Dynamic Multi-Agent Team Forming: Asymptotic Results on Throughput Versus Delay

Stephen L. Smith Francesco Bullo

***Abstract*— In this paper we focus on problems in which tasks (demands for service) arrive in an environment sequentially over time. A task is completed when a robot (or team of robots) provides the required service, and the goal is to minimize the expected delay between a task's arrival, and its completion. We develop a general framework in which these problems can be described, and propose a set of scaling laws for studying the relationship between the number of robots, the expected task delay, and the task arrival rate. We describe two existing problems in our framework, namely the dynamic traveling repairperson problem, and the dynamic pickup delivery problem, and present their asymptotic performance. We then introduce the dynamic team forming problem, in which tasks require services that can be provided only through complex teams of heterogeneous robots. We determine a lower bound on the problem's achievable performance, and propose three policies for solving the problem. We show that for each policy, there is a broad class of tasks for which the policy's performance is within a constant factor the optimal.**

I. Introduction

Consider a heterogeneous fleet of mobile robotic agents deployed in an environment $\mathcal{E} \subset \mathbb{R}^2$. Each robot in the fleet is capable of providing certain services. Tasks, which consist of a set of required services, arrive in the environment sequentially over time. The fleet is notified of each task upon its arrival, and a task is completed once the fleet provides the required services. The goal is to minimize the expected delay between a task's arrival and its completion. Thus, it is a dynamic task allocation problem; determine which robots should service which tasks, and in what order.

In static task allocation problems, a set of tasks is given *a priori* and the goal is to assign vehicles in order to maximize the "score" of the mission. In [1] a taxonomy of task allocation problems is given, dividing problems into groups based on criteria such as the number of tasks a robot can execute, and the number of robots required for a task. In papers such as [2], [3], advanced heuristic methods are developed, and their effectiveness is demonstrated through extensive simulation or real world implementation.

In dynamic task allocation problems, tasks arrive sequentially over a period of time. Only once a task has arrived can the robots determine the method in which they will provide service. In the dynamic traveling repairperson problem (DTRP) [4], [5], the robots are homogeneous, and each task consists of a location which requires on-site service. Spatially distributed algorithms for the DTRP were developed in [6] and [7]. In the dynamic pickup delivery problem (DPDP) [8] the task consists of a source-destination pair. A message must be picked up from the source, and delivered to the destination. In [8] the message must be picked up and delivered by the same robot, and in [9] the message can be relayed to between robots. For both the DTRP and the DPDP, lower bounds are found on the expected task delay (which depend on quantities such as the task arrival rate, environment size, and the number of robots), and policies are proposed which provide delays within a constant factor of this lower bound. In dynamic task allocation problems the expected delay depends on the task arrival rate; if tasks arrive more rapidly, then the expected delay increases. This tradeoff is well known in ad hoc wireless networks [10], [11]; If nodes increase the rate at which they send messages (i.e., the throughput), then this increases the expected delay a message will incur before arriving at its destination.

In this paper we introduce a framework for describing dynamic task allocation problems. As in the work on wireless networks, we propose scaling laws which allow us to study the expected task delay as a function of the throughput of the robotic network (i.e., the rate at which tasks are serviced). We revisit the DTRP and DPDP, and present the existing results on expected delay under our scaling laws. We then introduce the *dynamic team forming problem*. The problem consists of a heterogeneous group of n robots in which each robot is capable of providing one of k services. Tasks appear in the environment which require some subset of the k services. Thus, for each task, a team of robots must be formed which is capable of providing the required services. We derive a lower bound on the expected delay of the dynamic team forming problem, and propose three policies; Complete team, Task-specific team, and Scheduled task-specific team. We show that for each policy there is a broad class of tasks for which the policy performs within a constant factor the optimal. Due to space constraints all proofs are omitted.

II. Background material

In this section we review results on the Euclidean traveling salesperson problem (ETSP), queueing theory, and vertex coloring in graphs. We let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{N}$ denote the set of real numbers, positive real numbers, and positive integers, respectively. For a finite set A, we let $|A|$ denote its cardinality, and for an infinite set $A \subset \mathbb{R}^2$ we let $|A|$ denote its area. For two functions $f, g : \mathbb{N} \to \mathbb{R}_{>0}$, we write $f(n) \in O(g)$ (respectively, $f(n) \in \Omega(g)$) if there exist $N \in \mathbb{N}$ and $c \in \mathbb{R}_{>0}$ such that $f(n) \leq cg(n)$ for all $n \geq N$ (respectively, $f(n) \geq cg(n)$ for all $n \geq N$). If $f(n) \in O(g)$ and $f(n) \in \Omega(g)$, then we say $f(n) \in \Theta(g)$.

This material is based upon work supported in part by ARO MURI Award W911NF-05-1-0219 and ONR-N00014-07-1-0721.

S. L. Smith and F. Bullo are with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106, USA, {stephen,bullo}@engineering.ucsb.edu

a) The Euclidean Traveling Salesperson Problem: For a set $\mathcal{Q}$ of n points in $\mathbb{R}^2$, let $\mathrm{ETSP}(\mathcal{Q})$ denote the length of the shortest closed path through all points in $\mathcal{Q}$. The following result characterizes the length of this path when $\mathcal{Q} \subset \mathcal{E}$, where $\mathcal{E}$ is a square environment with area $|\mathcal{E}|$.

