Cooperative Set Function Optimization Without Communication or Coordination

Gustavo Malkomes
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, 63130, USA
luizgustavo@wustl.edu

Kefu Lu
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, 63130, USA
kefulu@wustl.edu

Blakeley Hoffman
Computer Science and Engineering, Mathematics
University of South Carolina
Columbia, SC, 29208, USA
hoffman1@email.sc.edu

Roman Garnett
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, 63130, USA
garnett@wustl.edu

Benjamin Moseley
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, 63130, USA
bmoseley@wustl.edu

Richard Mann
School of Mathematics
University of Leeds
Leeds LS2 9JT, UK
r.p.mann@leeds.ac.uk

ABSTRACT

We introduce a new model for cooperative agents that seek to optimize a common goal without communication or coordination. Given a universe of elements $\mathcal{V}$, a set of agents, and a set function f, we ask each agent i to select a subset $S_i \subset \mathcal{V}$ such that the size of S_i is constrained (i.e., $|S_i| < k$). The goal is for the agents to cooperatively choose the sets S_i to maximize the function evaluated at the union of these sets, $\cup_i S_i$; we seek max $f(\cup_i S_i)$. We assume the agents can neither communicate nor coordinate how they choose their sets. This model arises naturally in many real-world settings such as swarms of surveillance robots and colonies of foraging insects. Even for simple classes of set functions, there are strong lower bounds on the achievable performance of coordinating deterministic agents. We show, surprisingly, that for the fundamental class of submodular set functions, there exists a near-optimal distributed algorithm for this problem that does not require communication. We demonstrate that our algorithm performs nearly as well as recently published algorithms that allow full coordination.

Keywords

Cooperative agents, Set function optimization, Communication, Coordination

1. INTRODUCTION

We consider the problem of designing policies for groups of agents to jointly maximize a collective goal without communication. An archetypal example of decentralized collective action with only sporadic or episodic communication is found in eusocial insect colonies. Being genetically identical, these agents behave so as to serve their collective fitness, for example by maximising the food intake of the whole colony. Moreover, individuals, while acting for the collective, must make decisions independently; [12] states that “A social insect colony operates without any central control; no one is in charge, and no colony member directs the behavior of another.”, while [32] notes that a honeybee colony is “an ensemble of largely independent individuals that rather infrequently exchange information (directly or indirectly) with one another” (from [36]). The marginal utility of those independent choices is dependent on the decisions of others. How can these insects act effectively as a collective without a pre-coordinated strategy? Although the specific mechanisms of decision-making in each species can be highly complex [36], we should expect insect colonies to have evolved mechanisms that provide highly effective allocations of foraging for their typical environment. Therefore, identifying optimal procedures for collective, decentralized action is likely to shed light on the behavior of these colonies.

For a similar engineering challenge, consider swarms of autonomous underwater vehicles (AUVs). Recently, advancements in this technology have lead to algorithmic studies of these vehicles in cooperative settings to achieve tasks such as search, surveillance, and archaeological research [3]. During such operations, a single AUV takes many thousands of images near the sea floor before surfacing and transmitting the images for further analysis. As underwater communication is difficult to maintain, the AUVs cannot communicate with each other during the search; each is an independent agent. Naturally, performing operations underwater is expensive; thus it is vital to optimize the performance of the team as a whole. Previous work has only considered rudimentary coordination of AUVs, describing beneficial static formations for the search task [26].

We consider an abstract generalization of this problem that captures the above settings and others. We consider a team of n agents cooperatively maximizing a set function f. The agents may not communicate with each other and are indistinguishable. Each agent has a budget k on the number of elements they may choose. The team shares a reward equal to the value of the union of elements selected, as measured by f. We provide a decentralized randomized algorithm to solve this problem. Remarkably, the algorithm achieves a near-optimal approximation for the fundamental

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

class of submodular reward functions, despite the seemingly strong constraints on communication.

Our problem formation makes no assumption on the reward function; however, optimizing an arbitrary set function is intractable. Fortunately, many real-world reward functions belong to the class of *submodular functions,* which capture a natural notion of diminishing returns. Optimizing submodular functions has been considered in both sequential and distributed computational models. However, no previous work has addressed the multi-agent optimization problem considered in this paper.

One of the most well-studied problems is maximizing a monotone submodular function subject to a cardinality constraint. In this problem a single agent selects a subset S, $|S| \leq \alpha$, to maximize a submodular set function f evaluated on S. By setting $\alpha = nk$, this is precisely the problem the agents are trying to optimize in our problem formulation (the agents can collectively return at most nk unique elements). This problem is NP-hard even for a single agent. Further, a simple greedy procedure achieves a $(1 - \frac{1}{e})$-approximation, and this is the best possible assuming P$\neq$NP [7]. This algorithm is known the have strong practical performance [17]. Our work aims to discover how close we can come to this best-possible efficient algorithm using a decentralized strategy.

Contributions: In this paper, we formalize the problem we call *cooperative set function optimization without communication.* We show that for submodular reward functions there is a $(1 - \frac{1}{e})^2$-approximation algorithm. Therefore, the "price of decentralization" is only a small constant factor loss in the approximation guarantee. We further establish, using experiments on real-world data, that our algorithm is competitive with recently proposed sophisticated centralized algorithms [24, 5].

2. RELATED WORK

The problem formalized in this paper is new. However, there are several related problems that have been well studied. In the area of *cooperative game theory* and *multi-agent systems*, cooperative coordination among agents has been widely used for improving the performance of a task in comparison to single-agent decisions [30, 16, 33, 38]. A considerable amount of this literature is focused on coalition formation processes, i.e., how the agents should be partitioned [29, 40, 15], the division of coalition payoff [40] and how to design cooperation enforcement mechanisms [22] are some examples.

