Multirobot Charging Strategies: A Game-Theoretic Approach

Tianshuang Gao and Sourabh Bhattacharya

***Abstract*—This letter considers the problem of assigning multiple robots to charging stations in order to minimize the total time required by all robots for the charging operation. We first show that the centralized problem is NP-hard. Then, we formulate the charging problem as a non-cooperative game. We propose an algorithm to obtain the pure strategy Nash equilibrium of the non-cooperative game, and show its uniqueness. We investigate the price of anarchy of this equilibrium as a function of the number of robots and stations. Next, we leverage our analysis on static charging stations to propose strategies for reducing the total cost when the charging stations are mobile. Finally, we analyze the performance of the strategies proposed for the charging stations through extensive simulation.**

***Index Terms*—Planning, scheduling and coordination, multirobot systems.**

I. Introduction

IN THE last decade, multi-robot systems have been extensively deployed in civilian as well as in military applications [1]. With the rapid advancement in sensing and computing technology, the range of tasks performed by these robots has significantly increased. Multiple robots can perform tasks more efficiently than a single robot, and in some cases, accomplish tasks that cannot be completed by a single robot. Moreover, distributed sensing and decision-making increase the robustness of multirobot systems to failure compared to single robots. Coordination of multiple robots to accomplish tasks remains an active area of research [2]. However, coordination requires a reliable communication channel between robots to exchange information. Establishing such a communication network in open environments over a wide area is a challenge due to the sheer cost and maintenance involved. In this letter, we explore a "communication-free" technique for multirobot systems to complete tasks. Specifically, we investigate non-cooperative charging strategies for a team of mobile robots.

Manuscript received February 25, 2019; accepted June 2, 2019. Date of publication June 7, 2019; date of current version June 19, 2019. This letter was recommended for publication by Associate Editor F. Ju and Editor J. Li upon evaluation of the reviewers' comments. This work was supported by USDA/NIFA National Robotics Initiative under Grant 2017-67021-25965. *(Corresponding author: Sourabh Bhattacharya.)*

T. Gao is with the Department Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: tsgao@iastate.edu).

S. Bhattacharya is with the Department of Mechanical Engineering, and the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: sbhattac@iastate.edu).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2019.2921695

For persistent operations, robots rely on uninterrupted power supply. When the on-board power drops below a certain threshold, the robot needs to abort its task and move towards a charging station to get charged. Recharging facilities, for example, rechargeable batteries and self-recharge device or station are often deployed in order to provide power to the robots in the field. Current technology involves automated docking and battery swap systems [3], [4] for recharging robots. The time spent for the re-charging task, including traveling time and queuing time, depends on the location of the charging stations relative to the robots. In [5], the authors propose an algorithm that finds the minimum number of charging stations and their locations so that the robots can always reach a charging station from any location in the workspace without traveling more than a pre-specified number of steps. However, the queuing process is not considered when multiple robots are trying to recharge at a station. In a centralized system, a central authority can make a plan for the individual robots since the decision of the entire group has to be taken into account to minimize the wait time for any robot. However, determining the optimal plan for task allocation for a multirobot system (MRS) to maximize the expected overall system performance is typically an NP-hard problem [6]. It has been shown that the problem of persistent surveillance with energy and communication constraints is NP-complete [7] in a centralized system, and with the additional consideration of optimizing the total completion time, the complexity of the problem further increases.

The problem considered in our work, in centralized scenarios, is related to job-shop scheduling problem. It has been proved that one-machine job shop scheduling problem is NP-hard problem [8]. In [9], researchers studied job shop scheduling problem on a single machine and proposed an optimization algorithm called *marriage in honey-bees* that can provide a near-optimal solution. In [10], the authors introduced a customized approximate dynamic programming to the online version of the problem, where orders arrive at the system at random times. For multiple machine job shop scheduling problem, the authors in [11] developed several dominance properties and lower bounds for the problem of scheduling several independent jobs on multiple unrelated parallel machines with the objective of minimizing total tardiness. Processing times of a job on different machines are different on unrelated parallel-machine scheduling problems. For identical parallel machines job scheduling problem, the authors in [12] showed that the special case in which all jobs have equal processing times is solvable in polynomial time. The problem considered in our work can be treated as an extension of a multiple machine job scheduling problem with the objective of minimizing total tardiness. We investigate the problem of minimizing the sum of the total tardiness and the total traveling time.

Mobile robots deployed for outdoor applications generally rely on a decentralized communication architecture. Apart from robustness, a distinct advantage of decentralized communication is the requirement of fewer data packets among robots for task completion. This is especially beneficial in environments where communication is unreliable due to poor infrastructure or presence of adversarial elements [13]. In general, decentralized strategies can either be cooperative or non-cooperative. Cooperation arises in situations wherein multiple robots need to interact together to complete a task while increasing the total utility of the team [6]. Alternatively, cooperation is the interaction between the robots which work towards a common interest or reward [14]. Non-cooperation refers to a situation when robots act to fulfill their own self-interest [15]. In this case, robots with conflicting utility functions compete with each other. Each robot behaves selfishly from a sociological point of view because a single robot wants to make decision motivated by self-preservation. In such scenarios, a robot is unaware of the decision of other robots in the team. However, it may have information regarding their state, for example, relative position, orientation or configuration, based on the sensor information.

In the past, there has been some work related to charging mobile robotic platforms. In [16], authors provide a near-optimal recharging strategy on a self-sufficient robot boat. For one robot and multiple stations, the most effective strategy for the robot is to visit the closest station. Our current work takes into account the waiting time incurred while being queued at a station. In [17], the authors propose a strategy that does not require a robot to communicate, but to compete for the resource. They only consider the case of a single charging station. In contrast, our current work considers an arbitrary number of charging stations. In [18], the authors investigate the problem of when to recharge and how long to recharge for multi robot systems. They present a centralized solution which assumes robots communicate and share their states. Our current work addresses the problem of multirobot charging in the absence of any communication network which renders a centralized solution inutile in such circumstances. In [19], the authors address the problem of maintaining persistence in coordinated tasks performed by a team of autonomous robots, employing a dedicated team of charging robots to service the team to sustain their operations over extended periods of time. Unlike [19], our objective is to find a near-optimal allocation so that the total time spent in the recharging process for all robots is minimized.