Theorem 2.1 (ETSP tour length, [12]): There exists $\beta > 0$ such that for every set $\mathcal{Q}$ of n points in $\mathcal{E}$, $\mathrm{ETSP}(\mathcal{Q}) \leq \beta\sqrt{n|\mathcal{E}|}$.

The problem of computing an optimal ETSP tour is *NP*-hard. However, there exist many efficient approximation algorithms such as the *Christofides' algorithm* [13].

b) Queueing Theory: Consider a queueing system with Poisson arrivals at rate λ, and a single server which provides bulk service. As customers arrive they form a queue and are served in batches. Every t_{B} seconds a batch is served containing either the first M customers in the queue, or the entire queue, whichever is smaller. In [14] the following result is established.

Theorem 2.2 (Mean waiting time, [14]): If $M > \lambda t_{\mathrm{B}}$, then the expected waiting time W satisfies

$$W \leq \frac{M-1}{\lambda} + \frac{t_{\mathrm{B}}}{2(M-\lambda t_{\mathrm{B}})}. \tag{1}$$

c) Vertex Coloring: An *undirected graph* $G = (V, E)$ consists of a set of vertices V and a set of edges $E \subset V \times V$. An edge $\{v, w\} \in E$ is incident to v and w, and v and w are neighbors. The *degree* of $v \in V$ is the number of edges incident to v. A *vertex-coloring* of G is a mapping $f : V \to \mathbb{N}$ with $f(v) \neq f(w)$ for all $\{v, w\} \in E$. The number $f(v)$ is the *color* of v. Finding the minimum vertex coloring is *NP*-hard, and no approximation algorithms exist. However, the following theorem gives an upper bound on the number of colors required.

Theorem 2.3 (Vertex coloring [15]): Let G be an undirected graph with n nodes and with maximum degree α. Then G has a vertex coloring with at most $\alpha+1$ colors, and such a coloring can be found in $O(n)$ computation time.

An $\alpha + 1$ coloring can be found as follows.

Greedy coloring heuristic of $G = (V, E)$.

1 Let $V = \{v_1, \ldots, v_n\}$.
2 **for** $i = 1$ *to* n **do**
3 Set $f(v_i)$ to the minimum color $k \in \mathbb{N}$ such that $k \neq f(v_j)$ for all neighboring vertices v_j, $j < i$.

III. Network and task model

a) Robot model: Consider n robotic agents contained in a square environment $\mathcal{E} \subset \mathbb{R}^2$. The position of agent $i \in \{1, \ldots, n\}$, is denoted by $\mathbf{p}^{[i]} \in \mathcal{E}$, and we assume the first order dynamics $\dot{\mathbf{p}}^{[i]} = \mathbf{u}^{[i]}$, where $\|\mathbf{u}^{[i]}\| \leq v_{\max}$ for some $v_{\max} > 0$. Each agent is capable of providing services (or resources) in the set $\mathcal{R} := \{r_1, \ldots, r_k\}$. For agent i, $\mathcal{C}^{[i]} : \mathcal{R} \to \{0, 1\}$ records its capabilities, i.e., agent i provides service r_j only if $\mathcal{C}^{[i]}(r_j) = 1$. We assume that computations are centralized, and leave the problem of decentralizing our policies to future work.

b) Task model: A task $\mathcal{T}_j$ is described by the tuple $\{\mathcal{Q}_j, R_j, \mathcal{L}_j, S_j\}$. The set $\mathcal{Q}_j \subset \mathcal{E}$ contains locations and the map $R_j : \mathcal{Q}_j \to 2^{\mathcal{R}}$ gives the services required at each location. We assume all services at a location must be provided simultaneously (if this is not the case, then the location will appear more than once in $\mathcal{Q}_j$). The set $\mathcal{L}_j$ contains rules, or logical connectives, which describe the task. Possible rules are: (i) A partial ordering on $\mathcal{Q}_j$. For $\mathbf{q}_1, \mathbf{q}_2 \in \mathcal{Q}_j$, we write $\mathbf{q}_1 \prec \mathbf{q}_2$ if $\mathbf{q}_1$ must be serviced before $\mathbf{q}_2$, and $\mathbf{q}_1 \doteq \mathbf{q}_2$ if the locations must be serviced simultaneously. (ii) An equivalence relation $\sim$ on $\mathcal{Q}_j$, where $\mathbf{q}_1 \sim \mathbf{q}_2$ implies that $\mathbf{q}_1$ and $\mathbf{q}_2$ must be serviced by the same set of agents. (iii) The temporal operator $\mathcal{U}$ (until). For example,the statement ($\mathbf{q}_2$ is unknown)$\mathcal{U}$($\mathbf{q}_1$ is serviced) implies that location $\mathbf{q}_2$ is unknown until service $R_j(\mathbf{q}_1)$ is provided at location $\mathbf{q}_1$.

