The Dynamic Team Forming Problem: Throughput and Delay for Unbiased Policies☆

Stephen L. Smith[*,a], Francesco Bullo[a]

[a] *Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106, USA*

Abstract

The dynamic team forming problem (DTFP) for a heterogeneous group of robots is described as follows. Each robot is capable of providing a specific service. Tasks arrive sequentially over time, assume random locations in the environment, and require several different services. A task is completed when a team of robots travels to the task location and provides the required services. The goal is to minimize the expected *delay* between a task's arrival and its completion. We restrict our attention to unbiased policies for the DTFP, i.e., policies for which the expected delay is the same for all tasks. We introduce three intuitive policies, and in certain asymptotic regimes we analyze their *delay* as a function of the arrival rate of tasks (or *throughput*). For each policy we show that there is a broad class of system parameters for which the policy's performance is within a constant factor of the optimal.

Key words: dynamic vehicle routing, team/coalition forming, robotic networks

---

1. Introduction

Consider a heterogeneous fleet of mobile robotic agents deployed in an environment $\mathcal{E} \subset \mathbb{R}^2$. Each robot is capable of providing one of k services. Tasks appear in the environment sequentially over time, assume a random location in $\mathcal{E}$, and require some subset of the k services. To complete a task, all required services must be present at the task location. Thus, for each task, a team of robots which can provide the required services must be formed, and must travel to the task location. The goal is to minimize the expected delay between a task's arrival and its completion. We refer to this problem as the *dynamic team forming problem* (DTFP). This problem arises, for example, in

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

*Corresponding Author. Tel: (+1) 805 893-2801, Fax: (+1) 805 893-8651

Email addresses: stephen@engineering.ucsb.edu (Stephen L. Smith), bullo@engineering.ucsb.edu (Francesco Bullo)

UAV surveillance [1] 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.

The DTFP is a dynamic vehicle routing problem [2] since tasks arrive sequentially over time. A special case of the DTFP is the dynamic traveling repairperson problem (DTRP) [3, 4]. In this problem 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 [5] and [6]. Another dynamic vehicle routing problem is the dynamic pickup delivery problem (DPDP) [7] where each task consists of a source-destination pair. A message must be picked up from the source, and delivered to the destination. 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 both problems the expected delay increases with the task arrival rate. This trade-off is well known in ad hoc wireless networks [8, 9]; 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.

The contributions of this paper are the following. First, we introduce the novel *dynamic team forming problem*. Second, we propose three policies for dynamic team forming; the *complete team policy* where teams are formed that consist of all possible services; the *task-specific team policy* where teams are formed for each type of task; and the *scheduled task-specific team policy* where each type of task is serviced by a task-specific team, but only during certain intervals of time, as defined by a schedule. All three policies utilize Euclidean traveling salesperson tours to compute optimal routes through sets of tasks. Third, by making some assumptions on the system parameters (e.g., on relative number of robots, task-type frequency, etc.), we 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 derive a lower bound on the expected delay of the DTFP, upper bounds for the delay each of the three policies, and show that for certain classes of system parameters each policy performs within a constant factor of the optimal.

2. Background Material

Here we review results on the Euclidean traveling salesperson problem, environment partitioning, queueing theory, and vertex coloring. We let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{N}$ denote the set of real numbers, positive real numbers, and positive integers, respectively. For $A \subset \mathbb{R}^2$ 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 write $f(n) \in \Theta(g)$.

The Euclidean traveling salesperson problem (ETSP): For a set $\mathcal{Q}$ of n points in $\mathbb{R}^2$, let ETSP($\mathcal{Q}$) denote the length of the shortest closed path through all points in $\mathcal{Q}$. The following result characterizes this length when the point set $\mathcal{Q}$ lies inside a square environment $\mathcal{E}$ with area $|\mathcal{E}|$.

Theorem 2.1 (ETSP tour length, [10])**.** *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 efficient approximation algorithms such as the *Christofides' algorithm* [11].

Partitioning an environment: The following definition formalizes the idea of partitioning a square environment $\mathcal{E} \subset \mathbb{R}^2$ into n regions, such that each region is "approximately" a square of area $|\mathcal{E}|/n$.

Definition 2.2 (c-square partition)**.** *A partition of $\mathcal{E}$ into n regions is* c-square *if each region can be contained in a square of area $c|\mathcal{E}|/n$.*

One can easily create a 4-square partition by 1) gridding $\mathcal{E}$ into $\lceil\sqrt{n}\rceil^2$ squares, 2) selecting $\lceil\sqrt{n}\rceil^2 - n$ pairs of edge adjacent squares, such that no square appears in more than one pair, and 3) fusing each pair into a single region.

Queueing theory: Consider a queueing system with Poisson arrivals at rate λ, and a single server providing bulk service. As customers arrive they form a queue and are served in batches. Every t_{batch} seconds a batch is served containing either the first M customers in the queue, or the entire queue, whichever is smaller. In [12] the following result is established.

Theorem 2.3 (Expected waiting time, [12])**.** *If $M > \lambda t_{\mathrm{batch}}$, then the expected waiting time W satisfies*

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

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. The vertex-coloring problem is to find the vertex coloring $f : V \to \mathbb{N}$ which minimizes the number of required colors; that is, which minimizes $\max_{v \in V} f(v)$. The problem is *NP*-hard, and no approximation algorithms exist. However, the following theorem gives an upper bound on the number of colors required.