In [20], the authors present a case study on three robots sharing a charging station with non-direct communication. The authors propose an experimental bottom-up approach, and three robots can remain in operation and efficiently share a charging station using simple mechanisms. The task of coming up with a sequence of actions that will achieve a goal is called planning [15]. In [21], the authors propose a consensus-based bundle algorithm that appropriately handles time windows of validity for tasks, fuel costs of the vehicles, and heterogeneity in the agent capabilities while preserving the robust convergence properties. Although the authors proposed methodologies for handling disconnected network, it still requires communication. In [22], the authors proposed a distributed approach for a team of heterogeneous UAVs cooperating efficiently in patrolling missions around irregular areas, with low communication ranges and memory storage requirements. In contradistinction, our objective is to find a planning solution for multiple robots and charging stations with communication constraints.

The main contributions of this work are as follows:

1) In contradistinction to current approaches that formulate multirobot recharging problem as a multi-vehicle routing problem [16], [23], [23], [24], we formulate the problem as an optimal resource allocation problem. We show that the optimal allocation of robots to charging stations that minimizes the total sum of traveling, queueing and charging time for the entire team in a centralized setting is NP-hard. In a non-cooperative setting that arises due to the lack of a communication network, we present charging strategies for the robots that are in Nash equilibrium.
2) Previous efforts in decentralized planning for recharging/refueling in multirobot systems either rely on the presence of an underlying communication network [25] or assume only one charging station [17]. Our work is the first to propose non-cooperative strategies for allocating a team of robots to multiple charging stations in the absence of a communication network.
3) In this work, we use the concept of *price-of-anarchy* (PoA) as a metric to evaluate the efficiency of the proposed strategies. Although, researchers have previously used PoA in other engineering systems, for example, computer networks [26], internet [27], and transportation networks [28], our work is the first to utilize it in the context of robotic networks.

The letter is organized as follows. In Section II, we present the problem formulation. In Section III, we show that the optimal solution to the recharging assignment problem in centralized scenarios is NP-hard. In Section IV, we utilize a game-theoretic approach to formalize the charging problem. We propose an algorithm to compute the pure strategy Nash equilibrium of the game, and prove its uniqueness. In Section V, we investigate the Price of Anarchy of the Nash Equilibrium. In Section VI, we consider the case of mobile charging stations. In Section VII, we present our conclusions.

II. Problem Formulation

In this section, we present the problem formulation. Consider n mobile robots $\mathbf{R} = \{r_1, r_2, r_3, \ldots, r_n\}$ located in a region. We assume that each robot is equipped with sensors that can accurately provide the location of the other robots. Consider a set of m identical charging stations $\mathbf{S} = \{s_1, s_2, s_3, \ldots, s_m\}$ located on the plane. We assume that the charging stations are stationary and their location is known to all the robots. Let t_{ij} denote the traveling time of robot r_i on the shortest path to station s_j. Let t_i^c denote the recharging time for robot r_i at any charging station. We assume that the charging time is the same for all robots at all stations. This is a reasonable assumption since in the absence of any information about other robots (except their location), each robot can assume a worst-case scenario in which other robots are fully discharged before they are being recharged at a station. Let q_{ij} denote the queuing time for robot r_i at station s_j.

An allocation A of robots to stations is an injective mapping from $\mathbf{R}$ to $\mathbf{S}$. Let $\mathcal{A} \in \mathbb{R}^{m\times n}$ denote the set of all allocations of robots to the charging stations. In a centralized scenario in which all the robots coordinate, we define the following cost function for an allocation A:

$$f_c(A) = \sum_{i=1}^{n} t_{iA(i)} + q_{iA(i)} + t_i^c \tag{1}$$

The objective is to find an optimal assignment, A^*, defined as follows:

$$A^* = \min_A f_c(A) \tag{2}$$

Intuitively, when the on-board power drops below a certain threshold for a robot, it has to abort its task and get charged. We want to reduce the time required for traveling and charging so that the robot is ready to resume its task at the earliest. The objective function in (2) is the aggregate time spent by all the robots before they get fully recharged. The problem in (2) is to find the allocation for n robots to visit m charging stations that minimizes the total time required for traveling and charging for all the robots.

Since $\sum_{i=1}^{n} t_i^c$ is a constant that is independent of the allocation A, f_c in (2) can be redefined as follows:

$$f_c(A) = \sum_{i=1}^{n} t_{iA(i)} + q_{ij} \tag{3}$$

In a non-cooperative scenario, each robot is assumed to be a selfish agent with its own payoff which depends on its *action* (in this case the station it chooses). Let $\mathcal{U}_i$ denote the action set of each robot which represents the set of stations that can be chosen by robot r_i. For our problem, $\mathcal{U}_i = \mathbf{S}$. The cost incurred by robot r_i is given by the function $J_i : \mathcal{U} \to \mathbb{R}$, where $\mathcal{U} = \Pi_1^n \mathcal{U}_i$ represents the joint action space of all players. The elements of $\mathcal{U}$ are *action profiles* $(s_{j_1}, \ldots, s_{j_n})$, where s_{j_i} is the station chosen by robot r_i. To put things in perspective, an action profile $(s_{j_1}, \ldots, s_{j_n})$ corresponds to an allocation A, where $A(i) = s_{j_i}$. However, unlike the centralized scenario, each robot chooses its own station in a non-cooperative setting. Since the queing time for a robot also depends on the stations chosen by other robots, let $q_i(s_{j_1}, \ldots, s_{j_n})$ denote the queing time of robot r_i for an action profile $(s_{j_1}, \ldots, s_{j_n})$. J_i is defined as follows:

$$J_i(s_{j_1}, \ldots, s_{j_n}) = t_{ij_i} + q_i(s_{j_1}, \ldots, s_{j_n}), \tag{4}$$

Since the robots are assumed to be selfish, the objective is to compute the action profile $\mathbf{s}^* = (s^*_{j_1}, \ldots, s^*_{j_n})$ which satisfies the following:

$$J_i(\mathbf{s}^*) \le J_i(s_{j_i}, s^*_{-j_i}) \quad \forall i \quad \forall s_{j_i} \in \mathbf{S}, \tag{5}$$

where $s^*_{-j_i}$ denotes the actions of all robots except r_i in the action profile $\mathbf{s}^*$.