Finally, the random variable $S_j : \mathcal{Q}_j \to \mathbb{R}_{\geq 0}$ gives the on-site service time required at each location. Using S_j we can also define the total on-site service time s_j of task $\mathcal{T}_j$. The partial ordering $\preccurlyeq$ partitions the set $\mathcal{Q}_j$ into disjoint subsets $Q_1, \ldots, Q_p$, containing simultaneous tasks. That is, if $\mathbf{q}_1 \in Q_1$, then $\mathbf{q}_2 \in Q_1$ if and only if $\mathbf{q}_1 \doteq \mathbf{q}_2$. Thus, the total on-site service time s_j is $s_j := \sum_{i=1}^{p} \max_{\mathbf{q} \in Q_i} S_j(\mathbf{q})$.

c) Task arrival model: We assume that tasks enter the environment according to a Poisson process with intensity λ. We define $\bar{s} := \lim_{j \to +\infty} \mathbb{E}[s_j]$ to be the expected total on-site service time for a task. Consider a policy P by which agents service tasks. This policy induces a control law $\mathbf{u}^{[i]}(t)$ for each agent. For policy P, let D_j denote the difference between the service completion time and the arrival time of task $\mathcal{T}_j$. This time consists of a waiting time W_j, and a service time s_j. Then, we let $D_P := \lim_{j \to +\infty} \mathbb{E}[D_j]$, denote the *expected delay*. Little's result [16] states that if D_P exists, then the expected number of tasks $\bar{N}_P$ in the environment under policy P, is given by $\bar{N}_P = \lambda D_P$. If n agents are servicing tasks arriving at rate λ, then a necessary condition for there to exist a stable policy is that $\lambda\bar{s}/n < 1$. That is, during an on-site service time $\bar{s}$, fewer than n tasks must arrive.

With the task arrival model described above, we define the following quantities: the *total throughput* is λ, and the *per-agent throughput* $T(n)$ is λ/n.We let $D^*(n)$ denote the optimal (least achievable) delay, and $T^*(n)$ the maximum achievable throughput (capacity).

IV. Analyzing throughput and delay

In this section we introduce scaling laws for studying the expected delay as a function of the per-agent throughput and look at two existing dynamic task allocation problems.

A. Scaling laws

We are interested in studying the expected task delay $D(n)$ as a function of the per-agent throughput $T(n)$. In particular we look at the case where the number of agents becomes large and the arrival rate λ scales (increases) with n. We assume that $\bar{s}$, the expected on-site service time of a task, remains constant. Also, as n increases, the environment must grow to accommodate the increase in agents. In [17]

it was shown that in order to maintain a reasonable safety distance between agents, the ratio $\sqrt{|\mathcal{E}|}/v_{\max}$ must scale with n as $\sqrt{n}$. In [18] this scaling was referred to as a *critical environment*. These scaling laws can be summarize as follows.

Definition 4.1 (Asymptotic regime): In the asymptotic regime (i) the number of agents $n \to +\infty$; (ii) $\bar{s}$ is independent of n; (iii) $|\mathcal{E}(n)|/(nv_{\max}^2(n)) \to \text{const}_2 \in \mathbb{R}_{>0}$.

B. Multiple dynamic traveling repairperson problem

In the multiple dynamic traveling repairperson problem (DTRP), there is a single service, $\mathcal{R} := \{r_1\}$, and $\mathcal{C}^{[i]}(r_1) = 1$ for each agent i. The tasks are of the form $\mathcal{T} := \{\mathbf{q}, , r_1, \mathcal{L}, s\}$, where an agent must visit $\mathbf{q}$, which is known upon task arrival (i.e., ($\mathbf{q}$ is unknown)$\mathcal{U}(\mathcal{T}$ arrives)), and provide service r_1 for time s. Tasks arrive according to a Poisson process with rate λ , and the location $\mathbf{q}$ is independently and uniformly distributed in $\mathcal{E}$. The on-site service times are independent with mean $\bar{s}$.

In [5], two lower bounds on the expected delay are presented. The first states that

$$D^* \geq \frac{1}{v_{\max}} \mathbb{E}\left[\min_{\mathbf{q}_0 \in \mathcal{D}^*} \|\mathbf{q} - \mathbf{q}_0\|\right] + \bar{s}, \tag{2}$$

where $\mathcal{D}^*$ is the set of n locations that minimizes the expected distance to the uniformly distributed location $\mathbf{q}$. The second bound states that there exists $\gamma > 0$ such that

$$D^* \geq \gamma^2 \frac{\lambda|\mathcal{E}|}{n^2 v_{\max}^2 (1-\rho)^2} - \frac{\bar{s}(1-2\rho)}{2\rho} =: D_{\text{DTRP}}(n), \tag{3}$$

where $\rho := \lambda\bar{s}/n$. In the asymptotic regime (2) becomes $D^*(n) \in \Omega(1)$, and (3) becomes $D^*(n) \in \Omega(T(n))$. Note that for stability $\lambda\bar{s}/n < 1$, and thus $T(n) < 1/\bar{s}$.