Theorem 2.4 (Vertex coloring, [13])**.** *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 using the Greedy coloring heuristic.*

Greedy coloring heuristic of $G = (\{v_1, \ldots, v_n\}, E)$.

```
for i = 1 to n do
    Set f(v_i) to the minimum color k ∈ ℕ such that k ≠ f(v_j) for all
    neighboring vertices v_j, j < i.
```

3. The Dynamic Team Forming Problem (DTFP)

Robot model: Consider n robotic agents contained in a square environment $\mathcal{E} \subset \mathbb{R}^2$. Each robot has first order dynamics with speed bounded by $v_{\max} > 0$. Each robot is capable of providing one of k services. We assume there are $n_j > 0$ robots capable of providing service j (called robots of service-type j), for each $j \in \{1, \ldots, k\}$, and thus

$$n := \sum_{j=1}^{k} n_j.$$

Task model: There are $\mathcal{K}$ different types of tasks. Tasks of type $\alpha \in \{1, \ldots, \mathcal{K}\}$ arrive in the environment over time according to a Poisson process with rate λ_α. Upon arrival each task assumes an independent and identically distributed (i.i.d.) location uniformly in $\mathcal{E}$. Each task-type $\alpha \in \{1, \ldots, \mathcal{K}\}$ requires a subset of the k services. It will be useful to record the required services in a zero-one vector $R_\alpha \in \{0, 1\}^k$. The jth entry of R_α is 1 if service j is required for task-type α, and 0 otherwise. The on-site service time for a task of type α is an i.i.d. random variable with mean $\bar{s}_\alpha$ and finite variance. To complete a task of type α, a team of robots capable of providing the required services must travel to the task location and remain there for the on-site service time. Note that the total task arrival rate is

$$\lambda := \sum_{\alpha=1}^{\mathcal{K}} \lambda_\alpha.$$

Performance metric: We define a control policy for the group of robots as a map P which assigns a commanded velocity to each robot as a function of the current state of the system; that is, as a function of the current robot positions, and state of all unserviced tasks.[1] For a given policy P, let $D^i_{\alpha,P}$ denote the difference between the service completion time and the arrival time of the ith task of type α. Let $D_{\alpha,P} := \limsup_{i\to+\infty} \mathbb{E}[D^i_{\alpha,P}]$ denote the limiting *expected delay* of task-type α under policy P. Then, the DTFP is to find policies which minimize some cost function of the delays. In particular we consider two cost

[1] We assume that computations are centralized, and leave the problem of decentralizing our policies to future work.

functions; the worst-case delay, and the average delay:

$$\min_P \max_\alpha D_{\alpha,P} \quad \text{or} \quad \min_P \sum_\alpha D_{\alpha,P}.$$

As in the classical queueing literature [14], we can give a necessary condition for the existence of a stabilizing policy (i.e., a policy for which the limiting expected delays are all finite). First, we define the matrix

$$R := [R_1 \ \cdots \ R_{\mathcal{K}}] \in \{0,1\}^{k \times \mathcal{K}}. \tag{2}$$

Then, a necessary condition for stability is that

$$R[\lambda_1 \bar{s}_1 \ \cdots \ \lambda_{\mathcal{K}} \bar{s}_{\mathcal{K}}]^T < [n_1 \ \cdots \ n_k]^T \tag{3}$$

component-wise. The jth inequality in equation (3) states that the fraction of time a robot of type j is busy performing on-site service must be less than 1 for any stable policy.

Dynamic traveling repairperson problem (DTRP): The DTRP is a special case of the dynamic team forming problem in which there is only one service (i.e., $k = 1$), and thus only one task-type. When $k = 1$ equation (3) becomes $\lambda \bar{s} < n$, or $\rho := \lambda \bar{s}/n < 1$. In queueing theory [14], the quantity ρ is known as the load factor. In [4], two lower bounds on the optimal expected delay D^* are presented which will be useful in the upcoming analysis. First,

$$D^* \geq \frac{1}{v_{\max}} \mathbb{E}\left[\min_{\mathbf{p} \in \{\mathbf{p}_1, \dots, \mathbf{p}_n\}} \|\mathbf{q} - \mathbf{p}\| \right] + \bar{s}, \tag{4}$$

where $\mathbf{p}_1, \dots, \mathbf{p}_n$ are the n locations which minimize the expected distance to a uniformly distributed location $\mathbf{q}$. Second, there exists a $\gamma > 0$ such that

$$D^* \geq \gamma^2 \frac{\lambda |\mathcal{E}|}{n^2 v_{\max}^2 (1-\rho)^2} - \frac{\bar{s}(1-2\rho)}{2\rho} =: D_{\mathrm{DTRP}}(n, \lambda). \tag{5}$$

Several policies are developed in [4]. A policy which we will utilize in this paper is the ETSP partitioning policy. We slightly alter the policy and use the c-square partition in Definition 2.2.

The ETSP partitioning policy

Optimize: over task set-size M.

Partition $\mathcal{E}$ into n approximately square regions and assign one robot to each region.

foreach region-robot pair **do**

- As tasks arrive in the region, form sets of size M.
- As sets are formed, deposit them in a queue.
- Service the queue first-come, first-served, following an optimal ETSP tour on each set of M tasks.

From [4], the optimal value of M is $C\lambda^2|\mathcal{E}|/(n^2 v_{\max}^2(1-\rho)^2)$, for some $C>0$. While we will utilize the ETSP partitioning policy in what follows, it should be noted that we could equivalently utilize the receding-horizon policy in [15].

