \leq c(Q_u, Q) + \gamma \sum_{v \in \mathcal{V}} p_v V_{\text{median}}(Q_v).$$

Therefore, With the same stages in the proof of Lemma IV.2, we have $V_{\text{median}}(Q) \leq \frac{\beta+\gamma}{1-\gamma} \sum_{u \in \mathcal{V}} p_u d(Q, u)$.

Also from Lemma IV.3 we have, $V(Q) \geq \frac{\beta}{1-\gamma} \sum_{u \in \mathcal{V}} p_u d(J, u)$, where J is the actual k-median. Since Q is an α-approximation for the k-median,

$$V_{\text{median}}(Q) \leq \alpha(1 + \frac{\gamma}{\beta}) V(Q). \quad \square$$

In fact, the bound shows that the value for this policy approaches the optimal for large β. In other words, there exists a $\bar{\beta}$ such that for all $\beta \geq \bar{\beta}$ this policy is optimal. This result captures the optimal policy results for DVR in [18]. However, the policy can be arbitrarily sub-optimal for small values of β. For instance, consider a problem with $\beta = 0$ and $\gamma > 0$. In this case $V(Q) = 0$ while $V_{\text{median}}(Q) > 0$ due to the transitions to the k-median between task arrivals.

The policy does not provide desired behavior for small values of β. In the next section, we provide simulation results for two policies in the infinite horizon setting and evaluate their performance for different values of γ and β.

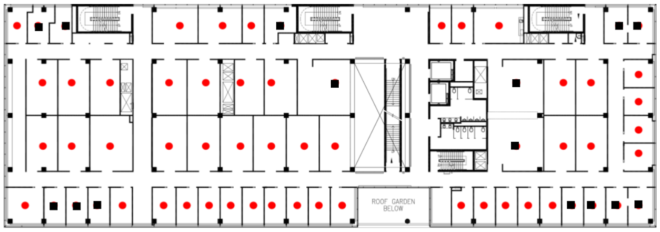
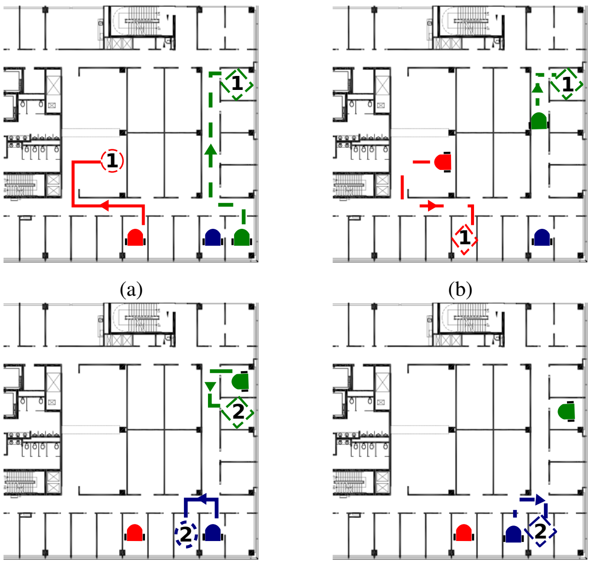


Fig. 3: Floor plan of E5 building at University of Waterloo. Floor plan contains 62 task locations (circles and squares) and 15 potential waiting locations for the robots (squares).



(a) (b) (c) (d)

Fig. 4: Two stages of the TWO-STAGE-HORIZON performing two tasks and two redeployments. The circles represent the task arrivals and diamonds are the waiting locations.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the two policies in Section IV in random environments and a sample work place (see Figure 3). To simulate the real-world applications, we also evaluate the performance of the policies in environments with unknown probability distribution and high task arrival rates.

In all the experiments we consider the infinite horizon problem, and the expected cost at each Q for policy Π is obtained by solving the following set of equations, known as policy evaluation:

$$V(Q) = c(Q, \Pi(Q)) + \gamma \sum_{u \in \mathcal{V}} V(\Pi(Q)_u) \quad \forall Q \in \mathcal{Q}.$$

Figure 4 shows two stages of the TWO-STAGE-HORIZON policy in a part of the work-place floor plan with $\beta = 5$ and $\gamma = 0.9$. The circles represent the tasks and the diamonds are the waiting locations. The numbers at each location represent the sequence of the robot movements. In stage 1 the red robot services a task and to redeploy, the red robot returns to its location and the green robot moves to diamond 1. In stage 2, the blue robot services a task and then returns to its waiting location while the green robot moves to diamond 2.