In [5] several policies are developed. When $T(n) \to 0^+$ as $n \to +\infty$, an optimal policy is to place the vehicles at locations $\mathcal{D}^*$ and service tasks first-come, first-served, by the closest vehicle, which returns to its location in $\mathcal{D}^*$ after each service is completed. When $T(n) \to \text{const} \in \mathbb{R}_{>0}$ as $n \to +\infty$, the TSP partitioning policy is developed.

The TSP partitioning policy

Optimize: over M.
1. Partition $\mathcal{E}$ into n regions and assign a vehicle to each region.
2. **foreach** region-vehicle pair **do**
3. As tasks arrive in the region, form sets of size M.
4. As sets are formed, deposit them in a queue.
5. Service the queue first-come, first-served, following an optimal TSP tour on each set of M tasks.

By combining the analysis in [5] , combined with Theorem 2.1 one can show that in the asymptotic regime, the delay of this TSP partitioning policy is in $O(\max\{T(n), 1\})$ when $T(n) < 1/\bar{s}$. Thus, we have the following.

Theorem 4.2 (DTRP delay, [5]): In the asymptotic regime, if $T(n) \to \text{const} < 1/\bar{s}$, then the optimal delay of the DTRP is in $\Theta(1)$. If $T(n) \to \text{const} > 1/\bar{s}$, then the optimal delay is infinite.

Thus, in the DTRP we can achieve a per-agent throughput of $\Theta(1)$, while incurring a delay of only $\Theta(1)$.

C. Dynamic pickup delivery problem

In the dynamic pickup delivery problem (DPDP) [8] there are two services, $\mathcal{R} := \{\texttt{pickup}, \texttt{deliver}\}$. Each agent has the capability of providing both services, and so for each agent i, $\mathcal{C}^{[i]}(\texttt{pickup}) = \mathcal{C}^{[i]}(\texttt{deliver}) = 1$. The tasks are of the form $\mathcal{T} := \{\{\mathbf{q}_1, \mathbf{q}_2\}, R, \mathcal{L}, S\}$, where the set of rules $\mathcal{L}$ indicates that $\mathbf{q}_1 \prec \mathbf{q}_2$, $\mathbf{q}_1 \sim \mathbf{q}_2$, and ($\mathbf{q}_1, \mathbf{q}_2$ are unknown)$\mathcal{U}(\mathcal{T}$ arrives).[1] The service requirements are $R(\mathbf{q}_1) = \texttt{pickup}$ and $R(\mathbf{q}_2) = \texttt{deliver}$. Thus, when a task arrives, a message must be picked up from a known source location $\mathbf{q}_1$ and delivered to known destination location $\mathbf{q}_2$ by the same agent. A fixed on-site service time of $s := S(\mathbf{q}_1) = S(\mathbf{q}_2)$ is incurred at each location. Tasks arrive according to a Poisson process with rate λ, and for each task, $\mathbf{q}_1$ and $\mathbf{q}_2$ are uniformly randomly distributed in $\mathcal{E}$. In [8] it is shown that for this problem

$$D^* \geq \max\left\{\frac{\gamma^2}{4}\frac{\lambda|\mathcal{E}|}{n^{3/2}v_{\max}^2(1-\rho)^2}, \frac{\sqrt{|\mathcal{E}|}}{2v_{\max}(1-\rho)}, s\right\}, \tag{4}$$

where $\rho := \lambda s/n$. In addition, a policy P is introduced which yields a delay D_P within a constant factor of the lower bound in (4). Thus, we have the following result.

Theorem 4.3 (DPDP delay, [8]): In the asymptotic regime, if $T(n) \to \text{const} < 1/(2s)$, then the optimal delay of the DPDP is in $\Theta(\sqrt{n})$. If $T(n) \to \text{const} > 1/(2s)$, then the optimal delay is infinite.

This result implies that a delay of order $\sqrt{n}$ must be incurred regardless of the per-agent throughput.

V. Dynamic team forming problem

We now introduce the *dynamic team forming problem* (DTFP) and present a lower bound on the optimal delay. In the DTFP there is a heterogeneous group of vehicles in which each vehicle provides one of k services. Tasks appear in the environment which require some subset of the k services. Thus, teams of agents must be formed in order to provide the services required for each task. This type of problem could arise in UAV surveillance [19] where the services represent waveforms for interrogation of a target/region, such as electro-optical, infra-red, synthetic aperture radar, foliage penetrating radar, and moving target indication radar.

A. Problem statement

In the DTFP, there is a set of services $\mathcal{R} := \{r_1, \ldots, r_k\}$. In addition, there are k different types of agents, and an agent of type $j \in \{1, \ldots, k\}$, can provide only service r_j. We assume that the total number of agents n satisfies $n/k \in \mathbb{N}$, and thus we say that for agent i, $\mathcal{C}^{[i]}(r_j) = 1$ only if $i(\text{mod}\, k) = j$. That is, agent i can provide only service $r_{i(\text{mod}\, k)}$. The task we consider is of the form $\mathcal{T} := \{\mathbf{q}, R, \mathcal{L}, S\}$, where $R(\mathbf{q}) \subset \mathcal{R}$, $\mathcal{L}$ dictates that ($\mathbf{q}, R(\mathbf{q})$ are unknown)$\mathcal{U}(\mathcal{T}$ arrives). Tasks arrive according to a Poisson process with rate λ, and the location $\mathbf{q}$ is independently and uniformly distributed in $\mathcal{E}$. For each task, the set $R(\mathbf{q})$ is independently and uniformly randomly selected from a set of subsets of $\mathcal{R}$ of cardinality $\mathcal{K} \leq 2^k - 1$ (at this time,

[1] The case where ($\mathbf{q}_2$ is unknown)$\mathcal{U}(\mathbf{q}_1$ is serviced) is also considered.