Note: In this work, we consider a special case of the problem in which $t_1^c = t_2^c = \ldots, t_n^c = t_c$. We assume each robot can sense the location of the other robots without explicit communication. However, the station chosen by a robot is a private information which is not shared with other robots due to lack of a communication channel. Additionally, we assume that the nearest station associated to a robot is unique. In reality, a robot can break ties by giving priority to stations with lower indices. Specifically, if $j < k$, then the robot r_i will take $t_{ij} + q_{ij} = t_{ik} + q_{ik}, i \in R, j, k \in S$ as $t_{ij} + q_{ij} < t_{ik} + q_{ik}, i \in R, j, k \in S$.

III. Centralized Allocation Problem

In this section, we show that the centralized problem of allocating the charging stations to robots is NP-hard. First, we describe the job shop scheduling problem of minimizing the tardiness on a single machine with arbitrary release dates.

In the job shop scheduling problem, let n denote the number of jobs, r_i denote the release date or starting time of job i, and T_i denote its tardiness in a schedule. The objective is to find a schedule minimizing $\sum_{i=1}^{n} T_i$, i.e. a beginning time t_i and a completion time c_i for each job i, that minimizes the following objective:

$$\sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \max(c_i - d_i, 0) = \sum_{i=1}^{n} \max(t_i + p_i - d_i, 0),$$

subject to $t_i \ge r_i$ where p_i is the processing time of job i and d_i is its due time. This problem is known to be NP-hard if the job release dates are not identical [8].

Theorem 1: The problem of finding the allocation A^* (in (2)) is NP-hard.

Proof: We show a reduction from the job shop scheduling problem of minimizing the total tardiness on a single machine with arbitrary release dates to our multi-robot charging problem. We construct an instance of our recharging problem from the job shop scheduling problem. In our case, the n jobs that need to be processed on a single machine represent n robots that need to get recharged in a single station. The starting time of each job is t_i, which is equal to the time required by the robot to reach the station. The processing time for each job p_i can be used to represent the recharging time of robot r_i. The due time of each job is $t_i + p_i$, which indicates the time at which r_i will get fully charged. Clearly, we constructed a multiple robots recharging problem, which is a special case when $m = 1$, by using the job shop scheduling problem. A solution to the job shop scheduling problem can be constructed in polynomial time using the optimal solution to the multi-robot charging problem. For this instance, we can find the solution of optimal total charging plus queuing time as long as we can find the minimal tardiness in the job scheduling problem. ■

From the above analysis, we conclude that the centralized problem of finding the optimal allocation is computationally intractable. In the next section, we adopt a game-theoretic framework to find the optimal allocation in non-cooperative scenarios that arise due to lack of a communication network between the robots.

Algorithm 1: Assigning n Robots to m Stations.

Input: Two sets R and S. R are n robots and S are m charging stations.
Output: Charging assignment
1: **function** ASSIGNMENT $(\mathbf{R}, \mathbf{S})$
2: **while** R is not empty **do**
3: $Q = \emptyset$
4: **for** each station $s_j \in S$ **do**
5: $x = 0$ and $t_j = +\infty$
6: **for** each robot $r_i \in R$ **do**
7: **if** $t_{ij} + q_{ij} < t_j$ **then**
8: $x \leftarrow i$ and $t_j \leftarrow t_{ij} + q_{ij}$
9: **end if**
10: **end for**
11: $Q[j] \leftarrow x$
12: **end for**
13: Find $\min_{w \in [1,2,\ldots,|Q|]}(t_{Q[w]w})$
14: Assign robot $r_{Q[w]}$ to station s_w
15: Remove $r_{Q[w]}$ from R
16: **end while**
17: **return**
18: **end function**

IV. ALLOCATION AS A NON-COOPERATIVE GAME

Consider the following version of the charging problem under consideration which formulates it as a non-cooperative game. Assume that each robot behaves selfishly, and tries to minimize its own time required for getting fully charged. Each robot can sense the location of the other robots. However, the station chosen by a robot is unknown to other robots due to the lack of a communication network. Next, we present an algorithm to calculate the pure strategy Nash equilibrium (PSNE) for the non-cooperative game, and show the uniqueness of the PSNE. Then we show that the algorithm works for closed loop (each robot takes into account the current position of other robots) as well as open loop (each robot only considers the initial position of the other robots) information pattern.

A. Allocation Algorithm

The input of the Algorithm 1 are positions of n robots (denoted as R) and m stations (denoted as S). At the beginning of each iteration (Line 4 to Line 12), the algorithm finds the robot which incurs the lowest cost (travel time+queuing time) to get charged for each charging station, and adds them in an array (denoted as Q). The index of Q indicates the index of each station. The robot which incurs the lowest cost in Q is assigned to its corresponding charging station, and removed from Q. This process is repeated until all the robots have been assigned to stations.

Lemma 2: The assignment created by Algorithm 1 is a Nash equilibrium.