Random Environment: In this experiment, we compare the average expected cost of all configurations obtained from different policies to the optimal value from DP in random environments. In each instance, 4 robots are servicing a set of 20 tasks randomly generated in the Euclidean space of size 10×10. Recall that the proposed policies in Section IV are polynomial in the input, on the other hand, the exact DP problem grows exponentially in the number of robots which limits the size of experiments. The probability of a task arriving in each location is independent and identically distributed. Figure 5 shows the average error percentage (i.e., average of $(\textit{value function} - \textit{optimal value})/\textit{optimal value}$ over all configurations) of three policies for different values of β and $\gamma = 0.9$. For each β, a set of 2000 random instances are generated and each box depicts the median, first quartile and third quartile. The DP is solved via the policy iteration method with an average time of 894.4 seconds on an Intel Core$i5$ @3.6Ghz processor. The average computation time of the TWO-STAGE-HORIZON policy and the MOVE-TO-MEDIAN policy for a single configuration are 0.0078 seconds and 0.0027 seconds, respectively.

Figure 5a illustrates the asymptotic improvement in the performance of the MOVE-TO-MEDIAN policy as β increases. In contrast, the greedy SINGLE-STAGE policy performs almost optimally when the deployment cost and response time are equally weighted. However, the TWO-STAGE-HORIZON policy provides a solution within 5% of the optimal for all β, out-performing MOVE-TO-MEDIAN policy at $\beta = 2$ and single stage policy at $\beta = 5$ by 18.1% and 10.1% on average, respectively.

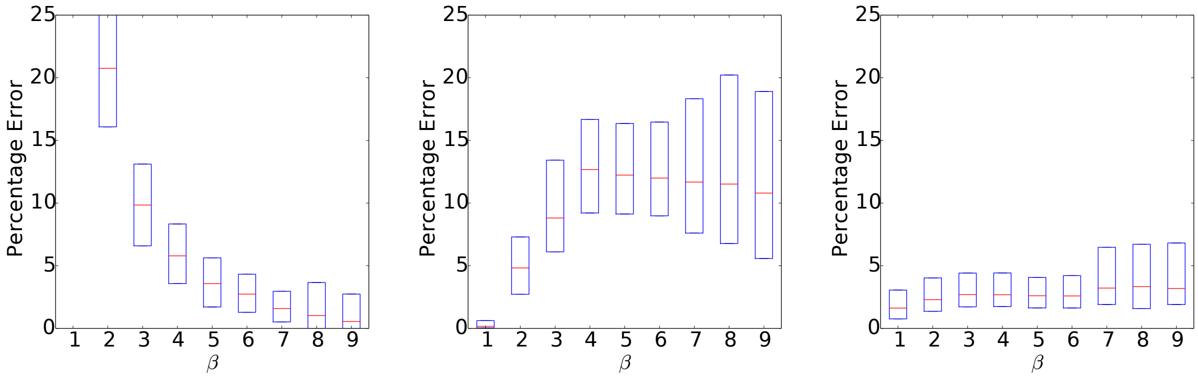
We observed the similar behavior of the three policies when varying γ. For small γ, the single-stage policy provides near-optimal solutions. As γ increases, the MOVE-TO-MEDIAN policy improves in quality. Similar to the previous experiment, the TWO-STAGE-HORIZON policy provides better solutions in for a large range of γ values.

Unknown Probability Distribution: In this experiment, the spatial probability distribution of the tasks is unknown to the robots. The robots, starting with a uniform Dirichlet prior (with all parameters equal to 2), update their estimate of the distribution according to the maximum a posteriori estimator [19]:

$$\hat{p}_u = \frac{\text{number of arrivals at } u + 1}{\text{number of arrivals} + \text{number of task locations}}.$$

The environment is a floor plan with 62 task locations and 5 robots. In some applications, e.g. hospital setting, the robots cannot stay at the task locations, therefore, the potential waiting location of the robots are limited to 15 locations marked with squares in Figure 3. The probability distribution is drawn from a Dirichlet distribution where the task locations are weighted with their x-coordinates, i.e. the probability of a task is higher for vertices on the right side.