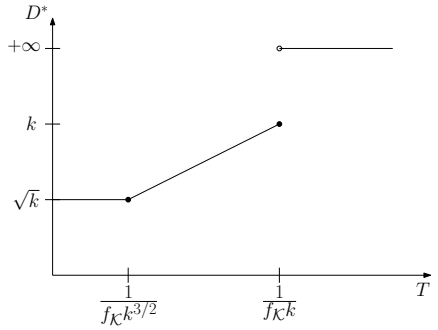

Fig. 1. Dynamic team forming lower bound: Delay versus throughput.

we leave the set of subsets unspecified). The on-site service time is independent of the number of services required for a task, has mean $\bar{s}$, and is upper bounded by $s_{\max} \in \mathbb{R}_{>0}$. Thus for a task with $R(\mathbf{q}) = \{r_1, r_4\}$, the task is completed when agents $1(\mathrm{mod}\, k)$ and $4(\mathrm{mod}\, k)$ simultaneously spend an on-site service time of $S(\mathbf{q}) = s$ at location $\mathbf{q}$.

B. Lower bound on optimal delay

In total there are $\mathcal{K}$ different task types. Let f_j denote the fraction of task types that require service r_j. In order to derive a lower bounds, and analyze proposed policies we make two simplifications. First, we assume that $f_1 = \cdots = f_k =: f_{\mathcal{K}}$. This implies that the required services are spread evenly over the task types, and thus, each service appears in $f_{\mathcal{K}}\mathcal{K}$ task types. Since the service set for each task is chosen uniformly and randomly, this also implies that $f_{\mathcal{K}}$ is the probability that a task requires service r_i. Notice that $1/k \leq f_{\mathcal{K}} \leq 1$. Second, we only consider *task-type unbiased* policies. These are policies for which the delay for each task type is equal, $\lim_{j\to+\infty} \mathbb{E}[D_j|\text{task } j \text{ is of type } R(\mathbf{q})] = D$, for each task type $R(\mathbf{q}) \in \mathcal{R}$.

With these assumptions we can lower bound the optimal delay. Note, that all parameters are potentially a function of n and we should be writing $k(n)$, $\mathcal{K}(n)$, $f_{\mathcal{K}}(n)$, and so on. However, to simplify the notation we omit the explicit dependence. For convenience, Table I contains a list of parameters and their definitions.

Theorem 5.1 (Optimal delay): In the asymptotic regime, if $kf_{\mathcal{K}}T(n) \to \text{const} < 1/\bar{s}$, then the optimal delay of the dynamic team forming problem is in $\Omega(\max\{f_{\mathcal{K}}k^2T(n), \sqrt{k}\})$. If $kf_{\mathcal{K}}T(n) \to \text{const} > 1/\bar{s}$, then the delay is infinite.

Fig. 1 shows the order of the optimal delay as a function of the per-agent throughput.

VI. Policies for dynamic team forming

A. Complete team policy

Here we propose a policy that has good performance when each service is required in a constant fraction of the tasks.

Complete team policy

1 Form n/k teams of k agents, where each team contains one agent of each type.
2 Have each team meet and move as a single entity.
3 As tasks arrive, service them by one of the n/k teams according to the TSP partitioning policy.

TABLE I

Parameters used in the dynamic team forming problem.

Parameter	Definition
T	expected per-agent throughput
D	expected task delay
k	number of different services
$\mathcal{K}$	number of different task types, $\leq 2^k - 1$
$f_{\mathcal{K}}$	fraction of tasks requiring an individual service
$\bar{s}$	expected on-site service time
$s_{\max}$	maximum on-site service time
L	number of time slots in service schedule
w	fractional length of service schedule, $L/\mathcal{K}$
b	maximum number of services required for a task

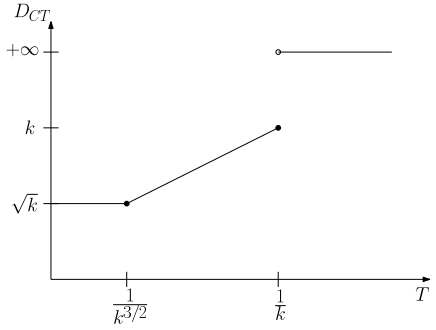

Fig. 2. Complete team policy: Delay versus throughput.

With this policy, the problem is simply a DTRP with n/k vehicles, and an arrival rate of λ. Hence, we have the following result.