4. Dynamic Team Forming Policies

In what follows we will consider *task-type unbiased* policies; policies P for which the delay of each task is equal and thus

$$D_{1,P} = D_{2,P} = \cdots = D_{\mathcal{K},P}.$$

For task-type unbiased policies, the worst-case delay and average delay problems are equivalent. Policies of this type are amenable to analysis because the task-type unbiased constraint collapses the feasible set of delays from a subset of $\mathbb{R}^{\mathcal{K}}$ to a subset of $\mathbb{R}$. Because of this, we can simply talk about the delay of a policy P as D_P, and the least achievable delay as D^*.

We will now introduce three policies for the dynamic team forming problem. Their performance will be analyzed in Section 5.

Policy 1 (complete team): We begin by proposing a policy which essentially turns the problem into a dynamic traveling repairperson problem.

Policy 1: Complete team

1. Form $N_{\text{CT}} := \min\{n_1, \dots, n_k\}$ teams of k robots, where each team contains one robot of each service-type.
2. Have each team meet and move as a single entity.
3. As tasks arrive, service them by one of the N_{CT} teams according to the ETSP partitioning policy.

Policy 2 (task-specific team): Recalling that $R := [R_1, \dots, R_{\mathcal{K}}] \in \mathbb{R}^{k\times\mathcal{K}}$, the vector $R\mathbf{1}_{\mathcal{K}}$ records in its jth entry the number of task-types that require service j, where $\mathbf{1}_{\mathcal{K}}$ is a $\mathcal{K}\times 1$ vector of ones. Thus, if

$$R\mathbf{1}_{\mathcal{K}} \leq [n_1, \dots, n_k]^T \tag{6}$$

component-wise, then there are enough robots of each service-type to create a dedicated team for every task-type. More specifically, we could create

$$N_{\text{TST}} := \left\lfloor \min\left\{ \frac{n_j}{e_j^T R\mathbf{1}_{\mathcal{K}}} \,\middle|\, j \in \{1, \dots, k\} \right\} \right\rfloor$$

teams for each task-type, where e_j is the jth vector of the standard basis of $\mathbb{R}^k$. Thus, when equation (6) is satisfied, we have the following policy.

Policy 2: Task-specific team

Assumes: Equation (6) is satisfied.

For each of the $\mathcal{K}$ task-types, create N_{TST} teams of robots, where there is one robot in the team for each service required by the task-type.

Service each task by one of its N_{TST} corresponding teams, according to the ETSP partitioning policy.

Policy 3 (scheduled task-specific team):

The task-specific team policy can be applied only when equation (6) is satisfied; that is, when there is a sufficient number of robots of each service-type. Here we propose a policy which requires only a single robot of each service type. The policy partitions the task-types into groups, where each group is chosen such that there is a sufficient number of robots to create a dedicated team for each task-type in the group. The task-specific team policy is then run on each group sequentially. We begin by introducing a *service schedule* which defines the partition of task-types into groups.

Definition 4.1 (Service schedule). *A service schedule $\mathcal{S}$ is a partition of the $\mathcal{K}$ task-types into L time slots, such that each task-type appears in precisely one time slot, and the task-types in each time slot are pairwise disjoint (i.e., In a given time slot, each service appears in at most one task-type).*

With the definition of a service schedule we can present the third policy.

Policy 3: Scheduled task-specific team

Assumes: A service schedule with time slot duration t_{B}.
Optimize: over time slot duration t_{B} and task set-size M.

```
Partition ℰ into N_CT := min{n_1, ..., n_k} approximately square regions and
assign one robot of each service-type to each region.
foreach region do
    Form a queue for each of the 𝒦 task-types.
    foreach time slot in the schedule do
        For each task-type in the time slot, create a team containing one robot
        for each required service.
        For each team, service the first M tasks in the corresponding queue, or
        as many as can be served in time t_B, by following an optimal ETSP
        tour.
When the end of the service schedule is reached, repeat.
```

5. Analysis of the Dynamic Team Forming Problem

In this section we present simplifying assumptions, scaling laws, and the canonical throughput-delay profile which will allow us to study the asymptotic expected delay of the three team forming policies. We derive a lower bound on the achievable delay (independent of policy), and upper bounds for the delay of each policy.

5.1. *Simplifying assumptions and asymptotic regime*

To analyze the performance of the three policies we assume the following:

(A1) There are n/k robots of each service-type (i.e., $n_j = n/k$ for each $j \in \{1, \ldots, k\}$).
(A2) The arrival rate is $\lambda/\mathcal{K}$ for each task-type, (i.e., $\lambda_\alpha = \lambda/\mathcal{K}$ for each task $\alpha \in \{1, \ldots, \mathcal{K}\}$).
(A3) The on-site service time has mean $\bar{s}$ and is upper bounded by $s_{\max}$ for all task-types (i.e., $\bar{s}_\alpha = \bar{s}$ for each task-type $\alpha \in \{1, \ldots, \mathcal{K}\}$).
(A4) There exists $p \in [1/k, 1]$ such that for each $j \in \{1, \ldots, k\}$, the service j appears in $p\mathcal{K}$ of the $\mathcal{K}$ task-types. Thus, each task will require service j with probability p.

With these assumptions, the stability condition in equation (3) simplifies to

$$\frac{\lambda}{n} < \frac{1}{pk\bar{s}}. \tag{7}$$

For a stable policy P we say that λ is the *total throughput* of the system (i.e., the total number of tasks served per unit time), and $T_n := \lambda/n$ is the *per-robot throughput* (we use the notation T_n to remind the reader that the throughput depends on the number of robots n).