Comparatively less attention has been given to the case that agents cannot communicate or coordinate. [11] have considered how automated agents could use a technique called *focal points* to perform coordination without communication. A key assumption of their method is that agents are interested in prominent objects which can be easily identified. Our framework does not impose restrictions of this nature.

Another related topic is the formation of *ad hoc* teams — agents who do not know each other but they face a situation in which they have to cooperate [1, 2, 14, 35, 39]. These works differ from ours because they allow communication among the agents.

Our work is also similar in spirit with [28]. These authors introduce and analyze the so-called *joint search* problem. In joint search, the agents need to select a particular option among several choices, and all agents may benefit from the final outcome. Joint search differs from our problem because the agents are given disjoint search spaces, which is a pre-coordination step. Nevertheless, in our experimental section, we compare with a baseline called CENTRAL-PARTITION, which effectively matches the joint search formulation. Interestingly, as we shall see, our algorithm can in some cases outperform CENTRAL-PARTITION. [28] used tools from game theory to analyze their model, whereas we cast our problem as (multi-agent) set function optimization.

In the set function optimization literature, the maximization of a submodular set function subject to a cardinality constraint is well studied. For non-monotone submodular functions there is a constant factor approximation and for monotone submodular functions there is a better ratio, $(1-\frac{1}{e})$ [8, 25, 20]. These results are for standard computational models and do not admit a naïve adaptation to the multi-agent problem we consider.

A problem similar to our setting is the *submodular welfare problem.* In this problem there are a set of agents, each with their own value function. The objective is to maximize the sum of the values of the set each agent receives. The distinct difference between this setting and ours is that our objective considers the function evaluated on the union of the items selected (rather than summing the value of the sets). The submodular welfare problem is designed to capture the utility achieved where each agent has their own evaluation, and in our setting the agents seek to optimize a common goal. Some work on the welfare problem, such as [21], specifically requires each agent's set to be disjoint. However, this problem is usually studied as an allocation problem, with a centralized agent allocating items to the collective. Several constant approximations are known [6, 9]. Unfortunately, these results do not extend to our problem where there is neither communication nor centralized planning.

3. PROBLEM DESCRIPTION

Here we formally define the problem we call *cooperative set function optimization without communication.* Let $\mathcal{V}$ be an arbitrary finite ground set with $|\mathcal{V}| = m$. Let $f: 2^{\mathcal{V}} \to \mathbb{R}$ be an arbitrary set function. There exists a set of n agents, each of whom selects a set $S_i \subset \mathcal{V}$ with $|S_i| \leq k$, where $k \geq 1$ is given as input to the problem. The subscript i denotes that this is the subset selected by the ith agent. The agents must make these selections without exchanging information. The only information available to the agents is the total number of agents. This implies that the agents are indistinguishable. After the selection stage, the agents share a global reward of $f(\cup_{i=1}^{n} S_i)$.

Unfortunately, without further assumptions, maximizing this reward is no easier than the problem of maximizing a set function under cardinality constraints; if we let $n = 1$, we recover exactly that problem. For general functions it is easy to see that one has to evaluate all $\binom{m}{nk}$ possible sets if you are only given oracle access to f and desire bounded error.

Luckily many real-world value functions are tractable. We assume the set function is a non-negative monotone submodular function; that is, f satisfies

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \quad \forall X \subseteq Y \subseteq \mathcal{V}, x \in \mathcal{V}$$

This condition implies the marginal gain in value from a specific element to a set X cannot increase after the inclusion of other elements in X. We do not make further assumptions about the function f. As standard in submodular function literature, we assume that we are given oracle access to f.

Agent Strategies: We now formalize the types of strategies for the agents that will be considered in this paper. We consider a restricted class of mixed strategies of the following form. Motivated by the fact that the agents are indistinguishable, we assume that each agent uses the same strategy. Let $\pi \in \mathbb{R}^m$ be a probability distribution over $\mathcal{V}$. Given π, each agent i samples and returns a set of size at most k from the corresponding multinomial distribution:

$$S_i \sim \mathcal{MN}(\pi, k).$$

We will use the shorthand notation $S_i \sim \pi$ to indicate drawing from this distribution. S_i might have cardinality less than k because we sample the elements independently with replacement.[1] All agents use the same strategy and do not need to communicate with one another. Further, each agent may select no more than k elements. A strategy which takes this form clearly obeys the constraints of the problem. The goal is to determine a suitable probability distribution π over the elements in $\mathcal{V}$.

4. ALGORITHMS

In this section, we give our algorithms for determining π and show that it is near optimal. We begin by describing a simple sampling procedure and later we will show how to improve on this solution using local optimization methods.

4.1 Sampling from the greedy solution

As mentioned, there is a simple greedy procedure for optimizing a monotone submodular function subject to a cardinality constraint. Our algorithm for each agent is designed to closely mimic this greedy algorithm. To begin, we define the standard greedy procedure. Say that α elements in total are to be chosen from $\mathcal{V}$ to form some set S. A greedy algorithm for solving this problem is as follows. We begin by initializing $S \leftarrow \emptyset$. Then we sequentially find the element $x \in \mathcal{V} \setminus S$ with the largest marginal gain, $i^* = \arg\max_{i \in \mathcal{V}\setminus S} f\big(S \cup \{i\}\big) - f(S)$, and add this element to S, setting $S \leftarrow S \cup \{i^*\}$. We continue in this manner until α elements have been selected. We refer to this simple algorithm as $\textsc{greedy}(\mathcal{V}, f, \alpha)$.

Note that this algorithm is completely deterministic. Perhaps the most natural (and unfortunately bad) strategy for our distributed setting would be for all agents to run $\textsc{greedy}(\mathcal{V}, f, k)$ and return the resulting set. This gives a poor solution because every agent will select the same elements, receiving a shared reward of $f(\cup_{i=1}^n S_i) = f(S)$, under the constraint $|S| = k$, which can be a factor n smaller than the best-possible solution.