Theorem 6.1 (Complete team delay): In the asymptotic regime, if $kT(n) \to \text{const} < 1/\bar{s}$, then the expected delay of the Complete team policy is $O(\max\{k^2T(n), \sqrt{k}\})$. If $kT(n) \to \text{const} > 1/\bar{s}$, then the delay is infinite.

Notice that if $f_{\mathcal{K}} \in \Omega(1)$, then the policy is within a constant factor of the optimal. Fig. 2 shows the order of the delay as a function of the per-agent throughput.

Thus, when each service is required in a constant fraction of the tasks, or when $k(n) \to \text{const} \in \mathbb{N}$ as $n \to +\infty$, the Complete team policy is within a constant factor of the optimal. However, in certain instances the above policy may be inefficient as each agent visits every task, not just the ones which require its service. This manifests itself as a limit on the per-agent throughput to $1/k$, independent of $f_{\mathcal{K}}$.

B. Task-specific team policy

Notice that there are n/k agents of each type, and each service appears in $f_{\mathcal{K}}\mathcal{K}$ service sets. Thus, if $f_{\mathcal{K}}\mathcal{K} \leq n/k$ there are enough agents of each type to create all $\mathcal{K}$ service sets. More specifically, we could create $N_{\text{copy}} := \lfloor n/(kf_{\mathcal{K}}\mathcal{K})\rfloor$ copies of each of the $\mathcal{K}$ service sets. Thus, when $f_{\mathcal{K}}\mathcal{K} \leq n/k$ we have the following policy.

Task-specific team policy

Assumes: $f_{\mathcal{K}}\mathcal{K} \leq n/k$.
1 For each of the $\mathcal{K}$ different service sets, create $N_{\text{copy}} := \lfloor n/(kf_{\mathcal{K}}\mathcal{K})\rfloor$ teams of agents, where the number of agents in each team is equal to the number of required services, and each agent provides a required service.
2 Service each task by one of its N_{copy} corresponding teams, according to the TSP partitioning policy.

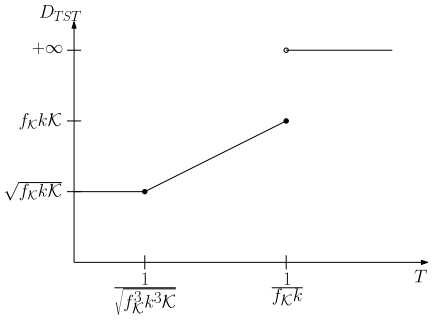

Fig. 3. Task-specific team policy: Delay versus throughput.

In the following theorem we characterize the delay of Task-specific team policy.

Theorem 6.2 (Task-specific policy delay): In the asymptotic regime, If $kf_{\mathcal{K}}T(n) \to \text{const} < 1/(2\bar{s})$, then the expected delay of the Task-specific policy is $O(\max\{f_{\mathcal{K}}^2k^2\mathcal{K}T(n), \sqrt{f_{\mathcal{K}}k\mathcal{K}}\})$. If $kf_{\mathcal{K}}T(n) \to \text{const} > 1/\bar{s}$, then the delay is infinite.

Fig. 3 shows the order of the delay as a function of the per-agent throughput for the Task-specific team policy.

C. Scheduled task-specific team policy

The Task-specific team policy can only be applied when $f_{\mathcal{K}}\mathcal{K} \leq n/k$. Here we propose a policy for all parameter values which divides the task types into several groups, and then runs the Task-specific policy on each group sequentially. We begin by defining a *service schedule*.

Definition 6.3 (Service schedule): A *service schedule* $\mathcal{S}$ is a partition of the $\mathcal{K}$ service sets into L time slots, such that each service set appears in exactly one time slot, and the service sets in each time slot are pairwise disjoint. The schedule has *length* L, and *fractional length* $w := L/\mathcal{K}$.

The following lemma lower bounds the length of a service schedule by using the fact that for each $i \in \{1, \ldots, k\}$, $f_{\mathcal{K}}\mathcal{K}$ contain service r_i.

Lemma 6.4 (Schedule length I): If $\mathcal{S}$ is a service schedule, then it contains at least $f_{\mathcal{K}}\mathcal{K}$ time slots (i.e., $w \geq f_{\mathcal{K}}$).

From Lemma 6.4, every service schedule must contain at least $f_{\mathcal{K}}\mathcal{K}$ slots. We now give a method for creating a schedule. Consider the graph consisting of $\mathcal{K}$ vertices, one for each service set, and edges connecting any two vertices whose service sets have a non-empty intersection. This is known as an intersection graph. A service schedule, is then simply a vertex coloring of this graph. From Section II the problem of determining the optimal (minimal) coloring is *NP*-hard. However, we can color the graph using the greedy heuristic in Section II. An example is shown in Fig. 4. Using Theorem 2.3 we arrive at the following result.

Lemma 6.5 (Schedule length II): If each task requires no more than $b \leq k$ services, then a service schedule with $w \leq \min\{bf_{\mathcal{K}}, 1\}$ can be found in $O(\mathcal{K})$ computation time.

We are now ready to present the Scheduled task-specific team policy.