We are interested in studying the expected delay of each task-type as a function of the per-robot throughput T_n. In particular, in the next sections we study the performance as the number of robots n becomes large. As n increases, if the density of robots is to remain constant, then the environment must grow. In fact, the ratio $\sqrt{|\mathcal{E}|}/v_{\max}$ must scale as $\sqrt{n}$, [16]. In [17] this scaling is referred to as a *critical environment*. Thus we will study the performance in the following regime.

Definition 5.1 (Asymptotic regime)**.** *In the asymptotic regime (i) the number of robots $n \to +\infty$; (ii) on-site service times are independent of n; (iii) $|\mathcal{E}(n)|/(nv_{\max}^2(n)) \to \text{constant} > 0$.*

5.2. *Canonical throughput-delay profile*

In what follows we will characterize the way in which the delay varies with the per-robot throughput T_n. We will see that there is a canonical throughput-delay profile $f_{D_{\min},D_{\text{ord}},T_{\text{crit}}} : \mathbb{R}_{>0} \to \mathbb{R}_{>0} \cup \{+\infty\}$ which has the form

$$T_n \mapsto \begin{cases} \max\left\{D_{\min}, \dfrac{D_{\text{ord}}(T_n/T_{\text{crit}})}{(1 - T_n/T_{\text{crit}})^2}\right\}, & \text{if } T_n < T_{\text{crit}}, \\ +\infty, & \text{if } T_n \geq T_{\text{crit}}. \end{cases} \tag{8}$$

This profile is described by the three positive parameters $D_{\min}$, D_{ord} and T_{crit}, where $D_{\text{ord}} \geq D_{\min}$. These parameters have the following interpretation:

- $D_{\min}$ is the minimum achievable delay for any positive throughput.

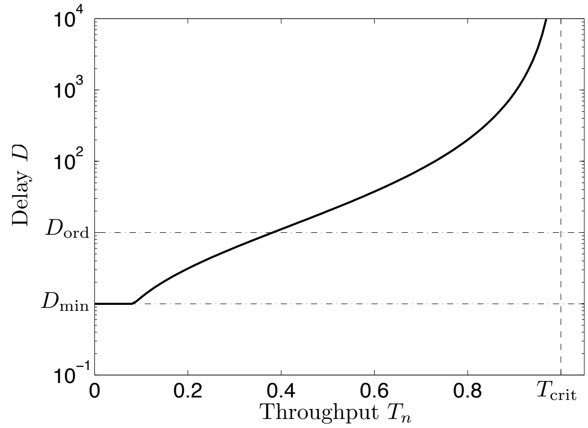

Figure 1: The canonical throughput-delay profile for the dynamic team forming problem. The semi-log plot is for parameter values of $D_{\min} = 1$, $D_{\mathrm{ord}} = 10$, and $T_{\mathrm{crit}} = 1$. If $T_n \geq T_{\mathrm{crit}}$, then the delay is $+\infty$.

- T_{crit} is the maximum achievable throughput (or capacity).
- D_{ord} is the delay when operating at $(3-\sqrt{5})/2 \approx 0.38$ of capacity T_{crit}. Additionally, D_{ord} captures the order of the delay when operating at a constant fraction of capacity.

An example of the throughput-delay profile with parameters $D_{\min} = 1$, $D_{\mathrm{ord}} = 10$, and $T_{\mathrm{crit}} = 1$ is shown in Figure 1 on a semi-log graph. In what follows we will use these three parameters to compare the performance of our policies.

5.3. Lower bound on the achievable delay

We now lower bound the achievable delay D^* for task-type unbiased policies. Note that all parameters are potentially a function of n. However, to simplify the notation we omit the explicit dependence. For convenience, Table 1 contains all parameters and their definitions.

Table 1: Parameters used in the dynamic team forming problem.

Parameter	**Definition**
k	number of different services
$\mathcal{K}$	number of different task-types
p	fraction of tasks requiring an individual service
$\bar{s}$, $s_{\max}$	expected and maximum on-site service time
L	number of time slots in service schedule
b	maximum number of services required for a task

Theorem 5.2 (Optimal delay)**.** *In the asymptotic regime, the optimal delay of the DTFP as a function of the per-robot throughput T_n is in $\Omega\big(f_{D_{\min},D_{\mathrm{ord}},T_{\mathrm{crit}}}(T_n)\big)$, where*

$$D_{\min} = \sqrt{k}, \quad D_{\mathrm{ord}} = k, \quad T_{\mathrm{crit}} = \frac{1}{pk\bar{s}}.$$

Proof. By assumption (A4), service $j \in \{1, \dots, k\}$ is required in $p\mathcal{K}$ of the $\mathcal{K}$ task-types. By assumption (A2), the arrival rate of tasks requiring service j is $p\lambda$. By assumption (A1), $n_j = n/k$ robots can provide service j. Thus, we can use the results on the DTRP to lower bound the achievable delay of n/k robots servicing tasks arriving at rate $p\lambda$. That is, for every policy P we have

$$\sum_{\text{tasks } \alpha \text{ requiring service } j} \frac{\lambda_\alpha}{p\lambda} D_{\alpha,P} \geq D_{\mathrm{DTRP}}(n/k, p\lambda). \tag{9}$$