An ideal solution would be to coordinate the agents such that the union of their outputs is equal to $\textsc{greedy}(\mathcal{V}, f, nk)$. That is, each agent selects k distinct elements corresponding to running the greedy algorithm with $\alpha = nk$. This ideal is impossible to achieve without coordination. We present a simple randomized algorithm designed to simulate $\textsc{greedy}(\mathcal{V}, f, nk)$. First, each agent runs GREEDY to identify nk elements and then samples these elements uniformly at random k times. This strategy will be called $\textsc{greedy-sampling}(\mathcal{V}, f, nk)$.

More precisely, let G be the set returned from running $\textsc{greedy}(\mathcal{V}, f, nk)$. The $\textsc{greedy-sampling}(\mathcal{V}, f, nk)$ strategy sets the following probability distribution over all elements i in $\mathcal{V}$:

$$\pi_i = \begin{cases} \frac{1}{nk} & i \in G; \\ 0 & i \notin G. \end{cases}$$

Note that this strategy belongs to the class of mixed strategies we are studying since it creates a probability distribution π from which an agent samples k times.

THEOREM 1. *For any function f that is nonnegative, monotonic, and submodular the,* GREEDY-SAMPLING *procedure has an approximation guarantee of* $(1 - \frac{1}{e})^2$.

Proof of [Theorem 1] Let OPT denote the value of the best possible solution

$$\textsc{opt} = \max_{S \subseteq \mathcal{V}, |S| \le nk} f(S).$$

Our goal is to show that

$$\mathbb{E}_{S_i \sim \pi}\left[f\left(\bigcup_{i=1}^{n} S_i\right)\right] \ge \left(1 - \frac{1}{e}\right)^2 \textsc{opt}.$$

As before, let G be the set returned by $\textsc{greedy}(\mathcal{V}, f, nk)$. Let the elements of G be ordered $e_1, e_2, \ldots e_{nk}$ in the order selected by the greedy algorithm. Let $G_i = \cup_{1 \le j \le i}\{e_i\}$ be the first i elements that GREEDY would choose. Let $w_i = f\big(G_{i-1} \cup \{e_i\}\big) - f(G_{i-1})$ be the incremental value that element e_i gives to the greedy solution. We begin by bounding the probability that element e_i is included in the final solution. The probability that an individual agent j includes e_i is $\frac{k}{nk} = \frac{1}{n}$ and therefore, the probability that an element e_i is not in S_j is $1 - \frac{1}{n}$. The probability that e_i is not chosen by any agent is then $(1 - \frac{1}{n})^n \le \frac{1}{e}$ (this inequality holds for all n). Thus, the probability that some agent choose e_i is at least $1 - \frac{1}{e}$.

Now we will show that $f(\cup_i S_i) \ge \sum_{e_j \in \cup_i S_i} w_j$; in particular, we show that for any $X \subseteq G$, $f(X) \ge \sum_{e_j \in X} w_j$. Fix any set X and let $e'_1, e'_2, \ldots e'_{|X|}$ be the ordering of the elements in X in the same order these elements appear in the ordering of G. Let X_i denote the first i elements in the order and for notational convenience set $X_0 = \emptyset$. Finally, set $\rho(i)$ be the index of e'_i in the ordering of the elements in G. We see that,

$$\begin{aligned} f(X) &= f(\emptyset) + \sum_{i=1}^{|X|} f(X_i) - f(X_{i-1}) \\ &\ge f(\emptyset) + \sum_{i=1}^{|X|} f(G_{\rho(i)}) - f(G_{\rho(i-1)}) \quad \text{[Submodularity]} \\ &= f(\emptyset) + \sum_{e_i \in X} w_i \\ &\ge \sum_{e_i \in X} w_i \quad [f \text{ is positive}] \end{aligned}$$

We may now complete the proof. Let I_i be an indicator random variable that is 1 if e_i is in the final solution and 0 otherwise. From the arguments above, $\Pr[I_i = 1] \ge 1 - \frac{1}{e}$.

[1]Sampling without replacement is also possible, but for analysis we assume we sample with replacement.

Thus,

$$\begin{aligned}
\mathbb{E}\big[f(\cup_i S_i)\big] &\geq \mathbb{E}\left[\sum_{e_j \in \cup_i S_i} w_j I_j\right] && \text{[Above arguments]}\\
&\geq \sum_{e_j \in \cup_i S_i} w_j \mathbb{E}[I_j] && \text{[Linearity]}\\
&\geq \left(1 - \frac{1}{e}\right) \sum_{e_j \in \cup_i S_i} w_j && [\Pr[I_i = 1] \geq 1 - \tfrac{1}{e}]
\end{aligned}$$

Thus, we our solution has value within $(1 - \frac{1}{e})f(G_{nk})$. We know that $f(G_{nk}) \geq (1 - \frac{1}{e})\textsc{opt}$ by the known performance of GREEDY. Thus, we conclude the result. □

We also show that the algorithm achieves a better approximation ratio if the function is modular,[2] and that this is the best approximation guarantee possible in general.

COROLLARY 1. *Assume f is nonnegative and modular. Then this procedure has an approximation guarantee of $(1 - \frac{1}{e})$. This is the best possible.*

Proof of [Corollary 1] As before, let G be the solution produced by GREEDY$(\mathcal{V}, f, nk)$. It is not difficult to see that $f(G) = \textsc{opt}$ in the case of modularity. In this case, the distribution π as analyzed in the previous theorem implies that each element is selected with probability at least $1 - \frac{1}{e}$. Thus the expected reward is $(1 - \frac{1}{e})f(G) = (1 - \frac{1}{e})\textsc{opt}$.

To see that this is the best possible bound in general, consider the case $k = 1$, $n = m > 1$ and define the following modular set function:

$$f\big(\{e\}\big) = 1, \forall e \in \mathcal{V}.$$

The optimal value in this case is that the agents select distinct elements (covering the entire universe), giving $\textsc{opt} = n$.