Proof: If any robot chooses a station different from the one chosen by Algorithm 1, its cost will increase since, in Line 13, we assign a robot to a station with the lowest cost at each stage. This implies that a unilateral deviation in the action of a single robot benefit it. Therefore, it is a Nash equilibrium. ■

The assignment process terminates in finite number of steps since the number of robots is finite. The algorithm is solvable in polynomial time. The two **for** loops from Line 4 to Line 12 take time of $O(n * m)$. Line 13 takes $O(m)$ time since finding the minimal number in an array has linear run-time. Line 14 and Line 15 take constant time. The while loop of Line 2 takes $O(n)$ time. Therefore, the complexity of the overall algorithm is $O(n * (n * m + m))$.

Next, we show the uniqueness of PSNE.

Lemma 3: The PSNE generated by Algorithm 1 is the unique NE.

Proof: We present a proof by contradiction. Let A denote the allocation obtained from Algorithm 1. Let $A' \neq A$ denote another allocation which is also a Nash Equilibrium (NE). We take each robot $r_i \in R$ and the assigned station $s_j \in S$ from allocation A, and compare it to the assignment from allocation A'. If the assignment is the same, we move on to the next robot. Otherwise, we can re-assign robot r_i to the assigned station in A to improve its cost, since the assignment from A is minimized at each stage, and the nearest station associated to a robot is unique. It contradicts the fact that A' is a NE which is that a player does not can receive any incremental benefit from unilateral deviation. Therefore, A' is not NE and allocation A generated by Algorithm 1 is the unique NE. ■

Therefore, the strategy that maximizes the utility of each robot is unique. If each robot implements its best response [29], the overall strategy is in Nash equilibrium.

An important question is the stability of Nash Equilibrium under different information patterns. In open-loop mode, the allocation is based on the initial position of the robots, and remains fixed thereafter. In closed-loop mode, the allocation is based on the current position of the robots, and is updated at each instant. In the next Lemma, we show that the NE obtained under both information patterns is the same.

Lemma 4: The Nash Equilibrium given by Algorithm 1 (which is open-loop) is also an equilibrium in closed-loop.

Proof: In Algorithm 1, the assignment order is based on the relative position between a station and its closest robot. Once the assignment is done, all the robots will start moving towards their allocated station together at a same constant speed. Since each robot moves in a straight line with the same speed, the closest robot $r_i \in \mathbf{R}$ to a station $s_X \in \mathbf{S}$ always remains the same. Therefore, the assignment remains the same in an closed-loop information pattern. ■

V. EFFICIENCY OF THE NASH EQUILIBRIUM

Nash equilibrium are known not to always optimize overall performance, with the Prisoner's Dilemma [30], [31] being the best-known example. Conditions under which Nash equilibrium can achieve the overall optimum have been studied extensively [32], [33]. In order to understand consequences of non-cooperative behavior in our problem, we need to know when is the cost of selfish behavior significantly large in comparison to a cooperative strategy [34]. The price of anarchy (PoA),

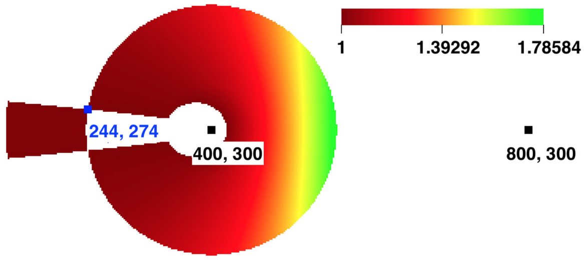


Fig. 1. Figure shows sample PoA for different location of r_2 when $t_c = 500$, the ratio of speed of the robots to charging time is 500, and robot r_1 and two stations are fixed. The two black dots represent the two station s_1 and s_2. The blue dot is the position of the robot r_1. The color at a given location represents the magnitude of PoA, if r_2 were at that location. The color chart along with PoA values is showed on top.

first defined by Koutsoupias and Papadimitriou [35], measures the extent to which competition approximates cooperation. The PoA measures the ratio between decentralized solution and centralized optimal solution. Lower PoA of a decentralized system implies higher efficiency of a system in a selfish behavior. In this section, we investigate the PoA of the Nash Equilibrium given by Algorithm 1.

In our problem, the positions of the robots and charging stations vary. In order to understand the properties under which Nash equilibrium can achieve the overall optimum, we analyze the PoA in this section. First, we thoroughly investigate the case of two robots and two charging stations, and find a tight bound on the PoA.

A. *PoA for 2 Robots and 2 Stations*

In this section, we compute the upper bound on PoA for 2 robots and 2 stations. Fig. 1 shows the variation of PoA for 2 robots and 2 stations. The blue dot represents r_1. The color at a given location corresponds to the magnitude of the PoA (> 1) if r_2 is present at that location. The white region represents PoA $= 1$, which means the efficiency of our solution is equal to the optimal solution.

In order to compute an upper bound on the price of anarchy, we need the total costs (Equation (3)) for both the decentralized solution from our non-cooperative Algorithm 1 and the optimal solution.

Fig. 2 shows all the cases for 2 robots and 2 station when $t_{11} < t_{21}$. We define a vector (q, p) as the allocation A, where q is A_1 and p is A_2. Since our goal is to compute the upper bound on PoA, the results for the symmetric case when $t_{11} > t_{21}$ follow. If both the robots choose station s_1 as the NE, the optimal cost is either the NE cost or the cost when robot r_1 chooses s_2 and r_2 chooses s_1.