We compare performance to a policy that has access to the true distribution over 80 task arrivals. Figure 6 shows the average total stage costs for $\beta = 5$ and 10 over 15000



(a) MOVE-TO-MEDIAN (b) SINGLE-STAGE (c) TWO-STAGE-HORIZON

Fig. 5: Average percentage error of different policies

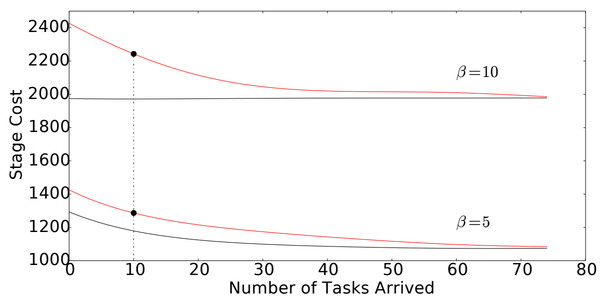


Fig. 6: TWO-STAGE-HORIZON with unknown probability distribution.

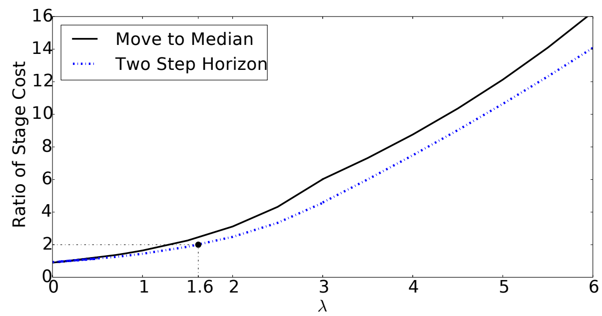


Fig. 7: Stage cost of TWO-STAGE-HORIZON and MOVE-TO-MEDIAN policies as a function of task arrival rate.

instances. Observe that a poor estimation of the probability distribution in the first few arrivals propagates proportionally with the value of β. The MOVE-TO-MEDIAN policy is equivalent to the two-stage policy for sufficiently large β. Therefore, the MOVE-TO-MEDIAN is more vulnerable to error in the estimate.

Different Arrival Times: In Section II, we assumed that the time between arrivals is sufficient for the robots to reconfigure. Now we repeat the experiment with $\beta = 5$ and $\gamma = 0.9$ on the environment given in Figure 3 for different task arrival rates. The time to service a task at u is the difference between the arrival time and the time that a robot visits the task location. We perform 15,000 simulations, each containing a sequence of 80 tasks arrivals according to the Poisson process with parameter λ, where λ represents the number of tasks arrive in a time unit. The *time unit* on the x-axis is the expected time to service a task from the k-median and return, i.e. $D(J)$. Figure 7 shows the ratio of the average stage cost of the two policies to the average stage cost of the TWO-STAGE-HORIZON policy with $\lambda \approx 0$. The stage cost increases exponentially with λ. Notice that when there are approximately 1.6 arrivals per time unit, and thus new tasks frequently arrival while the robots are redeploying, the average stage cost increases by a factor of 2 from the cost for $\lambda \approx 0$ (i.e., the regime where our assumption holds).

VI. CONCLUSION

This paper considers the problem of deploying a set of robots to efficiently service tasks that arrive sequentially in an environment over time. The presented policies provide a close to an optimal solution and show significant improvement in the run-time compared to DP. Also, various experiments show the robust behavior of the policies in application. In addition, the results for the experiment with unknown distribution and high arrival rate provides a direction for analyzing the performance of the policies in these scenarios.