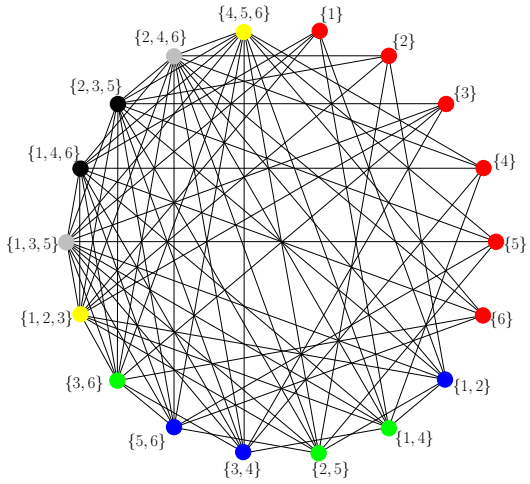

Fig. 4. Creating a service schedule using the greedy vertex coloring heuristic. In this figure, $k = 6$, $\mathcal{K} = 18$, $f_{\mathcal{K}} = 6/18$, and the resulting schedule has length $L = 6$.

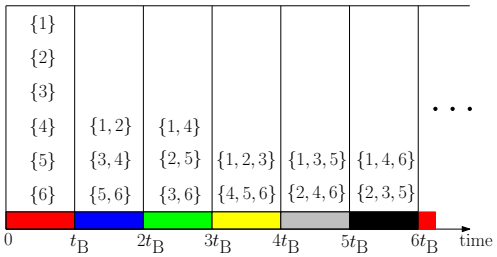

Fig. 5. Service schedule created by the coloring in Fig. 4. The task types serviced during each time slot are shown (e.g., in time slot $[t_{\mathrm{B}}, 2t_{\mathrm{B}}[$, agents $1 (\mathrm{mod}\, k)$ and $2 (\mathrm{mod}\, k)$ meet to service tasks with service set $\{1, 2\}$).

Scheduled task-specific team policy

Assumes: A service schedule with time slot duration t_{B}.
Optimize: over t_{B} and M.
Partition $\mathcal{E}$ into n/k regions and assign one agent of each type to each region.
foreach region **do**
  Form a queue for each of the $\mathcal{K}$ task types.
  foreach time slot in the schedule **do**
    Divide agents into teams to form required task types.
    For each team, service the first M tasks in the queue, or as many as can be served in time t_{B} (whichever comes first), by following an optimal TSP tour.
When the end of the service schedule is reached, repeat.

By applying the results on the Euclidean traveling salesperson tour and on batch queues in Section II, we are able to bound the delay of the Scheduled task-specific team policy.

Theorem 6.6 (Scheduled task-specific team delay): In the asymptotic regime, if $kwT(n) \to \text{const} < 1/s_{\max}$, then the expected delay of the Scheduled task-specific team policy is $O\left(\max\left\{w^2k^2\mathcal{K}T(n), w\mathcal{K}\sqrt{k}\right\}\right)$. If $kwT(n) \to \text{const} > 1/\bar{s}$, then the delay is infinite.

Fig. 6 shows delay as a function of the per-agent throughput for the Scheduled task-specific team policy.

Remark 6.7 (Comments on Theorem 6.6): When $bf_{\mathcal{K}} \leq 1$, the greedy vertex coloring scheme creates a service schedule with $w \leq bf_{\mathcal{K}}$. In this case, one can achieve

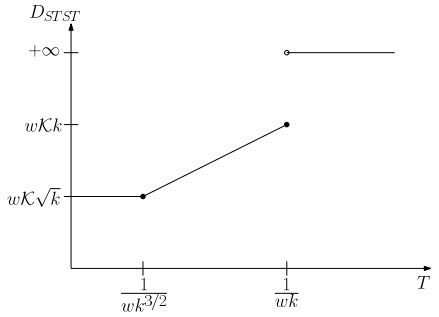

Fig. 6. Scheduled task-specific team policy: Delay versus throughput.

TABLE II
POLICY COMPARISON WITH $\mathcal{K} = bk$, AND $f_{\mathcal{K}} \in \Theta(b/k)$.

Policy	Capacity	Delay at capacity
Optimal	$\Theta(1/b)$	$\Omega(k)$
Complete team	$\Theta(1/k)$	$O(k)$
Task-specific team	$\Theta(1/b)$	$O(b^2k)$
Scheduled task-specific (greedy)	$\Theta(1/b^2)$	$O(b^3k)$
Scheduled task-specific (optimal)	$\Theta(1/b)$	$O(b^2k)$

a per-agent throughput of $\Theta(1/(bf_{\mathcal{K}}k))$, with a delay of $O(bf_{\mathcal{K}}k\mathcal{K})$. Thus, if b is small compared to k, the service schedule can provide a near optimal maximum per-agent throughput (i.e., capacity). However, the delay depends on the number of different task types, $\mathcal{K}$, and thus could be significantly larger than the optimal delay.