By assumption (A2), $\lambda_\alpha = \lambda/\mathcal{K}$ for each $\alpha \in \{1, \dots, \mathcal{K}\}$, and by restricting our attention to task-type unbiased policies, $D_{\alpha,P} = D_P$ for each $\alpha \in \{1, \dots, \mathcal{K}\}$. Applying the bound in equation (5), we can write equation (9) as

$$D^*(n) \geq D_{\mathrm{DTRP}}(n/k, p\lambda) \in \Omega\left(\frac{p\lambda|\mathcal{E}|}{(n/k)^2 v_{\max}^2 (1 - pk\bar{s}\lambda/n)^2}\right),$$

In the asymptotic regime the above equation becomes

$$D^*(n) \in \Omega\left(\frac{pk^2 T_n}{(1 - pk\bar{s}T_n)^2}\right).$$

In addition, in the asymptotic regime, equation (4) yields $D^*(n) \in \Omega(\sqrt{k})$. Combining the two results we obtain a lower bound of $\Omega(f_{D_{\min},D_{\mathrm{ord}},T_{\mathrm{crit}}}(T_n))$, where $f_{D_{\min},D_{\mathrm{ord}},T_{\mathrm{crit}}}$ is the canonical throughput-delay profile defined in equation (8), and the parameters are $D_{\min} = \sqrt{k}$, $D_{\mathrm{ord}} = k$ and $T_{\mathrm{crit}} = 1/(pk\bar{s})$. □

5.4. Upper bounds on the policy throughput-delay

In this section we characterize the performance of each policy in terms of the canonical throughput-delay profile of equation (8).

Policy 1: Complete team

The complete team policy is simply the ETSP partitioning policy with n/k robots and with arrival rate λ. In the limiting regime as $\rho \to 1^-$, the performance of this policy [4] is within a known constant factor of the lower bound in equation (5). The proof in [4] utilizes the following facts: as $\rho \to 1^-$, the number of unserviced tasks M tends to $+\infty$, and for a set $\mathcal{Q}$ of M i.i.d. uniform points in a square environment of area $|\mathcal{E}|$

$$\lim_{M \to +\infty} \frac{\mathrm{ETSP}(\mathcal{Q})}{\sqrt{M}} = \beta_{\mathrm{ETSP}} \sqrt{|\mathcal{E}|}, \tag{10}$$

for some constant $\beta_{\text{ETSP}} > 0$. Following the same proof as in [4] but replacing equation (10) with the result in Theorem 2.1 (which is not as tight, but holds for all values of M), and using the fact that the partition is approximately square, one can prove that the delay of the ETSP partitioning policy with n robots and with arrival rate λ is in

$$O\left(\max\left\{\frac{1}{v_{\max}}\sqrt{\frac{|\mathcal{E}|}{n}}, \frac{\lambda|\mathcal{E}|}{n^2 v_{\max}^2(1-\lambda\bar{s}/n)^2}\right\}\right), \tag{11}$$

in the asymptotic regime. Combining equation (11) with the throughput-delay profile in equation (8) we obtain the following result.

Theorem 5.3 (Complete team delay)**.** *In the asymptotic regime, the expected delay of the complete team policy as a function of the per-robot throughput T_n is in $O\big(f_{D_{\min},D_{\text{ord}},T_{\text{crit}}}(T_n)\big)$, where*

$$D_{\min} = \sqrt{k}, \quad D_{\text{ord}} = k, \quad T_{\text{crit}} = \frac{1}{k\bar{s}}.$$

Notice that if $p \sim 1$ (i.e., each service is required in a constant fraction of the tasks), then the policy is within a constant factor of the optimal. However, in certain instances policy 1 may be inefficient as each robot visits every task, not just the ones which require its service. This inefficiency appears as a limit on the per-robot throughput of $1/k$, independent of p.

Remark 5.4 (Dynamic traveling repairperson delay)**.** *In the DTRP we have $k = p = \mathcal{K} = 1$, and thus combining Theorems 5.2 and 5.3 we see that the expected delay in the asymptotic regime is in $\Theta\big(f_{D_{\min},D_{\text{ord}},T_{\text{crit}}}(T_n)\big)$, where $D_{\min} = D_{\text{ord}} = 1$, and $T_{\text{crit}} = 1/\bar{s}$.* •

Policy 2: Task-specific team

With assumptions (A1)-(A4), the necessary condition on the number of robots required for this policy, given in equation (6), becomes $p\mathcal{K} \leq n/k$, and thus $N_{\text{TST}} := \lfloor n/(kp\mathcal{K})\rfloor$. In the following theorem we characterize the delay of the task-specific team policy.

Theorem 5.5 (Task-specific team delay)**.** *In the asymptotic regime, if $p\mathcal{K} \leq n/k$, then the expected delay of the task-specific team policy as a function of the per-robot throughput T_n is in $O\big(f_{D_{\min},D_{\text{ord}},T_{\text{crit}}}(T_n)\big)$ where*

$$D_{\min} = \sqrt{pk\mathcal{K}}, \quad D_{\text{ord}} = pk\mathcal{K}, \quad T_{\text{crit}} = \frac{1}{C\bar{s}pk},$$

and $C \in [1, 2[$ is defined as $C = n/(kp\mathcal{K}N_{\text{TST}})$.

