Lower Bounds for Dynamic Distributed Task Allocation
[Extended Abstract]

Hsin-Hao Su
Boston College

Nicole Wein*
MIT

Abstract

We study the problem of distributed task allocation in multi-agent systems e.g. the division of labor in an ant colony or robot swarm. Suppose there is a collection of agents, a collection of tasks, and a *demand vector*, which specifies the number of agents required to perform each task. The goal of the agents is to collectively allocate themselves to the tasks to satisfy the demand vector. We study the *dynamic* version of the problem where the demand vector changes over time. Here, the goal is to minimize the *switching cost*, which is the number of agents that change tasks in response to a change in the demand vector. The switching cost is an important metric since changing tasks may incur significant overhead.

We study a mathematical formalization of the above problem introduced by Su, Su, Dornhaus, and Lynch [21]. We prove the first non-trivial lower bounds for the switching cost.

*nwein@mit.edu, supported by an NSF Graduate Fellowship and NSF Grant CCF-1514339

1 Background

Task allocation in multi-agent systems is a fundamental problem in distributed computing, and is especially pertinent to *biological* distributed algorithms such as the division of labor in insect colonies. Given a collection of tasks, a collection of task-performing agents, and a *demand vector* which specifies the number of agents required to perform each task, the agents must collectively allocate themselves to the tasks to satisfy the demand vector. This problem has been studied in a wide variety of settings. For example, agents may be identical or have differing abilities, agents may or may not be permitted to communicate with each other, agents may have limited memory or computational power, agents may be faulty, and agents may or may not have full information about the demand vector. See Georgiou and Shvartsman's book [8] for a survey of the distributed task allocation literature. See also the more recent line of work by Dornhaus, Lynch and others on algorithms for task allocation in ant colonies [5, 21, 6, 18].

We consider the setting where the demand vector *changes dynamically* over time and agents must redistribute themselves among the tasks accordingly. We aim to minimize the *switching cost*, which is the number of agents that change tasks in response to a change in the demand vector. The switching cost is an important metric since changing tasks may incur significant overhead. Dynamic task allocation has been extensively studied in practical, heuristic, and experimental domains. For example, in insect biology it has been empirically observed that demands for tasks in ant colonies change over time based on environmental factors such as climate, season, food availability, and predation pressure [16]. Accordingly, there is a large body of biological work on developing hypotheses about how insects collectively perform task allocation in response to a changing environment (see surveys [1, 19]). Additionally, in swarm robotics, there is much experimental work on heuristics for dynamic task allocation (see e.g. [11, 20, 14, 15, 12, 13]).

Despite the rich experimental literature, to the best of our knowledge there are only two works on dynamic distributed task allocation from a theoretical algorithmic perspective. Su, Su, Dornhaus, and Lynch [21] present and analyze gossip-based algorithms for dynamic task allocation in ant colonies. Radeva, Dornhaus, Lynch, Nagpal, and Su [18] analyze dynamic task allocation in ant colonies when the ants behave randomly and have limited information about the demand vector.

2 Our setting

Before formally defining the problem statement, we outline the setting that we consider:

Objective: Our goal is to minimize the *switching cost*, defined as the number of agents that change tasks in response to a change in the demand vector.

Properties of agents:

1. the agents have complete information about the changing demand vector
2. the agents are heterogeneous
3. the agents cannot communicate
4. the agents are memoryless

The first two properties specify *capabilities* of the agents while the third and fourth properties specify *restrictions* on the agents. Although the exclusion of communication and memory may appear overly restrictive, our setting captures well-studied models of both collective insect behavior and swarm robotics:

Our setting in collective insect behavior There are a number of hypotheses that attempt to explain the mechanism behind task allocation in ant colonies (see the survey [1]). One such hypothesis is the *response threshold model*, in which ants decide which task to perform based on individual preferences and environmental factors. Specifically, the model postulates that there is an environmental stimulus associated with each task, and each individual ant has an internal threshold for each task, whereby if the stimulus exceeds the threshold, then the ant performs that task. The response threshold model was introduced in the 70s and has been studied extensively since (for comprehensive background on this model see the survey [1] and the introduction of [7]). Our setting captures the response threshold model since agents are permitted to behave based on individual preferences (property 2: agents are heterogeneous) and environmental factors (property 1: agents have complete information about the demand vector). We study whether models like response threshold model can achieve low switching costs.

Our setting in swarm robotics Inspired by insect behavior, researchers have also studied the response threshold model (which our setting captures) in the context of swarm robotics [2, 10, 23].