When the per-agent throughput is "high," the delay of the Task-specific team policy is $O(\max\{k^2 f_{\mathcal{K}}^2 \mathcal{K} T(n))$. In comparison, the delay of the Scheduled task-specific team policy is $O(\max\{w^2k^2\mathcal{K}T(n))$. By Lemma 6.4, $w \geq f_{\mathcal{K}}$, and thus the Task-specific policy performs at least as well as the Scheduled task-specific policy. However, we can only use the policy when $f_{\mathcal{K}}\mathcal{K} \leq n/k$. Also, the policy does not easily adapt to situations where new tasks types are added, and old task types are removed, since the entire partitioning of the agents into teams must be recalculated. •

D. Policy comparison

To compare the performance of the policies, consider the specific case where, for each $j \in \{1, \ldots, b\}$, with $b < k$, there are k service sets with cardinality j. That is, k task types require one service, k task types require two services, and so on. Thus, $\mathcal{K} = bk$. Further, assume that each individual service appears in j of the k service sets of cardinality j, for each $j \in \{1, \ldots, b\}$. From this, we obtain

$$f_{\mathcal{K}} = \frac{\sum_{j=1}^{b} j}{\mathcal{K}} = \frac{b(b+1)/2}{bk} \in \Theta(b/k).$$

An example of service sets satisfying these assumptions is shown in Fig. 4 and Fig. 5. Using these values, we can compare the maximum achievable throughput (or capacity) for each of the policies, and the delay at capacity. These results are summarized in Table II, where Scheduled task-specific team bounds assume $b^2 \leq k$. In Table II we see that if $b \in \Theta(k)$ when n is large, then the Complete team policy is within a constant factor of the optimal. However, if $b \in \Theta(1)$, then the per-agent throughput of the Complete team policy cannot be raised above $1/k$, whereas the Scheduled task-specific and Task-specific policies provide capacity, and delay, within a constant factor of the optimal.

VII. CONCLUSIONS

In this paper we presented a model for dynamic task allocation problems, and a framework within which they can be studied. We introduced the dynamic team forming problem, and proposed three team forming policies. There are many areas for future work. We would like to look into creating distributed versions of our policies, and extend our dynamic team forming analysis to nonuniform task type distributions, task-type biased policies, and the case where services are not evenly spread among task types.

REFERENCES

[1] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int J Robotic Research*, vol. 23, no. 9, pp. 939–954, 2004.

[2] M. F. Godwin, S. Spry, and J. K. Hedrick, "Distributed collaboration with limited communication using mission state estimates," in *Proc ACC*, Minneapolis, MN, Jun. 2006, pp. 2040–2046.

[3] M. Alighanbari and J. P. How, "Robust decentralized task assignment for cooperative UAVs," in *Proc AIAA GN&C*, Keystone, CO, Aug. 2006.

[4] D. J. Bertsimas and G. J. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, pp. 601–615, 1991.

[5] ——, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles," *Operations Research*, vol. 41, no. 1, pp. 60–76, 1993.

[6] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proc CDC*, Paradise Island, Bahamas, Dec. 2004, pp. 3357–3363.

[7] M. Pavone, E. Frazzoli, and F. Bullo, "Decentralized algorithms for stochastic and dynamic vehicle routing with general target distribution," in *Proc CDC*, New Orleans, LA, Dec. 2007, pp. 4869–4874.

[8] H. A. Waisanen, D. Shah, and M. A. Dahleh, "A dynamic pickup and delivery problem in mobile networks under information constraints," *IEEE Trans Automatic Ctrl*, Nov. 2006, accepted.

[9] ——, "Fundamental performance limits for multi-stage vehicle routing problems," *Operations Research*, Aug. 2007, submitted.

[10] G. Sharma and R. M. N. Shroff, "Delay and capacity trade-offs in mobile ad hoc networks: A global perspective," in *INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[11] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Optimal throughput-delay scaling in wireless networks. Part I: The fluid model," *IEEE Trans Inf. Theory*, vol. 52, no. 6, pp. 2568–2592, 2006.

[12] J. M. Steele, "Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space," *Mathematics of Operations Research*, vol. 15, no. 4, p. 749, 1990.

[13] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Carnegie-Mellon University, Pittsburgh, PA, Tech. Rep. 388, Apr. 1976.

[14] N. T. J. Bailey, "On queueing processes with bulk service," *Journal of the Royal Statistical Society. Series B*, vol. 16, no. 1, pp. 80–87, 1954.

[15] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 3rd ed., ser. Algorithmics and Combinatorics. New York, NY: Springer Verlag, 2005, no. 21.

[16] L. Kleinrock, *Queueing Systems. Volume I: Theory*. New York, NY: John Wiley and Sons, 1975.

[17] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris, "Transfer time complexity of conflict-free vehicle routing with no communications," *Int J Robotic Research*, vol. 26, no. 3, pp. 255–272, 2007.

[18] S. L. Smith and F. Bullo, "Target assignment for robotic networks: Worst-case and stochastic performance in dense environments," in *Proc CDC*, New Orleans, LA, Dec. 2007, pp. 3585–3590.

[19] E. K. P. Chong, C. M. Kreucher, and A. O. Hero III, "Adaptive sensing via partially observable Markov decision process approximations," *IEEE Proceedings*, 2007, submitted.