Any symmetric mixed strategy for this problem is completely specified by a probability vector π over the elements of $\mathcal{V}$. Given an arbitrary selection distribution π, we may calculate the expected reward as

$$\mathbb{E}\big[f(\cup_i S_i)\big] = \sum_i 1 - (1 - \pi_i)^n = n - \sum_i (1 - \pi_i)^n.$$

From the above analysis, we note that the uniform probability distribution $\pi_i = \frac{1}{n}$ gives expected reward $(1 - \frac{1}{e})\textsc{opt} = (1 - \frac{1}{e})n$. We show that the uniform vector is in fact the optimal distribution for this problem.

Suppose there were a symmetric mixed strategy with nonuniform selection distribution π' with expected value greater than that given by π. Let i, j be any two indices of π' with $\pi'_i \neq \pi'_j$, and define $\bar{\pi}$ to be their arithmetic mean. By the strict concavity of $(1 - \pi)^n$ in π, we have

$$2(1 - \bar{\pi})^n < (1 - \pi'_i)^n + (1 - \pi'_j)^n,$$

thus we may strictly improve the expected value by replacing any nonequal probabilities in π' with their (equal) arithmetic means. This contradicts the optimality of π' and we conclude that the optimal π is uniform. □

4.2 Adapting the marginal probabilities

Although GREEDY-SAMPLING$(\mathcal{V}, f, nk)$ achieves a strong approximation ratio, it may be possible to improve on the solution obtained. In particular, note that although the agents may not communicate, they can assume that all agents are using the same strategy. Therefore given π, assuming that an agent knows the number of agents n, she may compute the expected global reward, under the assumption that all other agents operate under the same strategy:

$$\phi(\pi) = \mathbb{E}_{S_i \sim \pi}\big[f(\cup_{i=1}^n S_i)\big].$$

The agent may then maximize the expected reward as a function of the π; that is, each agent seeks

$$\pi^* = \arg\max_\pi \phi(\pi) \tag{1}$$

Finally after this maximization process, each agent samples their $S_i \sim \pi^*$.

The main question is how to find π^*. We have already shown one approach to give an approximately optimal π in GREEDY-SAMPLING$(\mathcal{V}, f, nk)$, but it is possible in some cases to do better taking the following approach. We assume that π must be a probability distribution over the elements, each value π_i must be positive and the entries must sum to 1. It can be useful to relax the latter condition and instead insist that $\sum_i \pi_i \leq 1$, the remaining mass reflecting "no element selected."

Here we introduce some notation for convenience. Consider a point x in the unit cube $[0,1]^m$. We define a function $g(x; f) \colon [0,1]^m \to \mathbb{R}$ from f in the following way. Let $S \subseteq \mathcal{V}$ be a random set, where we include element e_i in S independently with probability x_i: $\Pr(e_i \in S \mid x) = x_i$. We will abuse notation and write $S \sim x$ to indicate this generative process defined by x. We define

$$g(x; f) = \mathbb{E}_{S \sim x}\big[f(S)\big].$$

The function g, defined by taking the expected value of $f(S)$, allowing S to be generated randomly according to the probabilities in x, is called the *multilinear extension* of f. Note that if we set x to a binary vector, then the value of g is exactly equal to the set function evaluated on a corresponding set. For example $g\big([1, 0, 0, \dots]^\top; f\big) = f\big(\{e_1\}\big)$. The function g therefore "extends" the set function f from the corners of the unit cube (binary vectors, exact set function values) to the entire unit cube by allowing "fuzzy" set membership.

Consider again our multi-agent problem, and define the random variable $S = \cup_i S_i$. Given a per-agent probability vector π, we define the vector $x(\pi)$ by

$$x(\pi; n, k)_i = \Pr(e_i \in S \mid \pi, n, k) = (1 - (1 - \pi_i)^{nk}).$$

The vector $x(\pi; n, k)$ now gives the element-wise probability that each element $e \in \mathcal{V}$ is in S, the union of the agents' choices, assuming the agents each select k elements independently according to π. Note that x is in fact a function of π, n, and k, but we will ignore the dependence on n and k for brevity as all agents are assumed to know this information.

Now, given π, n, k, and f, each agent may use the multilinear extension to compute the expected global reward shared by the agents, if each were to use the vector π:

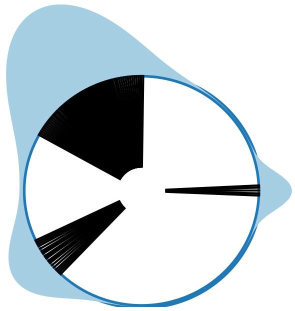
$$\phi(\pi) = \mathbb{E}_{S_i \sim \pi}\big[f(\cup_{i=1}^n S_i)\big] = \mathbb{E}_{S \sim x(\pi)}\big[f(S)\big] = g\big(x(\pi); f\big).$$

Our problem is to find

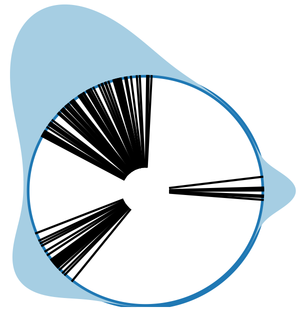
$$\arg\max_\pi \phi(\pi) = \arg\max_\pi g\big(x(\pi); f\big),$$

subject to the constraints on π. To facilitate this optimization, we may compute the gradient of our objective with

[2]A modular function satisfies $f(X) = \sum_{e \in X} f\big(\{e\}\big)$.