More generally, there is a body of work in swarm robotics specifically concerned with property 3 of our setting: eliminating communication (e.g. [22, 3, 9, 17]). In practice, communication may be unfeasible, costly, or unlikely. In particular, it may be unfeasible to build a fast and reliable network infrastructure capable of dealing with delays and failures, especially in a remote location.

Regarding property 4 of our setting, it is desirable for agents to not rely on memory while performing task allocation because if robots arrive to the site at different times e.g. if a robot fails and is subsequently replaced, a newly arriving robot should ideally be able to determine what task to work on simply by observing the environment.

Concretely, our setting for dynamic task allocation in swarm robotics may be applicable to disaster containment (e.g. for forest fires [17] or a oil spills [24]), agricultural foraging, mining, drone package delivery, and environmental monitoring [20].

3 Problem Statement

Formally, the problem is defined as follows. There are three parameters: n is the number of agents, k is the number of tasks, and s is the target *switching cost*, which we define later. The goal is to define a set of n deterministic functions $f_1^{n,k}, f_2^{n,k}, \ldots, f_n^{n,k}$, one for each agent, with the below properties. The function associated with each agent determines which task that agent is assigned to given the current demand vector.

- **Input:** For each agent a, the function $f_a^{n,k}$ takes as input a *demand vector* $\vec{d} = \{d_1, d_2, \ldots, d_k\}$ where each d_i is a non-negative integer and $\sum_i d_i = n$. Each d_i is the number of agents required for task i, and the total number of agents required for tasks is exactly the total number of agents.

- **Output:** For each agent a, the function $f_a^{n,k}$ outputs some $i \in [k]$. The output of $f_a^{n,k}(\vec{d})$ is the task that agent a performs when the demand vector is $\vec{d}$.

- **Demand satisfied:** For all demand vectors $\vec{d}$ and all tasks i, we require that the number of agents a for which $f_a^{n,k}(\vec{d}) = i$ is exactly d_i. That is, the allocation of agents to tasks defined by the set of functions $f_1^{n,k}, f_2^{n,k}, \ldots, f_n^{n,k}$ exactly satisfies the demand vector.

- **Switching cost satisfied:** The *switching cost* of a pair of demand vectors $\vec{d}, \vec{d'}$ is defined as the number of agents a for which $f_a^{n,k}(\vec{d}) \neq f_a^{n,k}(\vec{d'})$; that is, the number of agents that switch tasks if the demand vector changes from $\vec{d}$ to $\vec{d'}$ (or from $\vec{d'}$ to $\vec{d}$). We say that a pair of demand vectors $\vec{d}$, $\vec{d'}$ are *adjacent* if $|\vec{d} - \vec{d'}|_1 = 2$; that is, if we can get from $\vec{d}$ to $\vec{d'}$ by moving exactly one unit of demand from one task to another. The *maximum switching cost* of a set of functions $f_1^{n,k}, f_2^{n,k}, \ldots, f_n^{n,k}$ is defined as the maximum switching cost over all pairs of adjacent demand vectors; that is, the maximum number of agents that switch tasks in response to the movement of a single unit of demand from one task to another. We require that the maximum switching cost of $f_1^{n,k}, f_2^{n,k}, \ldots, f_n^{n,k}$ is at most s.

We are interested in the question of for which values of the parameters n, k, and s, there exists a set of functions $f_1^{n,k}, \ldots, f_n^{n,k}$ that satisfies the above properties. In particular, we aim to minimize the maximum switching cost:

Question. *Given n and k, what is the minimum possible maximum switching cost achievable over all sets of functions $f_1^{n,k}, \ldots, f_n^{n,k}$?*

3.1 Remarks

Remark 1. The problem statement only considers the switching cost of pairs of *adjacent* demand vectors. We note that this also implies a bound on the switching cost of non-adjacent vectors: if every pair of adjacent demand vectors has switching cost at most s, then every pair of demand vectors with ℓ_1 distance D has switching cost at most $sD/2$ (D is always divisible by 2 since for every demand vector $\sum_i d_i = n$).

Remark 2. We note that the problem statement is consistent with our setting described in section 2. In particular, the agents have complete information about the changing demand vector because for each agent, the function $f_a^{n,k}$ takes as input the current demand vector. The agents are heterogeneous because each agent a has an separate function $f_a^{n,k}$. The agents cannot base their decision for which task to perform on communication or memory because the *only* input to each function $f_a^{n,k}$ is the current demand vector.

Remark 3. Forbidding communication among agents is crucial in the formulation of the problem, as otherwise the problem would be trivial. In particular, it would always be possible to achieve maximum switching cost 1: when the current demand vector changes to an adjacent demand vector, the agents simply reach consensus about which single agent will move. As such, one technical challenge of this problem is that there are exponentially many ways to move demand over time to reach the present demand vector, however the allocation of agents must be the same regardless of how the present demand vector was reached.