Based on the allocation from the NE, if two robots are assigned to two different stations, the allocation is same as the allocation from the optimal solution. The cost of the optimal solution is different from the cost of the NE only when two robots are assigned to the same station (station s_1 when $t_{11} < t_{21}$). Therefore, we only focus on the allocation $(q, p) = (1, 1)$. We

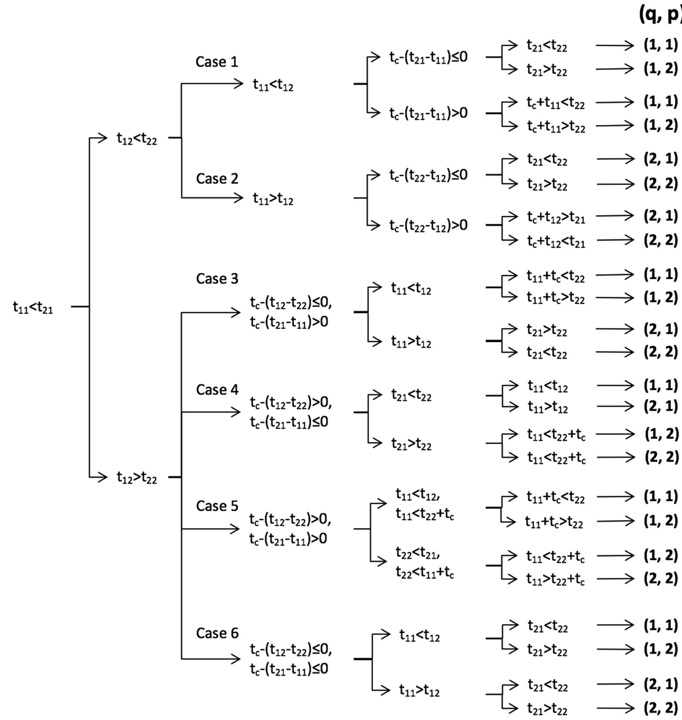


Fig. 2. Figure shows the allocation of the robots to the stations for all the 5 cases.

show the optimal cost can either be the NE cost or the cost when r_1 chooses s_2 and r_2 chooses s_1 case by case.

In case 1, and case 2, since robot r_1 dominates the game, it is closer to both stations s_1 and s_2. The reason why r_2 is not choosing s_2 is that the cost of choosing s_1 for r_2 is lower. Therefore, the optimal cost will not be the case when r_1 and r_2 choose s_1 and s_2, respectively. Similarly, the optimal cost is not the allocation when both robots choose s_2. The cost will increase for both r_1 and r_2 if they choose s_2. Therefore, for case 1 and case 2, the optimal cost can either be the NE cost or the cost when r_1 chooses s_2 and r_2 chooses s_1.

In case 3, the cost when both robots choose s_1 is $t_{11} + t_{11} + t_c$. The cost when robot r_2 change to s_2 while r_1 choose s_1 is $t_{12} + t_{22}$. However, the condition when both robots choose s_1, $t_{11} + tc < t_{22}$. And if both robots choose s_2, the cost is $t_{12} + t_{22}$, which is more than $t_{11} + t_{11} + t_c$ since $t_{12} > t_{11}$ and $t_{22} > t_{11} + t_c$. Therefore, the statement holds for case 3.

In case 4, the cost when both robots choose s_1 is $t_{11} + t_{21}$. The cost when robot r_2 chooses s_2 while r_1 chooses s_1 is $t_{11} + t_{22}$. However, the condition when both robots choose s_1 is $t_{22} > t_{21}$. Therefore the cost is larger and cannot be the optimal cost. Moreover, if both robots choose s_2, the cost is $t_{22} + t_{22} + t_c$, which is more than $t_{11} + t_{21}$ since $t_{22} > t_{21}$ and $t_{11} < t_{12} < t_{22} + t_c$. Moreover, if r_1 and r_2 choose s_2 and s_1 respectively, the cost is more than that of the NE cost, since $t_{12} > t_{11}$. Therefore, the statement holds for case 3. Similar reason holds for case 5 and case 6. Therefore, when $t_{11} < t_{21}$, if the robots both choose station s_1, the optimal cost is either the NE cost or the cost when robot r_1 chooses s_2 and r_2 chooses s_1.

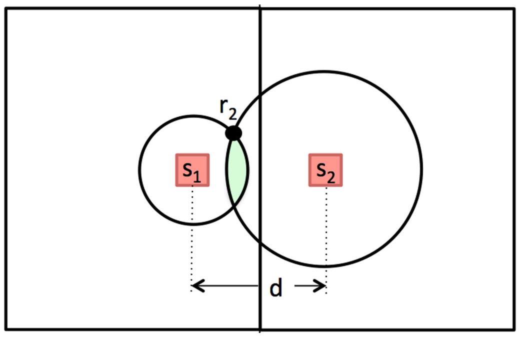


Fig. 3. Shaded green region shows the possible location of robot r_1 (for a fixed location of r_2) when $(t_{11} < t_{21}) \wedge (t_{12} < t_{22}) \wedge (t_{11} < t_{12})$.