(a) GREEDY-nk



(b) ADAPTIVE-SAMPLING

Figure 1: Social insect log DPP example, $n = 100$, $k = 1$. A collection of agents must jointly maximize the PDF of an induced determinantal point process as a function of which direction to forage. The blue shaded region shows a probability distribution over angles, indicating the distribution of food; the black lines show selected foraging directions. The decentralized agents can provide results nearly as good as a centralized planner.

respect to π. Applying the chain rule, we have

$$\frac{\partial g((x(\pi); f)}{\partial \pi} = \frac{\partial g((x(\pi); f)}{\partial x} \frac{\partial x}{\partial \pi}.$$

The derivative of x with respect to π is simple to compute:

$$\tfrac{\partial x}{\partial \pi_i} = nk(1 - \pi_i)^{nk-1}.$$

Computing the gradient of g with respect to x is a little more complicated. By considering the definition of the multilinear extension again, and writing out the expectation as a sum over all 2^m possible sets S, we may compute the gradient as follows:

$$\tfrac{\partial g}{\partial x_i}\Big|_x = \mathbb{E}_{S\sim x}\Big[f\big(S \cup \{e_i\}\big) - f\big(f \setminus \{e_i\}\big)\Big].$$

Define the following abuses of notation:

$$S + i = S \cup \{e_i\}; \qquad S - i = S \setminus \{e_i\};$$

then,

$$\tfrac{\partial g}{\partial x_i}\Big|_x = \mathbb{E}_{S\sim x}\big[f(S + i) - f(S - i)\big].$$

Note that the sets $S+i$ and $S-i$ differ by exactly one element. We may interpret this gradient as giving the instantaneous benefit from increasing each entry in the element-wise probability vector x.

Unfortunately, computing this gradient in closed form requires summing over all 2^m possible values of S. However, we may make a Monte Carlo estimate of the gradient by repeatedly sampling S element-wise according to x and averaging the contributions above. Theoretically, a polynomial number of samples are needed to estimate the gradient with high probability [10]. Throughout this text, we refer to this strategy as ADAPTIVE-SAMPLING.

5. DEMONSTRATIONS

In this section, we show how we can apply the proposed algorithms to solve different tasks: foraging in insect colonies, sensor placement to detect contamination, and maximum coverage of Wikipedia pages. Our goal is to provide intuition about the algorithms' choices; we leave the quantitative results for the next section.

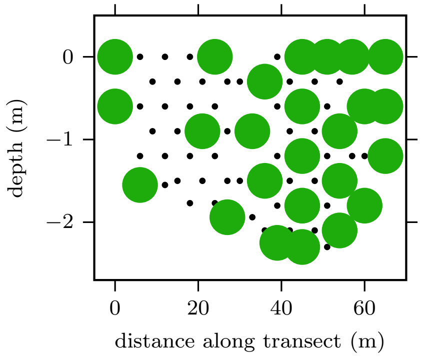
5.1 Simulation of social insects

We first demonstrate our algorithm with a simulation of social insects. Imagine that a collection of $n = 100$ insects (we will call them bees) have a shared belief about in which directions away from the hive food can be found. In bees, this knowledge is shared and communicated via the so-called "waggle dance." We consider a collective optimization problem wherein each bee must now independently select a direction to forage in. There is a natural incentive for the bees to induce diversity in their foraging coverage to maximize total food collected and also to potentially locate new sources of food to be communicated to the hive, improving the shared knowledge. We model the quality vs. diversity trade off in foraging directions via a set function that is the log probability of an associated determinantal point process (DPP) over sets of angles [19]. This function is both submodular and also encourages simultaneous quality and diversity in the chosen directions. Note the function is not monotone, but we may nonetheless run our algorithms in this setting. Figure 2(a) shows the directions that would be allocated to the bees using an inaccessible centralized greedy plan. Figure 2(b) shows sample directions chosen by the agents using the ADAPTIVE-SAMPLING algorithm. Without planning the bees independently select foraging directions that benefit the collective.

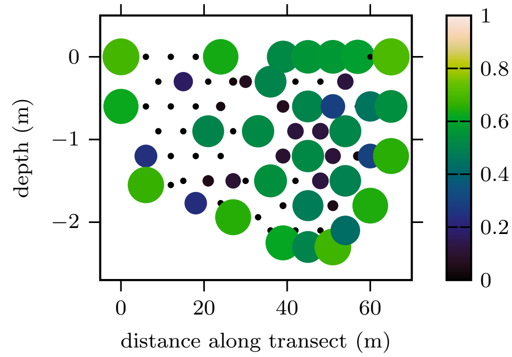
5.2 Environmental monitoring

We consider decentralized aquatic robots monitoring water quality. We use a dataset representing water pH measured at $m = 86$ discrete locations along a ~ 60 m transect of Lake Merced near San Fransisco. Previous work has considered the problem of estimating pH along this transect using a Gaussian process (GP) estimator. A custom Gaussian process model was built from measurements in [34]; we use this model here. Given a set of measurements, the information gained about the entire random function in the GP setting may be computed in closed form as the entropy of a multivariate Gaussian distribution. Further, considering the set of discrete sampling locations to be a universe $\mathcal{V}$ and defining the set function $f(S)$ to be the information gain associated with measuring the function at the locations in S, we have that f is nonnegative, monotone, and submodular [18].

We illustrate our algorithm in action in this setting using the information gain value function f, and setting $n = k =$



(a) Greedy-Sampling. Expected value: 7.2. Expected number of elements selected: 15.8



(b) Adaptive-Sampling. Expected value: 7.4. Expected number of elements selected: 17.6.

Figure 2: Lake Merced transect Gaussian process entropy maximization example, $n = k = 5$. The marginal probability of including each element in the union is indicated by size and color.