Remark 4. We recognize that the problem statement does not reflect the literal behavior of insect colonies, as it is unreasonable to assume that agents can observe and satisfy an *exact* demand vector deterministically. However, the problem statement captures the core combinatorial structure of dynamic distributed task allocation.

4 Past Work

The problem statement was formulated by Su, Su, Dornhaus, and Lynch [21], who presented two upper bounds and a lower bound.

The first upper bound is a very simple set of functions $f_1^{n,k}, \ldots, f_n^{n,k}$ with maximum switching cost $k - 1$. Each agent has a unique ID in $[n]$ and the tasks are numbered from 1 to k. The functions are defined so that for all demand vectors, the agents populate the tasks in order from 1 to k in order of increasing agent ID. That is, for each agent a, $f_a^{n,k}$ is defined as the task j such that $\sum_{i=0}^{j-1} d_i < \mathrm{ID}(a)$ and $\sum_{i=0}^{j} d_i \geq \mathrm{ID}(a)$. Now, starting with any demand vector, if one unit of demand is moved from task i to task j, then at most one agent from each task numbered between i and j (including i but not including j) shifts to a new task. There are at most $k-1$ such tasks so the maximum switching cost is $k-1$.

The lower bound of Su et al. is also very simple. It shows that there does not exist a set of functions with maximum switching cost 1 for $n \geq 2$ and $k \geq 3$. Suppose for contradiction that there exists a set of functions with maximum switching cost 1 for $n = 2$ and $k = 3$ (the argument can be easily generalized to higher n and k). Suppose the current demand vector is $[1, 1, 0]$, that is, one agent is required for each of tasks 1 and 2 while no agent is required for task 3. Suppose agents a and b are assigned to tasks 1 and 2, respectively, which we denote $[a, b, \emptyset]$. Now suppose the demand vector changes from $[1, 1, 0]$ to the adjacent demand vector $[1, 0, 1]$. Since the maximum switching cost is 1, only one agent moves, so agent a remains at task 1 and agent b moves to task 3, so we have $[a, \emptyset, b]$. Now suppose the demand vector changes from $[1, 0, 1]$ to the adjacent demand vector $[0, 1, 1]$. Again, since the maximum switching cost is 1, from $[a, \emptyset, b]$ agent a moves from task 1 to task 2 resulting in $[\emptyset, a, b]$. Now suppose the demand vector changes from $[0, 1, 1]$ to the adjacent demand vector $[1, 1, 0]$, which was the initial demand vector. Since the maximum switching cost is 1, from $[\emptyset, a, b]$ agent b moves from task 3 to task 1 resulting in $[b, a, \emptyset]$. The problem statement requires that the allocation of agents depends *only* on the current demand vector, so the allocation of agents for any given demand vector must be the same *regardless* of the history of changes to the demand vector. However, we have shown that the allocation of agents for $[1, 1, 0]$ was initially $[a, b, \emptyset]$ and is now $[b, a, \emptyset]$, a contradiction. Thus, the maximum switching cost is at least 2.

The second upper bound of Su et al. states that there exists a set of functions with maximum switching cost 2 if $n \leq 6$ and $k = 4$. They prove this result by exhaustively listing all 84 demand vectors along with the allocation of agents for each vector.

5 Our results

Su et al. left the problem open of whether the maximum switching cost can always be upper bounded by 2, regardless of the number of tasks. Our results show that it is not true and provide further evidence that the maximum switching cost grows with the number of tasks.

In particular, one might expect that the limitations on n and k in the second upper bound of Su et al. is due to the fact the space of demand vectors grows exponentially with n and k so the method of proof by exhaustive list becomes unfeasible. However, our first result is that the upper bound of Su et al. is actually *tight*. In particular, achieving maximum switching cost 2 is impossible even if the number of tasks is increased by 1.

Theorem 5.1. *For $n \geq 3$, $k \geq 5$, every set of functions $f_1^{n,k}, \ldots, f_n^{n,k}$ has maximum switching cost at least 3.*

We then consider the next natural question: *For what values of n and k is it possible to achieve maximum switching cost 3?* Our second result is that that maximum switching cost 3 is not always possible:

Theorem 5.2. *There exist n and k such that every set of functions $f_1^{n,k}, \ldots, f_n^{n,k}$ has maximum switching cost at least 4.*

The value of k for Theorem 5.2 is an extremely large constant expressed as a tower function and derived from hypergraph Ramsey numbers [4].