We show that the upper bound on the PoA is 2. Among the five cases, the PoA is $\min(\frac{f_c(1,1)}{f_c(1,1)}, \frac{f_c(1,1)}{f_c(2,1)})$, bounded by a constant 2 regardless of the position of the stations and the robots. Among the five cases that we analyzed, the situation when PoA is not equal to 1 can possibly occur only when $t_c - (t_{21} - t_{11}) > 0$ and $p = 1, q = 1$. $\frac{f_c(1,1)}{f_c(2,1)} = \frac{2t_{11}+t_c}{t_{12}+t_{21}} = \frac{t_{11}}{t_{12}+t_{21}} + \frac{t_{11}+t_c}{t_{12}+t_{21}}$. We can analyze the expression with two parts, $\frac{t_{11}}{t_{12}+t_{21}}$ and $\frac{t_{11}+t_c}{t_{12}+t_{21}}$. For the first part, $\frac{t_{11}}{t_{12}+t_{21}}$, the maximum value is bounded by 0.5, since $(t_{11} < t_{12}) \wedge (t_{11} < t_{21})$. If t_{11} extremely close to both t_{12} and t_{21}, it reaches the maximum value. For the second part, $\frac{t_{11}+t_c}{t_{12}+t_{21}}$, instead of finding the upper bound, we find the upper bound for $\frac{t_{22}}{t_{12}+t_{21}}$ which is always greater than $\frac{t_{11}+t_c}{t_{12}+t_{21}}$ since $t_{22} > t_{11} + t_c$. We define d as the distance between the two stations. To find maximum of $\frac{t_{22}}{t_{12}+t_{21}}$, we need to find the bound for the denominator. The minimum of t_{12} is between $d - t_{21}$ and $\frac{d}{2}$, since $(t_{11} < t_{21}) \wedge (t_{12} < t_{22}) \wedge (t_{11} < t_{12})$. Replacing t_{12} with this value, we obtain $\max(\frac{t_{22}}{d}, \frac{t_{22}}{\frac{d}{2}+t_{21}})$. In the region shaded in green in Fig. 3 which represents $(t_{11} < t_{21}) \wedge (t_{12} < t_{22}) \wedge (t_{11} < t_{12})$, we obtain $t_{22} < \frac{3}{2d}$. Moreover, $\frac{t_{22}}{\frac{d}{2}+t_{21}} < \frac{d+t_{21}}{\frac{d}{2}+t_{21}}$, since $d + t_{21} > t_{22}$. Finding $\max(\frac{t_{22}}{d}, \frac{t_{22}}{\frac{d}{2}+t_{21}})$ can be simplified into finding $\max(\frac{\frac{3}{2}d}{2d}, 1 + \frac{\frac{d}{2}}{t_{21}+\frac{d}{2}})$, which is $1\frac{1}{2}$. Therefore the total PoA is bounded by 2.

B. Results for $n, m > 2$

In this section, we show the variation of PoA as a function of the number of robots and stations based on extensive simulation. We record the average and maximum PoA for n robots and m stations, where $n = 2, 3, 4, 5, 6, 7$ and $m = 2, 3, 4, 5, 6, 7$ from the simulations. For each simulation, we randomly generate initial locations for n robots and m stations from a uniform distribution within a bounded region on the plane. If the robots are far from all charging stations, the travel time to the station dominates the cost incurred by each robot and is the same irrespective of the station to which they are allocated. From the simulations, we see that robots located farther than 10 times the maximum distance between the stations do not affect the PoA. Therefore, we restrict the charging stations to be located within a region of 100×150 pixels in our simulations, and the robots to be located in a rectangular region of 1000×1500 pixels around the region containing the charging stations. The ratio between charging rate and speed of the robots is set at 500. For each simulation, we calculate the optimal cost by finding the minimum cost among all the possible permutations between the robots and the charging stations since we have shown that the problem is NP-complete in Section III. For a specific n and m, the result is based on 5000 rounds of simulation.

TABLE I
TABLE SHOWS THE AVERAGE PoA ALONG WITH THE HEAT MAP BASED ON 5,000 ROUNDS OF SIMULATION OF THE GAME

		Stations					
		2	3	4	5	6	7
Robots	2	1.01086	1.01274	1.00995	1.00993	1.00747	1.00835
	3	1.01238	1.02034	1.01745	1.01818	1.01571	1.01625
	4	1.01659	1.02244	1.02222	1.02322	1.02317	1.02167
	5	1.00723	1.02269	1.02504	1.02945	1.02615	1.02694
	6	1.00705	1.02394	1.02775	1.02871	1.03133	1.02993
	7	1.00306	1.01315	1.02961	1.02712	1.03027	1.03137

TABLE II
TABLE SHOWS THE MAXIMUM PoA ALONG WITH THE HEAT MAP FROM 5,000 ROUNDS OF SIMULATION OF THE GAME

		Stations					
		2	3	4	5	6	7
Robots	2	1.66851	1.62181	1.69156	1.40916	1.5444	1.67181
	3	1.42858	1.63709	1.67409	1.4536	1.47286	1.44277
	4	1.32195	1.33033	1.47609	1.38571	1.50148	1.33695
	5	1.24279	1.26457	1.27417	1.33234	1.29548	1.3417
	6	1.24836	1.25434	1.34198	1.23906	1.2639	1.28538
	7	1.17521	1.18062	1.23463	1.17825	1.23797	1.26657

Table I shows the average PoA for each specific n and m. The average PoA is close to 1 which means that the NE has a high efficiency. This implies that a selfish behavior is also close to social optimum. From the heat-map, we observe that the average PoA slowly increases with an increase in the number of robots and stations. The maximum of the simulated average PoA is 1.03137 for 7 robots and 7 stations. However, the average of PoA is still close to 1. Table II shows the maximum PoA. From the simulation results, we observe that the maximum PoA is less than 2 even for $n, m > 2$. With increasing number of stations, the maximum PoA is decreasing. As the number of stations increases, the maximum PoA decreases. Intuitively, with increasing number of stations, the robots can lower their queuing times since they have more stations to choose from. The total cost will be dominated by the traveling time in the optimal solution. Therefore, with increasing number of stations, the maximum PoA is decreasing. As the number of robots increases, the queuing time is expected to increase since each station has to serve a larger number of robots. Therefore, the PoA decreases with increasing number of robots and stations. From the diagonal cells in the Table II, we observe that the maximum PoA decreases with an increase in the number of robots or stations.

VI. MOBILE CHARGING STATIONS

In this section, we explore the scenario when the charging stations are mobile. Mobility of charging stations improves the total cost (Equation (3)) incurred by the robots. We assume that the charging stations and robots have the same speed. The total cost in this case also includes the time for which the charging stations are mobile in addition to the traveling time and queuing time of the robots. We present two strategies that decrease the total cost. Both strategies are based on Algorithm 1.