5. Figure 2(a) shows the sensors selected by running our Greedy-Sampling algorithm on this problem. The size and color of the circles correspond to the marginal probability that each sensing location is selected by the collective using the marginal selection probability vector π that each algorithm returns. We see that the agents choose randomly from a set of locations that cover the sensing region well. Notice that the pH function showed significant nonstationarity in the region corresponding to 30–60 m along the transect, hence the denser sampling in that region. In Figure 2(b), we see what happens if we allow these marginal probabilities to be optimized by performing gradient ascent on the multilinear extension of f. We see that the solution returned by our Adaptive-Sampling algorithm is similar to the solution returned by Greedy-Sampling, although the selection vector π now has support on more than nk elements. Intuitively, allowing the probabilities to be optimized shifted some probability mass from the greedy solution to points that are "nearly as good," especially in the region 30–60 m along the transect. The result of this is an increase in the expected shared value (information gain of 7.4 nats versus 7.2 nats for Greedy-Sampling), and a larger expected cardinality of the union (17.6 versus $15.8 \approx nk(1 - \frac{1}{e})$).

5.3 Wikipedia maximum coverage

Table 2: Marginal probabilities found on some Wikipedia Country pages

Wikipedia page	π_{GS}	π_{AS}	# links
British Empire	0.025	0.041	133
Axis Powers	0.025	0.035	106
Russian SSR	0.025	0.025	56
Confederation of the Rhine	0.025	0.024	53
Chu (state)	0.025	0.024	27
Somalia	0.025	0.024	62
World government	0	0.007	84
Russia	0	0	60
European Union	0	0	52

The maximum coverage problem is a fundamental algorithmic problem widely used in several applications [4, 13, 37]. In social network analysis, for instance, this problem could be used to find the most influential individuals of a particular group [27]. This is an interesting question to marketing campaign designers who want to focus on particular users and reach a target audience. Given several sets $S_1, S_2, \ldots, S_m$ and a number k, $k < m$, the maximum coverage problem asks to find k sets that contain the largest number of different items, i.e., the union of the selected sets has maximal size. Mapping this problem to our framework is simple; we just need to define the set function as the cardinality function $f(A) = |A|$, which is submodular.

In our experiments, we studied the coverage of Wikipedia pages, because it represents a rich and free available source of data. Specifically, we used the DBpedia[3] knowledge base to access structured content from Wikipedia. From the 2014 DBpedia dataset, we constructed a graph whose nodes are Wikipedia pages and whose edges represent links between pages. In this section, we restricted the domain of pages to those which are in the category of COUNTRY, according to the DBpedia ontology. We pose the following question: which Wikipedia pages from the class COUNTRY can cover the most number of pages of the same class with its set of outgoing links?

We fixed the number of agents to $n = 8$ and let each agent select up to $k = 5$ items. We then ran both the Greedy-Sampling and Adaptive-Sampling strategies, computing the expected reward using a total of 10 000 samples from the multilinear extension. The expected reward for the set returned by Greedy-Sampling was 824, whereas the expected reward for the set returned by Adaptive-Sampling was 878. This shows that Adaptive-Sampling succeeded in improving upon Greedy-Sampling. We further explore this result by taking a closer look to the selection made by these algorithms.

Table 2 presents several Wikipedia pages from the class COUNTRY. The first group of pages are ordered according to their marginal probability of inclusion in the union $\cup S_i$ found by the Adaptive-Sampling strategy π_{AS}. For Greedy-Sampling, some pages have marginal probability of $\frac{1}{nk}$, and others have probability zero; this defines the π_{GS} distribution. The second group is ranked by their number of outgoing links.