APPENDIX

Proof of Observation IV.5. The statement $\Pi_1^*(Q) = Q$ is equivalent to $\beta \sum_{u\in\mathcal{V}} p_u d(Q,u) \leq \text{Assgn}(Q,F) + \beta\sum_{u\in\mathcal{V}} p_u d(F,u)$, for all $Q, F \in \mathcal{Q}$. Now consider a task at u and the closest robot to u in configuration Q, namely $\sigma_Q(u)$, then $d(u,\sigma_Q(u)) \leq d(u,v)$, $\forall v \in Q$.Let $v \in Q$ be the vertex matched with $\sigma_F(u)$ in the assignment $\text{Assgn}(F,Q)$. Therefore, we have,

$$d(u,\sigma_Q(u)) \leq d(u,v) \leq d(v,\sigma_F(u)) + d(u,\sigma_F(u)), \quad (15)$$

where the last inequality is due to the triangle inequality. Taking the expectation on both sides of Equation (15) gives

$$\sum_{u\in\mathcal{V}} p_u d(u,Q) \leq \text{Assgn}(Q,F) + \sum_{u\in\mathcal{V}} p_u d(u,F).$$

Therefore, the result follows immediately from $\beta < 1$. $\square$

Proof of Lemma IV.6. Let Q_i be the solution of the α-approximation algorithm to Problem (11), and Q_i^* be the optimal solution to Problem (10), then we have,

$$c(Q_{i-1}, Q_i) + 2\gamma\beta D(Q_i) \leq \alpha(c(Q_{i-1}, Q_i^*) + 2\gamma\beta D(Q_i^*)). \tag{16}$$

By the triangle inequality we have $\sum_{v\in\mathcal{V}} p_v D(Q_{i,v}) \leq 2\gamma\beta D(Q_i)$, then by Equation (16) the following holds,

$$c(Q_{i-1},Q_i) + \gamma\beta \sum_{v\in\mathcal{V}} p_v D(Q_{i,v}) \leq \alpha(1+2\gamma)\big(c(Q_{i-1}, Q_i^*) + \gamma\beta \sum_{u\in\mathcal{V}} p_u D(Q_{i,u}^*)\big). \square$$

Proof of Lemma IV.7. Consider Q_i, Q_{i+1} as the configurations returned by the approximation algorithm for Problem (12) and Q_i^* be the optimal configuration for Problem (9). By the assumption of the Lemma, we have,

$$c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_i, \Pi_1(Q_i)) \leq \eta\big(c(Q_{i-1}, Q_i^*) + \gamma D(Q_i^*) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_i^*, \Pi_1^*(Q_i))\big). \tag{17}$$

And also observe that by the triangle inequality we have,

$$\begin{aligned} &c(Q_{i-1}, Q_i^*) + \gamma D(Q_i^*, u) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_i^*, \Pi_1^*(Q_{i,u})) \\ &\leq c(Q_{i-1}, Q_i^*) + 2\gamma D(Q_i^*, u) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_{i,u}^*, \Pi_1^*(Q_{i,u})). \end{aligned} \tag{18}$$

Finally by Equations (17) and (18) we have,

$$c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_i, \Pi_1(Q_i)) \leq \eta(1 + \frac{2\gamma}{\beta})\big(c(Q_{i-1}, Q_i^*) + \gamma \sum_{u\in\mathcal{V}} p_u c(Q_i^*, \Pi_1^*(Q_{i,u}))\big). \square$$

Proof of Lemma IV.8. (i) Suppose Q_i be the solution obtained from the single-stage approximation algorithm for Problem (13). By the triangle inequality we have,

$$\begin{aligned} &c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma c(Q_i, \Pi_1(Q_i)) \\ &\leq \mathrm{Assgn}(Q_{i-1}, Q_i) + (\beta + \gamma + \beta\gamma) D(Q_i). \end{aligned} \tag{19}$$

Also observe that by the triangle inequality we have,

$$\begin{aligned} &\mathrm{Assgn}(Q_{i-1}^*, Q_i^*) + (\beta + \gamma + \beta\gamma) D(Q_i^*) \\ &\leq c(Q_{i-1}, Q_i^*) + \gamma D(Q_i^*) + \beta\gamma c(Q_i^*, \Pi_1^*(Q_i)). \end{aligned} \tag{20}$$

Finally, by applying the α-approximation factor of single-stage algorithm to Equations (19) and (20) we have,

$$\begin{aligned} &c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma c(Q_i, \Pi_1(Q_i)) \\ &\leq \alpha\beta\big(c(Q_{i-1}^*, Q_i^*) + \gamma D(Q_i^*) + \gamma c(Q_i^*, \Pi_1^*(Q_i^*))\big). \end{aligned}$$