A. Centroidal Rendezvous

An important property of the centroid is that the centroid of a set of vectors minimizes the weighted sum of the generalized squared Euclidean distances from the vectors to any point in the space [36]. We utilize this property to decrease the traveling time of the robots. For each charging station s_j, a set $R(s_j) \subseteq R$ of n_j robots are allocated to it.

$$R(s_j) = \{r_1, r_2, ..., r_{n_j}\} = \{(x_1, y_1), (x_2, y_2), ...(x_{n_j}, y_{n_j})\}$$

We define centroid point $c(s_j)$ of a charging station s_j as the center of set $R(s_j)$.

$$c(s_j) = (\bar{x_j}, \bar{y_j}) = \left(\frac{1}{n_j} \sum_{i=1}^{n_j} x_i, \frac{1}{n_j} \sum_{i=1}^{n_j} y_i \right)$$

In this strategy, all charging stations move towards the centroid of the robots allocated to the station while robots move towards their charging stations. If any robot reaches its charging station before the charging station reaches the centroid, the charging station stops, and fully charges the robot. After the charging is completed, the station resumes its motion towards the centroid. The charging station stops moving after it reaches the centroid.

B. Charging Stations Move Towards the Closest Robot

In this strategy, a charging station s_j moves towards the closest robot among the ones allocated to it. The robot closest to a charging station s_j is defined as follows:

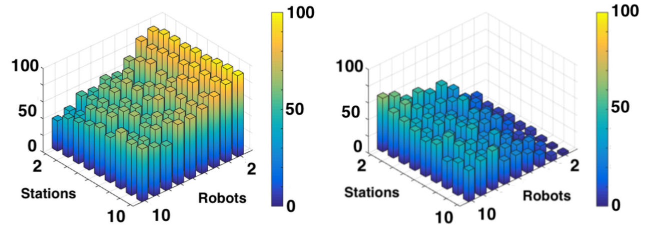
$$r_{R(s_j)} = \left\{ r_i \,\middle|\, \min_{\forall r_i \in R(s_j)} dist(r_i, s_j) \right\}.$$

As the charging station moves towards the closest robot, all the robots allocated to it move towards the station. If any robot reaches its charging station, the charging station stops, and fully charges the robot. After the charging is completed, the station resumes its motion towards the next closest robot until all robots get charged.

C. Simulation Results

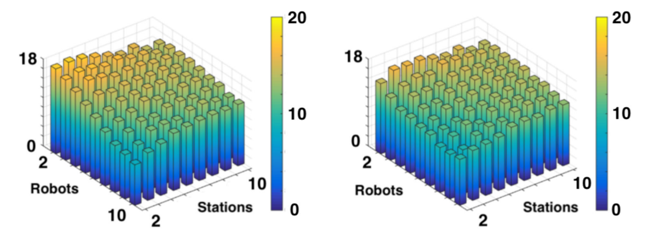
In this section, we compare the costs incurred by the two strategies with the costs when the charging stations are immobile, and the allocation strategy for the robots is given by Algorithm 1. We simulate the recharging process for n robots and m stations where $n = 2, 3, 4, ..., 10$ and $m = 2, 3, 4, ..., 10$. Each simulation is one round of implementation with a given number of robots and stations. For each simulation, we randomly generate initial locations for n robots and m stations from a uniform distribution over a rectangular region of dimension 1000×1500 pixels for 100 rounds. The ratio between the charging rate and speed of robots is set at 500.

We simulate the recharging process for three strategies: (1) Robots with static charging stations (2) Centroidal rendezvous



(a) (b)

Fig. 4. Figure (a) and (b) show the number of times strategy (2) and (3) wins in 100 rounds respectively.



(a) (b)

Fig. 5. Figure (a) and (b) show the percentage improvement in the total cost of the strategies (2) and (3) compared to the cost incurred for static charging stations.

(3) Nearest robot pursuit. Then we compare the total cost of robots for these three strategies, and declare the strategy that has the lowest cost as the winning strategy. Finally, we count the number of times each strategy wins in 100 rounds. Strategy (1) never wins which implies mobile charging stations with strategies (2) and (3) can improve the total cost in every simulation. The simulation results are shown as two three-dimensional bar graphs in Fig. 4, where Fig. 4a and Fig. 4b represent the number of times strategies (2) and (3) win in 100 rounds, respectively. We find strategy (2) is better than strategy (3) when the number of robots is less than or equal to the number of charging stations.

Next, we show the percentage improvement in total cost of the strategies (2) and (3). The percentage improvement for a strategy for mobile charging stations is defined as reduction in the cost from static charging stations expressed as a percentage. The results are based on 200 rounds of simulation. The simulation results are shown as two three-dimensional bar graphs in Fig. 5. Fig. 5a and Fig. 5b show the percentage improvement for strategies (2) and (3), respectively, based on strategy (1). From the results in Table 5, we can conclude that strategies (2) and (3) can improve the total cost by over 8%. From the simulation results, we can conclude that strategy (2) is better than strategy (3) in most of the simulation scenarios.

VII. Conclusions

This work considered the problem of assigning multiple robots to charging stations in order to minimize the total time required by all robots for the charging operation. We first showed that the centralized problem is NP-hard. Then we formulated the charging problem as a non-cooperative game. We proposed an algorithm to obtain the pure strategy Nash equilibrium of the non-cooperative game, and showed its uniqueness. We investigated the price of anarchy (PoA) of this equilibrium as a function of the number of robots and stations. Next, we leveraged our analysis on static charging stations to propose strategies for reducing the total cost with mobile charging stations. Finally, we analyzed the two strategies through extensive simulation.

As a future research direction, we plan to investigate the performance of the proposed strategies for different speeds of the robots and charging stations, unequal charging times between robots and charging stations, and different storage capacity of batteries on-board the robots. Another interesting direction of future work is to investigate the problem in environments containing obstacles, and address practical problems, for example, collision and obstacle avoidance. Finally, investigating the improvement in the cost due to local communication (between neighboring robots), or specific communication topology is a problem of interest in the future.