Proof. The arrival rate for each task-type is $\bar{\lambda} = \lambda/\mathcal{K}$ (by assumption (A2)), and the number of teams that provide service to each task-type is N_{TST}. Since

$N_{\text{TST}} \geq 1$, and $N_{\text{TST}} \leq n/(kp\mathcal{K})$, we can define

$$C := \frac{n}{kp\mathcal{K}N_{\text{TST}}} \in [1, 2[.$$

From the ETSP partitioning policy result in equation (11), the delay is

$$\begin{aligned} D_{\text{TST}}(n) &\in O\left(\max\left\{\frac{1}{v_{\max}}\sqrt{\frac{|\mathcal{E}|}{N_{\text{TST}}}}, \frac{\bar{\lambda}|\mathcal{E}|}{N_{\text{TST}}^2 v_{\max}^2 (1-\bar{\lambda}\bar{s}/N_{\text{TST}})^2}\right\}\right) \\ &= O\left(\sqrt{pk\mathcal{K}}, \frac{p^2k^2\mathcal{K}T_n}{(1-C\bar{s}pkT_n)^2}\right). \end{aligned}$$

Letting $D_{\min} = \sqrt{pk\mathcal{K}}$, $D_{\text{ord}} = pk\mathcal{K}$ and $T_{\text{crit}} = 1/(2\bar{s}pk)$ we obtain the desired result. □

From this analysis we see that the task-specific team policy can achieve near optimal throughput. However, it requires that there are a sufficient number of robots. The following policy requires only a single robot of each service-type.

Policy 3: Scheduled task-specific team

The following theorem bounds the delay of policy 3.

Theorem 5.6 (Scheduled task-specific team delay)**.** *In the asymptotic regime, the expected delay of the scheduled task-specific team policy as a function of the per-robot throughput T_n is in $O\big(f_{D_{\min},D_{\text{ord}},T_{\text{crit}}}(T_n)\big)$ where*

$$D_{\min} = L\sqrt{k}, \quad D_{\text{ord}} = Lk, \quad T_{\text{crit}} = \frac{\mathcal{K}}{\nu s_{\max} Lk},$$

for any fixed $\nu > 1$.

Proof. Consider a service schedule with length L and time slot duration t_{B}. In each of the n/k regions (assumption (A1)), each task-type has arrival rate $\bar{\lambda} := \lambda k/(\mathcal{K}n)$ (assumption (A2)), and the queue for that task-type is serviced for t_{B} seconds every Lt_{B} seconds. (Notice that for stability we require that $t_{\text{B}} \geq \bar{s}\bar{\lambda}Lt_{\text{B}}$, which implies that the per-robot throughput must satisfy $T_n < \mathcal{K}/(L\bar{s}k)$.) Since each region can be contained in a square of area $c|\mathcal{E}|k/n$, where $c \leq 4$, we can use Theorem 2.1 to upper bound the amount of time required to service M tasks by $(2\beta/v_{\max})\sqrt{M|\mathcal{E}|k/n} + s_{\max}M$. Using the fact that there exists $C \in \mathbb{R}_{>0}$ such that $\sqrt{|\mathcal{E}|}/v_{\max} \geq C\sqrt{n}$, and redefining $\beta := 2C\beta$, the upper bound becomes

$$\beta\sqrt{Mk} + s_{\max}M. \tag{12}$$

Now, fix $\epsilon > 0$, and let us set $M := \eta\bar{\lambda}(Lt_{\text{B}})$, where $\eta \geq 1+\epsilon$ is the smallest number such that $M \in \mathbb{N}$. With this value of M we are guaranteed to service more tasks in time slot t_{B} than are expected to arrive in time Lt_{B}. Let us now consider two cases: $M = 1$, and $M > 1$. If $M = 1$, then in order to service one task in time t_{B}, we require from equation (12) that

$$t_{\mathrm{B}} \geq \beta\sqrt{k} + s_{\max}. \tag{13}$$

In the other case, when $M = \eta\bar{\lambda}(Lt_{\mathrm{B}}) \in \mathbb{N} \setminus \{1\}$, in order to service M tasks in time t_{B}, we require from equation (12) that

$$t_{\mathrm{B}} \geq \sqrt{\eta}\beta\sqrt{\bar{\lambda}Lt_{\mathrm{B}}k} + \eta s_{\max}\bar{\lambda}Lt_{\mathrm{B}}.$$

If $\eta s_{\max}\bar{\lambda}L < 1$, or equivalently $M \leq t_{\mathrm{B}}/s_{\max}$, then the previous condition can be rewritten as

$$t_{\mathrm{B}} \geq \frac{\eta\beta^2\bar{\lambda}Lk}{(1 - \eta s_{\max}\bar{\lambda}L)^2}. \tag{14}$$

The condition for t_{B} to be finite (i.e., $\eta s_{\max}\bar{\lambda}L < 1$) depends on $\eta \geq (1+\epsilon)$, which is not desirable since the exact value of η is implicitly defined. However, notice that as $\eta s_{\max}\bar{\lambda}L \to 1^-$, we have $t_{\mathrm{B}} \to +\infty$, and thus $M = \eta\bar{\lambda}(Lt_{\mathrm{B}}) \to +\infty$. This implies that as $\eta s_{\max}\bar{\lambda}L \to 1^-$, we have $(1+\epsilon)/\eta \to 1^-$. Thus, we can replace denominator of equation (14) by a constant times $(1 - (1+\epsilon)s_{\max}\bar{\lambda}L)^2$. Making this replacement and substituting $\bar{\lambda} = kT_n/\mathcal{K}$, we obtain

$$t_{\mathrm{B}} \geq \frac{\beta Lk^2T_n}{\mathcal{K}(1 - (1+\epsilon)s_{\max}LkT_n/\mathcal{K})^2}, \tag{15}$$

where the constant β has been redefined.