(ii) Suppose Q_i and $\Pi_1(Q_{i-1})$ be the solutions obtained from the single-stage approximation algorithm for the two independent single-stage problems in Problem (14). By triangle inequality, we have

$$\begin{aligned} &c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma c(Q_i, \Pi_1(Q_i)) \\ &\leq (1+\gamma)\mathrm{Assgn}(Q_{i-1}, Q_i) + (\beta + \gamma) D(Q_i) \\ &+ \gamma c(Q_{i-1}, \Pi_1(Q_{i-1})). \end{aligned} \tag{21}$$

Observe that by the triangle inequality we have,

$$\begin{aligned} &(1+\gamma)\mathrm{Assgn}(Q_{i-1}^*, Q_i^*) + (\beta+\gamma) D(Q_i^*) \\ &+ \gamma c(Q_{i-1}^*, \Pi_1^*(Q_{i-1}^*)) \leq c(Q_{i-1}, Q_i^*) + \gamma D(Q_i^*) \\ &+ \gamma c(Q_i^*, \Pi_1^*(Q_i)) + 2\gamma \mathrm{Assgn}(Q_{i-1}, Q_i^*). \end{aligned} \tag{22}$$

Finally, by applying the α-approximation factor of single-stage algorithm to Equations (21) and (22) we have,

$$\begin{aligned} &c(Q_{i-1}, Q_i) + \gamma D(Q_i) + \gamma c(Q_i, \Pi_1(Q_i)) \\ &\leq \alpha(1+2\gamma)\big(c(Q_{i-1}, Q_i^*) + \gamma D(Q_i^*) + \gamma c(Q_i^*, \Pi_1^*(Q_i))\big). \end{aligned}$$

$\square$

References

[1] M. M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal, "Cobots: Robust symbiotic autonomous mobile service robots." in *IJCAI*, 2015, p. 4423.

[2] K. Niechwiadowicz and Z. Khan, "Robot based logistics system for hospitals-survey," in *IDT Workshop on interesting results in computer science and engineering*, 2008.

[3] J. Le Ny and G. J. Pappas, "Adaptive deployment of mobile robotic networks," *IEEE Transactions on automatic control*, vol. 58, no. 3, pp. 654–666, 2013.

[4] M. de Roo, P. Frasca, and R. Carloni, "Optimal event handling by multiple unmanned aerial vehicles," in *IEEE ICRA*, 2016, pp. 1230–1236.

[5] C. Nam and D. A. Shell, "When to do your own thing: Analysis of cost uncertainties in multi-robot task allocation at run-time," in *IEEE ICRA*, 2015, pp. 1249–1254.

[6] ——, "Bundling policies for sequential stochastic tasks in multi-robot systems," *International Symposium on Distributed Autonomous Robotic Systems*, 2016.

[7] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 364–378, 2012.

[8] D. B. Shmoys, "Approximation algorithms for facility location problems," in *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 2000, pp. 27–32.

[9] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM Journal on computing*, vol. 33, no. 3, pp. 544–562, 2004.

[10] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam, "Minimizing movement," *ACM Transactions on Algorithms (TALG)*, vol. 5, no. 3, p. 30, 2009.

[11] S. Ahmadian, Z. Friggstad, and C. Swamy, "Local-search based approximation algorithms for mobile facility location problems," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1607–1621.

[12] R. Halper, S. Raghavan, and M. Sahin, "Local search heuristics for the mobile facility location problem," *Computers & Operations Research*, vol. 62, pp. 210–223, 2015.

[13] A. Meyerson, "Online facility location," in *IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 426–431.

[14] D. Fotakis, "Incremental algorithms for facility location and k-median," *Theoretical Computer Science*, vol. 361, no. 2-3, pp. 275–313, 2006.

[15] A. Ghodselahi and F. Kuhn, "Serving online demands with movable centers," *arXiv preprint arXiv:1404.5510*, 2014.

[16] G. Divéki and C. Imreh, "Online facility location with facility movements," *Central European Journal of Operations Research*, vol. 19, no. 2, pp. 191–200, 2011.

[17] E. Koutsoupias, "The k-server problem," *Computer Science Review*, vol. 3, no. 2, pp. 105–118, 2009.

[18] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.

[19] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.