References

[1] T. Arai, E. Pagello, and L. E. Parker, "Advances in multi-robot systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 655–661, Oct. 2002.

[2] E. Bahceci, O. Soysal, and E. Sahin, "A review: Pattern formation and adaptation in multi-robot systems," Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-03-43, 2003.

[3] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 1, pp. 1050–1055.

[4] K. A. Swieringa *et al.*, "Autonomous battery swapping system for small-scale helicopters," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3335–3340.

[5] T. Kundu and I. Saha, "Charging station placement for indoor robotic applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3029–3036.

[6] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 12, p. 6, 2013.

[7] J. Scherer and B. Rinner, "Persistent multi-UAV surveillance with energy and communication constraints," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2016, pp. 1225–1230.

[8] A. R. Kan, *Machine Scheduling Problems: Classification, Complexity and Computations*. Berlin, Germany: Springer, 2012.

[9] C.-C. Wu, S.-R. Cheng, W.-H. Wu, Y. Yin, and W.-H. Wu, "The single-machine total tardiness problem with unequal release times and a linear deterioration," *Appl. Math. Comput.*, vol. 219, no. 20, pp. 10401–10415, 2013.

[10] D. P. Ronconi and W. B. Powell, "Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming," *J. Scheduling*, vol. 13, no. 6, pp. 597–607, Dec. 2010.

[11] S.-O. Shim and Y.-D. Kim, "Minimizing total tardiness in an unrelated parallel-machine scheduling problem," *J. Oper. Res. Soc.*, vol. 58, no. 3, pp. 346–354, Mar. 2007.

[12] T. Cheng and Z.-L. Chen, "Parallel-machine scheduling problems with earliness and tardiness penalties," *J. Oper. Res. Soc.*, vol. 45, no. 6, pp. 685–695, 1994.

[13] V. J. Lumelsky and K. Harinarayan, "Decentralized motion planning for multiple mobile robots: The cocktail party model," *Auton. Robots*, vol. 4, no. 1, pp. 121–135, 1997.

[14] D. Barnes and J. Gray, "Behaviour synthesis for co-operant mobile robot control," in *Proc. Int. Conf. Control*, 1991, pp. 1135–1140.

[15] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. London, U.K.: Pearson Education, 2003.

[16] J. Wawerla and R. T. Vaughan, "Near-optimal mobile robot recharging with the rate-maximizing forager," in *Advances in Artificial Life*, F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, and A. Coutinho, Eds., Berlin, Germany: Springer, 2007, pp. 776–785.

[17] A. Munoz, F. Sempé, A. Drogoul, and R. G. Leclerc, "Sharing a charging station in collective robotics," Tech. Rep. LIP6 2002/026, Paris, France, 2002.

[18] F. Michaud and E. Robichaud, "Sharing charging stations for long-term activity of autonomous robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, vol. 3, pp. 2746–2751.

[19] N. Mathew, S. L. Smith, and S. L. Waslander, "A graph-based approach to multi-robot rendezvous for recharging in persistent tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3497–3502.

[20] F. Sempé, A. Muñoz, and A. Drogoul, "Autonomous robots sharing a charging station with no communication: A case study," in *Distributed Autonomous Robotic Systems 5*, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds., Tokyo, Japan: Springer, 2002, pp. 91–100.

[21] S. Ponda, J. Redding, H. L. Choi, J. P. How, M. Vavrina, and J. Vian, "Decentralized planning for complex missions with dynamic communication constraints," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 3998–4003.

[22] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 1, pp. 3–8, 2013.

[23] P. Maini, K. Yu, P. Sujit, and P. Tokekar, "Persistent monitoring with refueling on a terrain using a team of aerial and ground robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 8493–8498.

[24] K. Sundar, S. Venkatachalam, and S. Rathinam, "Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem," in *Proc. Amer. Control Conf.*, 2016, pp. 6489–6494.

[25] J. Senthilnath, K. Harikumar, and S. Suresh, "Efficient resource allocation for decentralized heterogeneous system using density estimation approach," *Encyclopedia Semantic Comput. Robot. Intell.*, vol. 2, no. 01, 2018, Art. no. 1850008.

[26] R. Johari, S. Mannor, and J. N. Tsitsiklis, "Efficiency loss in a network resource allocation game: The case of elastic supply," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1712–1724, Nov. 2005.

[27] T. Roughgarden and É. Tardos, "Bounding the inefficiency of equilibria in nonatomic congestion games," *Games Econ. Behav.*, vol. 47, no. 2, pp. 389–403, 2004.

[28] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks by estimating user cost functions from actual traffic data," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 789–794.

[29] J. B. Rosen, *Existence and Uniqueness of Equilibrium Points for Concave n-Person Games*. Stanford, CA, USA: Stanford University, 1964.

[30] G. Owen, *Game Theory*. New York, NY, USA: Academic, 1982.

[31] G. Kendall, X. Yao, and S. Y. Chong, *The Iterated Prisoners' Dilemma: 20 Years On*, vol. 4. Singapore: World Scientific, 2007.

[32] S. Özyıldırım and N. M. Alemdar, "Learning the optimum as a Nash equilibrium," *J. Econ. Dyn. Control*, vol. 24, no. 4, pp. 483–499, 2000.

[33] E. Maskin, "Nash equilibrium and welfare optimality," *Rev. Econ. Stud.*, vol. 66, no. 1, pp. 23–38, 1999.

[34] T. Roughgarden, *Selfish Routing and the Price of Anarchy*, vol. 174. Cambridge, MA, USA: MIT Press, 2005.

[35] C. H. Papadimitriou, "Algorithms, games, and the internet," in *International Colloquium on Automata, Languages, and Programming*. Berlin, Germany: Springer, 2001, pp. 1–3.

[36] H. Abdi, "Centroids," *Wiley Interdisciplinary Rev., Comput. Statist.*, vol. 1, no. 2, pp. 259–260, 2009.