Let us now examine the queue for a particular task-type and compute the expected delay. In this queue, tasks arrive at a rate $\bar{\lambda}$, and every Lt_{B} seconds, M are served (i.e., $t_{\mathrm{batch}} = Lt_{\mathrm{B}}$). Thus, from equation (1), the expected time W that a task spends waiting in the queue is

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

If $M = 1$, then we easily obtain that $W \in O(Lt_{\mathrm{B}})$.

On the other hand, if $M > 1$, then

$$\begin{aligned} W &\leq \frac{\eta\bar{\lambda}Lt_{\mathrm{B}} - 1}{2\bar{\lambda}} + \frac{Lt_{\mathrm{B}}}{2(\eta\bar{\lambda}Lt_{\mathrm{B}} - \bar{\lambda}Lt_{\mathrm{B}})} \\ &\leq \frac{\eta Lt_{\mathrm{B}}}{2} + \frac{1}{2(\eta-1)\bar{\lambda}}. \end{aligned}$$

Noticing that $M > 1$ implies that $1/\bar{\lambda} \leq \eta Lt_{\mathrm{B}}$, we again obtain that $W \in O(Lt_{\mathrm{B}})$. The expected delay for a task to be completed is $D_{\mathrm{STST}}(n) \leq W + t_{\mathrm{B}} \in O(Lt_{\mathrm{B}})$. Choosing t_{B} to be the smallest value that satisfies both equations (13) and (15), we can upper bound $D_{\mathrm{STST}}(n)$ by the canonical throughput-delay profile, where $D_{\min} = L\sqrt{k}$, $D_{\mathrm{ord}} = L\mathcal{K}$, and $T_{\mathrm{crit}} = \mathcal{K}/\big((1+\epsilon)s_{\max}Lk\big)$, for any positive constant ϵ. □

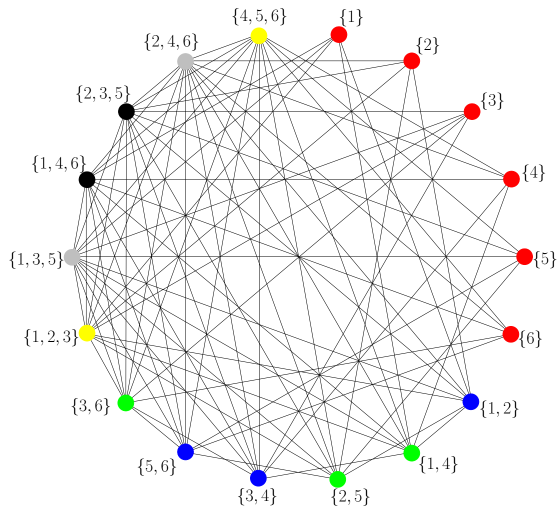

Figure 2: Creating a service schedule using the greedy vertex coloring heuristic. In this figure, $k = 6$, $\mathcal{K} = 18$, $p = 1/3$, and the resulting schedule has length $L = 6$.

Next, we will describe a method for creating a service schedule, and bound the schedule length L. The following lemma, lower bounding L, follows from assumption (A4).

Lemma 5.7 (Schedule length I)**.** *If $\mathcal{S}$ is a service schedule, then it contains at least $p\mathcal{K}$ time slots. (i.e., $L \geq p\mathcal{K}$).*

From Lemma 5.7, every service schedule must contain at least $p\mathcal{K}$ slots. We now give a method for creating a schedule. Consider the graph consisting of $\mathcal{K}$ vertices, one for each task-type, and edges connecting any two vertices that contain a common service. This is known as an intersection graph [18]. A service schedule is then simply a vertex coloring of this graph. From Section 2, the problem of determining the optimal (minimal) coloring is *NP*-hard. However, we can color the graph using the greedy heuristic in Section 2. An example is shown in Figure 2. Using Theorem 2.4 we arrive at the following result.

Lemma 5.8 (Schedule length II)**.** *If each task requires no more than $b \leq k$ services, then a service schedule with $L \leq \mathcal{K}\min\{bp, 1\}$ can be found in $O(\mathcal{K})$ computation time.*

5.5. Policy comparison

We have shown that the lower bound and the three policies all have delay profiles of the form

$$D(n) \sim \max\left\{D_{\min}, \frac{D_{\text{ord}}(T_n/T_{\text{crit}})}{(1 - T_n/T_{\text{crit}})^2}\right\}.$$

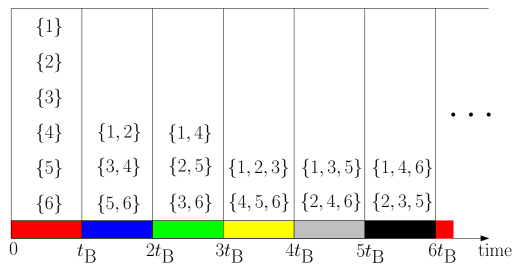

Figure 3: Service schedule created by the coloring in Figure 2. The task-types serviced during each time slot are shown (e.g., in time slot $[t_B, 2t_B[$, robots of service-type 1 and 2 meet to service tasks with task-type $\{1, 2\}$).

Table 2: A comparison the canonical throughput-delay parameters for the three policies. Two entries for the scheduled task-specific policy are shown depending on the value of $p \in [1/k, 1]$. Only the order of the capacity is shown, with the constant omitted.