The values on π_{AS} assigned to these pages exemplify great

[3] http://wiki.dbpedia.org/

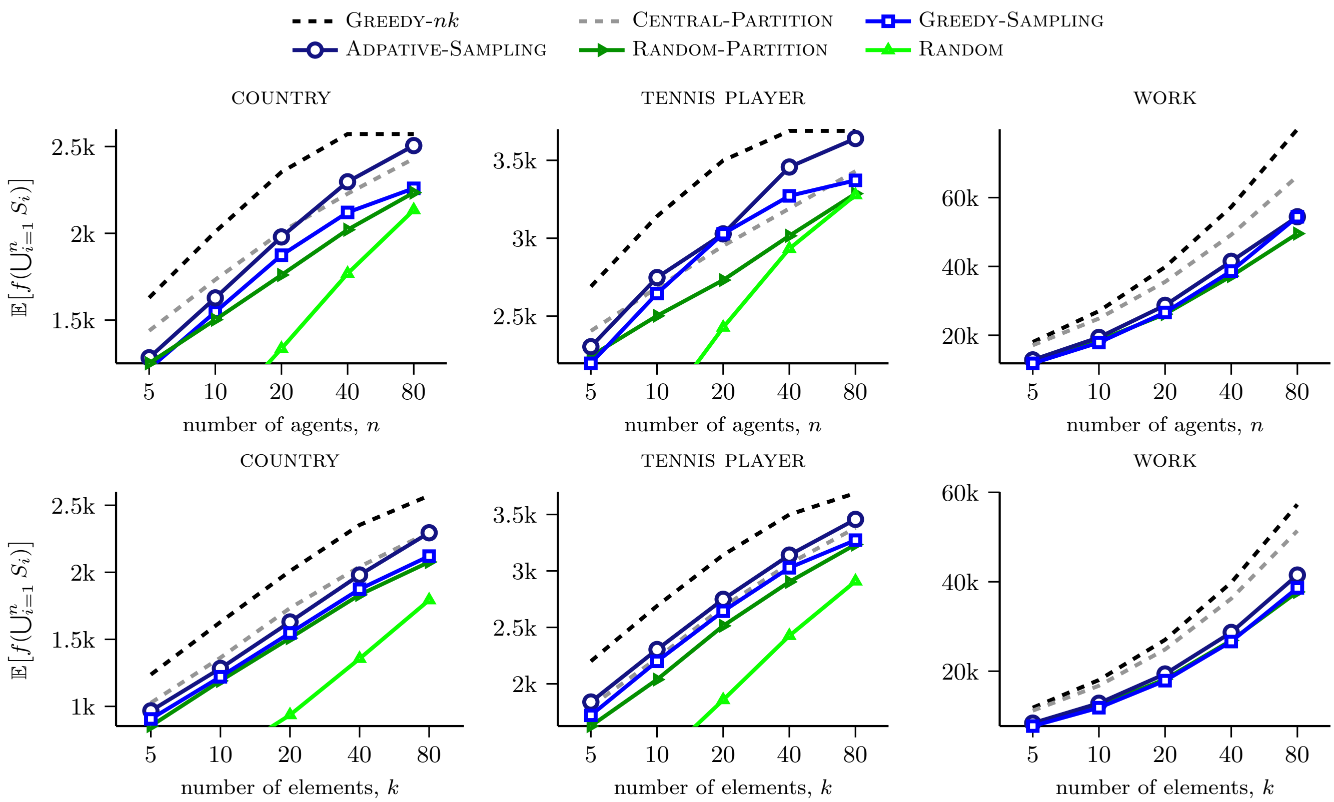


Figure 3: Performance of all methods on the datasets country, tennis player and work. The first row shows results for increasing number of agents n with a fixed number of elements $k = 20$; the second row shows results for a fixed number of agents $n = 10$ and increasing k. The vertical axis shows the expected reward in thousands. For better visualization we deliberately omit some results for the baseline Random since its performance was much worst than the other methods.

Table 1: Comparison of the expected reward for multiple instances. The number of elements is fixed at $k = 20$. The results are normalized against Greedy-nk for that particular instance. The displayed methods are Greedy-nk (G-nk), Central-Partition (cp), Adaptive-Sampling (as), Greedy-Sampling (gs), Random-Partition (rp) and Random. The best performance among the decentralized methods are displayed in bold.

Dataset	n	centralized		decentralized			
		G-nk	CP	AS	GS	RP	Random
book	5	1.00	0.93 (0.01)	**0.75**	0.66	**0.68 (0.08)**	0.15 (0.01)
	20	1.00	0.82 (0.00)	**0.73**	0.66	0.61 (0.02)	0.26 (0.00)
	80	1.00	0.75 (0.00)	**0.74**	0.67	0.58 (0.01)	0.44 (0.02)
country	5	1.00	0.88 (0.01)	**0.79**	0.75	**0.77 (0.03)**	0.37 (0.04)
	20	1.00	0.85 (0.00)	**0.84**	0.80	0.75 (0.02)	0.57 (0.01)
	80	1.00	0.94 (0.00)	**0.97**	0.88	0.87 (0.01)	0.83 (0.01)
game	5	1.00	0.89 (0.01)	**0.73**	0.69	**0.72 (0.02)**	0.38 (0.03)
	20	1.00	0.86 (0.00)	**0.83**	0.72	0.70 (0.03)	0.59 (0.02)
	80	1.00	1.00 (0.00)	**0.96**	0.80	0.80 (0.02)	0.93 (0.01)
tennis player	5	1.00	0.89 (0.01)	**0.86**	0.82	0.83 (0.02)	0.48 (0.02)
	20	1.00	0.84 (0.00)	**0.87**	**0.87**	0.78 (0.01)	0.69 (0.02)
	80	1.00	0.93 (0.00)	**0.99**	0.91	0.89 (0.00)	0.89 (0.00)
species	5	1.00	0.98 (0.00)	**0.69**	0.64	**0.70 (0.06)**	0.01 (0.00)
	20	1.00	0.95 (0.00)	**0.70**	0.64	0.64 (0.01)	0.03 (0.00)
	80	1.00	0.92 (0.00)	**0.72**	0.64	0.63 (0.02)	0.05 (0.00)
soccer player	5	1.00	0.90 (0.01)	**0.73**	0.67	**0.70 (0.03)**	0.13 (0.01)
	20	1.00	0.83 (0.00)	**0.74**	0.68	0.64 (0.01)	0.22 (0.01)
	80	1.00	0.78 (0.00)	**0.74**	0.69	0.61 (0.01)	0.36 (0.01)
work	5	1.00	0.95 (0.01)	**0.71**	0.66	**0.71 (0.03)**	0.03 (0.00)
	20	1.00	0.89 (0.00)	**0.72**	0.67	0.66 (0.02)	0.05 (0.00)
	80	1.00	0.83 (0.00)	**0.68**	**0.68**	0.62 (0.01)	0.09 (0.01)

submodularity. The pages in the second group of rows in Table 2 all cover more pages than some pages in the first group, but have trivial inclusion probability. On the other hand, pages such as Chu State, which has less than 30 links and Somalia have links to certain pages that are only reachable from these pages. Somalia, for example, links to Majeerteen Sultanate, Sultanate of Hobyo, Galmudug, and Ajuran Empire, among others. These pages would be difficult to otherwise cover. ADAPTIVE-SAMPLING also boosts the probabilities of some pages presumably considered more relevant to the maximum coverage (such as British Empire and Axis Powers). By adapting the distribution π to increase the chances of selecting valuable elements, ADAPTIVE-SAMPLING surpasses the performance of GREEDY-SAMPLING.

6. EXPERIMENTS

We provide quantitative results for the performance of both proposed algorithms, compared to several baselines. We focused on the Wikipedia maximum coverage setting illustrated in Section 5.3, using subgraphs corresponding to a larger range of ontology classes with a range of properties. We also consider four additional benchmark methods and compare the algorithms' performance.

Datasets. Besides COUNTRY (see Subsection 5.3), we considered five more graphs created from Wikipedia: BOOK, GAME, TENNIS PLAYER, SPECIES, SOCCER PLAYER, and WORK. They have a wide range of sizes, densities, and numbers of connected components (see Table 3).