References

[1] Samuel N Beshers and Jennifer H Fewell. Models of division of labor in social insects. *Annual review of entomology*, 46(1):413–440, 2001.

[2] Eduardo Castello, Tomoyuki Yamamoto, Yutaka Nakamura, and Hiroshi Ishiguro. Task allocation for a robotic swarm based on an adaptive response threshold model. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 259–266. IEEE, 2013.

[3] Jianing Chen. *Cooperation in Swarms of Robots without Communication.* PhD thesis, University of Sheffield, 2015.

[4] David Conlon, Jacob Fox, and Benny Sudakov. Hypergraph ramsey numbers. *Journal of the American Mathematical Society*, 23(1):247–266, 2010.

[5] Alejandro Cornejo, Anna Dornhaus, Nancy Lynch, and Radhika Nagpal. Task allocation in ant colonies. In *International Symposium on Distributed Computing*, pages 46–60. Springer, 2014.

[6] Anna Dornhaus, Nancy Lynch, Frederik Mallmann-Trenn, Dominik Pajak, and Tsvetomira Radeva. Self-stabilizing task allocation in spite of noise. *arXiv preprint arXiv:1805.03691*, 2018.

[7] Ana Duarte, Ido Pen, Laurent Keller, and Franz J Weissing. Evolution of self-organized division of labor in a response threshold model. *Behavioral ecology and sociobiology*, 66(6):947–957, 2012.

[8] Chryssis Georgiou and Alexander A Shvartsman. Cooperative task-oriented computing: Algorithms and complexity. *Synthesis Lectures on Distributed Computing Theory*, 2(2):1–167, 2011.

[9] Serge Kernbach, Dagmar Häbe, Olga Kernbach, Ronald Thenius, Gerald Radspieler, Toshifumi Kimura, and Thomas Schmickl. Adaptive collective decision-making in limited robot swarms without communication. *The International Journal of Robotics Research*, 32(1):35–55, 2013.

[10] Min-Hyuk Kim, Hyeoncheol Baik, and Seokcheon Lee. Response threshold model based uav search planning and task allocation. *Journal of Intelligent & Robotic Systems*, 75(3-4):625–640, 2014.

[11] Michael JB Krieger, Jean-Bernard Billeter, and Laurent Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992, 2000.

[12] Kristina Lerman, Chris Jones, Aram Galstyan, and Maja J Matarić. Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotics Research*, 25(3):225–241, 2006.

[13] Kathryn Sarah Macarthur, Ruben Stranders, Sarvapali D Ramchurn, and Nicholas R Jennings. A distributed anytime algorithm for dynamic task allocation in multi-agent systems. In *AAAI*, pages 701–706, 2011.

[14] James McLurkin and Daniel Yamins. Dynamic task assignment in robot swarms. In *Robotics: Science and Systems*, volume 8. Citeseer, 2005.

[15] James Dwight McLurkin. *Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots.* PhD thesis, Massachusetts Institute of Technology, 2004.

[16] George F Oster and Edward O Wilson. *Caste and ecology in the social insects.* Princeton University Press, 1979.

[17] Jacques Penders, Lyuba Alboul, Ulf Witkowski, Amir Naghsh, Joan Saez-Pons, Stefan Herbrechtsmeier, and Mohamed El-Habbal. A robot swarm assisting a human fire-fighter. *Advanced Robotics*, 25(1-2):93–117, 2011.

[18] Tsvetomira Radeva, Anna Dornhaus, Nancy Lynch, Radhika Nagpal, and Hsin-Hao Su. Costs of task allocation with local feedback: Effects of colony size and extra workers in social insects and other multi-agent systems. *PLoS computational biology*, 13(12):e1005904, 2017.

[19] Gene E Robinson. Regulation of division of labor in insect societies. *Annual review of entomology*, 37(1):637–665, 1992.

[20] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer, 2004.

[21] Hsin-Hao Su, Lili Su, Anna Dornhaus, and Nancy Lynch. Ant-inspired dynamic task allocation via gossiping. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 157–171. Springer, 2017.

[22] Zijian Wang and Mac Schwager. Multi-robot manipulation with no communication using only local measurements. In *CDC*, pages 380–385, 2015.

[23] Yongming Yang, Changjiu Zhou, and Yantao Tian. Swarm robots task allocation based on response threshold model. In *2009 4th International Conference on Autonomous Robots and Agents*, pages 171–176. IEEE, 2009.

[24] Emaad Mohamed H Zahugi, Mohamed M Shanta, and TV Prasad. Oil spill cleaning up using swarm of robots. In *Advances in Computing and Information Technology*, pages 215–224. Springer, 2013.