	$D_{\min}$	D_{ord}	Capacity T_{crit}
Lower bound	$\sqrt{k}$	k	$1/(pk)$
Policy 1: Complete team	$\sqrt{k}$	k	$1/k$
Policy 2: Task-specific team	$\sqrt{pk\mathcal{K}}$	$pk\mathcal{K}$	$1/(pk)$
Policy 3: Scheduled task-specific ($p \sim \frac{1}{k}$)	$p\sqrt{k}\mathcal{K}$	$pk\mathcal{K}$	$1/(pk)$
Policy 3: Scheduled task-specific ($p \sim 1$)	$\sqrt{k}\mathcal{K}$	$k\mathcal{K}$	$1/k$

The parameters $D_{\min}$, D_{ord}, and T_{crit} are summarized for the lower bound and each of the three policies in Table 2. From these results, we can make several conclusions. First, if the throughput is very low, then Policy 1 has an expected delay of $\Theta(\sqrt{k})$, which is within a constant factor of the optimal. In addition, if $p \sim 1$ and each task requires nearly every service, then Policy 1 is within a constant factor of the optimal in terms of capacity and delay. Second, if $p \sim 1/k$ and each task requires few services, then the capacity of Policy 1 is sub-optimal, and the capacity of both Policies 2 and 3 are within a constant factor of optimal. However, the delay of Policies 2 and 3 may be much higher than the lower bound when the number of task-types $\mathcal{K}$ is very large. Third, Policy 2 performs at least as well as Policy 3, both in terms of capacity and delay. Thus, one should use Policy 2 if there are a sufficient number of robots of each service-type. However, if $p \sim 1/k$ and if resources are limited such that Policy 2 cannot be used, then Policy 3 should be used to maximize capacity.

From this discussion we see that the policies are complementary, and have large parameter regimes for which their performance, either in terms of capacity or delay, is within a constant factor of the optimal.

6. Conclusions

In this paper we introduced the novel dynamic team forming problem for robotic networks. We proposed three policies for team forming and characterized their performance in certain asymptotic regimes. There are many areas for future work. We would like to relax or remove some of the simplifying assumptions in Section 5.1. Also, we would like to look into creating distributed versions of our policies, and extending our analysis to task-type biased policies.

References

[1] E. K. P. Chong, C. M. Kreucher, A. O. Hero III, Monte-Carlo-based partially observable Markov decision process approximations for adaptive sensing, in: Int. Workshop on Discrete Event Systems, Göteborg, Sweden, 2008, pp. 173–180.

[2] W. B. Powell, P. Jaillet, A. Odoni, Stochastic and dynamic networks and routing, in: M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser (Eds.), Handbooks in Operations Research and Management Science, 8: Network Routing, Elsevier, 1995, pp. 141–295.

[3] D. J. Bertsimas, G. J. van Ryzin, A stochastic and dynamic vehicle routing problem in the Euclidean plane, Operations Research 39 (1991) 601–615.

[4] D. J. Bertsimas, G. J. van Ryzin, Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles, Operations Research 41 (1) (1993) 60–76.

[5] E. Frazzoli, F. Bullo, Decentralized algorithms for vehicle routing in a stochastic time-varying environment, in: IEEE Conf. on Decision and Control, Paradise Island, Bahamas, 2004, pp. 3357–3363.
URL `http://motion.me.ucsb.edu/pdf/2003r-fb.pdf`

[6] M. Pavone, E. Frazzoli, F. Bullo, Decentralized algorithms for stochastic and dynamic vehicle routing with general target distribution, in: IEEE Conf. on Decision and Control, New Orleans, LA, 2007, pp. 4869–4874.
URL `http://motion.me.ucsb.edu/pdf/2007g-pfb.pdf`

[7] H. A. Waisanen, D. Shah, M. A. Dahleh, A dynamic pickup and delivery problem in mobile networks under information constraints, IEEE Transactions on Automatic Control 53 (6) (2008) 1419–1433.

[8] G. Sharma, R. Mazumdar, N. Shroff, Delay and capacity trade-offs in mobile ad hoc networks: A global perspective, in: IEEE Conf. on Computer Communications, Barcelona, Spain, 2006, pp. 1–12.

[9] A. E. Gamal, J. Mammen, B. Prabhakar, D. Shah, Optimal throughput-delay scaling in wireless networks. Part I: The fluid model, IEEE Transactions on Information Theory 52 (6) (2006) 2568–2592.

[10] L. Few, The shortest path and the shortest road through n points in a region, Mathematika 2 (1955) 141–144.

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

[12] N. T. J. Bailey, On queueing processes with bulk service, Journal of the Royal Statistical Society. Series B 16 (1) (1954) 80–87.

[13] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, 3rd Edition, Vol. 21 of Algorithmics and Combinatorics, Springer, 2005.

[14] L. Kleinrock, Queueing Systems. Volume I: Theory, John Wiley, 1975.

[15] J. D. Papastavrou, A stochastic and dynamic routing policy using branching processes with state dependent immigration, European Journal of Operational Research 95 (1996) 167–177.

[16] V. Sharma, M. Savchenko, E. Frazzoli, P. Voulgaris, Transfer time complexity of conflict-free vehicle routing with no communications, International Journal of Robotics Research 26 (3) (2007) 255–272.

[17] S. L. Smith, F. Bullo, Monotonic target assignment for robotic networks, IEEE Transactions on Automatic Control 54 (10), (Submitted June 2007) to appear.
URL `http://motion.me.ucsb.edu/pdf/2007l-sb.pdf`

[18] T. A. McKee, F. R. McMorris, Topics in Intersection Graph Theory, Vol. 2 of Monographs on Discrete Mathematics and Applications, SIAM, 1999.