Table 3: Some statistics for the datasets extracted from Wikipedia. "k" indicates thousands.

dataset	# nodes	# edges	mean degree
BOOK	31k	37k	1.24
COUNTRY	3k	31k	10.88
GAME	1k	2k	1.64
TENNIS PLAYER	4k	81k	20.29
SPECIES	252k	1644k	6.85
SOCCER PLAYER	96k	64k	0.67
WORK	411k	1819k	4.50

Methods. In order to evaluate the performance of the proposed algorithms GREEDY-SAMPLING and ADAPTIVE-SAMPLING, we implemented several benchmarks:

1. GREEDY-nk: runs GREEDY$(\mathcal{V}, f, nk)$;
2. CENTRAL-PARTITION: divides the elements of $\mathcal{V}$ into n disjoint parts $\mathcal{V}_i \subset \mathcal{V}$ and runs GREEDY$(\mathcal{V}_i, f, k)$ on each partition
3. RANDOM-PARTITION: randomly constructs a set $\mathcal{V}'_i \subset \mathcal{V}$ with $\frac{|\mathcal{V}|}{n}$ elements, and runs GREEDY$(\mathcal{V}'_i, f, k)$
4. RANDOM: randomly selects k items from $\mathcal{V}$ for each agent.

Recall that the first two strategies assume that the agents can either communicate or are not indistinguishable. Further, CENTRAL-PARTITION is the same strategy proposed by [24] in the context of distributed submodular maximization.

Implementation details. We will release reusable code in conjunction with this manuscript to ease implementation, but here we highlight some important aspects. For implementing GREEDY we considered the classical accelerated greedy [23] (or lazy greedy) algorithm, which allows a faster computation. ADAPTIVE-SAMPLING was implemented using an off-the-shelf projected quasi-Newton optimizer to optimize the objective on the simplex of valid probability distributions [31]. We started the optimization from the π vector provided by GREEDY-SAMPLING$(\mathcal{V}, f, nk)$.

Results. For the methods GREEDY-SAMPLING, ADAPTIVE-SAMPLING, we computed the expected reward considering a total of 10 000 samples. For the others randomized algorithms we repeated the experiments five times. The standard error across repetitions is too small to be seen on this graphical representation. We also present more results in Table 1, considering all datasets, for varying n. We normalize the results by dividing by the GREEDY-nk value of that particular instance (an upper bound on our performance).

Discussion. As expected, GREEDY-nk outperforms all other methods since it is the only method to always return exactly nk elements. It, of course, blatantly violates the constraints of the problem. However, the interesting result is that, in some cases, one or both of GREEDY-SAMPLING and ADAPTIVE-SAMPLING outperforms the centralized algorithm CENTRAL-PARTITION. In particular, for TENNIS PLAYER, GREEDY-SAMPLING is comparable to CENTRAL-PARTITION; further, ADAPTIVE-SAMPLING performs better than the centralized approach for most of the considered values of n and k. A possible explanation is that by partitioning the elements of $\mathcal{V}$, rather than reasoning about all of them jointly, the agents can choose items that locally are valuable but when combined with the other agent's outputs become worthless. Of the decentralized methods, ADAPTIVE-SAMPLING shows the best performance. For small values of k and n, it is only slightly better (sometimes tied), but as these values increase, it outperforms the other methods by up to 10 percent.

7. CONCLUSION

We introduced a novel framework for cooperative set function optimization in which the agents cannot communicate and we presented two strategies for solving this problem: GREEDY-SAMPLING and ADAPTIVE-SAMPLING. For GREEDY-SAMPLING we give an approximation bound of $(1 - \frac{1}{e})^2$; we then demonstrated how we can improve upon this strategy by using ADAPTIVE-SAMPLING. We showed how these algorithms could be used by agents in natural systems to perform effective decentralised, collective foraging tasks. Then, we provided an empirical evaluation of these techniques, comparing their performances with other benchmark algorithms. In particular, we even compare our methods with strategies that violate the problem's constraints. ADAPTIVE-SAMPLING was shown to outperform all benchmarks which do not assume communication among the agents and, in some cases, it was even comparable to strategies that allow prior coordination.

8. ACKNOWLEDGMENTS

GM and RG were supported by the National Science Foundation (NSF) under award number IIA-1355406. Additionally, GM was also supported by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). BH was supported by a Washington University Summer Engineering Fellowship (WUSEF). BM was supported in part by a Google Research Award, a Yahoo Research Award, and by the NSF under award number CCF-1617724.

REFERENCES

[1] S. Barrett and P. Stone. An analysis framework for ad hoc teamwork tasks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 357–364, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[2] S. Barrett, P. Stone, S. Kraus, and A. Rosenfeld. Teamwork with limited knowledge of teammates. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 102–108. AAAI Press, 2013.

[3] A. Caiti, V. Calabro, F. Di Corato, T. Fabbri, D. Fenucci, A. Munafo, B. Allotta, F. Bartolini, R. Costanzi, J. Gelli, et al. Thesaurus: AUV teams for archaeological search. Field results on acoustic communication and localization with the Typhoon. In *Mediterranean Conference on Control and Automation (MED)*, 2014.

[4] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *APPROX-RANDOM*, 2004.

[5] R. da Ponte Barbosa, A. Ene, H. L. Nguyen, and J. Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning (ICML)*, 2015.

[6] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *ACM–SIAM symposium on Discrete Algorithms*, 2006.

[7] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[8] U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

[9] U. Feige and J. Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010.

[10] M. Feldman, J. Naor, and R. Schwartz. A unified continuous greedy algorithm for submodular maximization. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2011.

[11] M. Fenster, S. Kraus, and J. S. Rosenschein. Readings in agents. chapter Coordination Without Communication: Experimental Validation of Focal Point Techniques, pages 380–386. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[12] D. M. Gordon. The organization of work in social insect colonies. *Nature*, 380(6570):121–124, 1996.

[13] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[14] M. Knudson and K. Tumer. Robot coordination with ad-hoc team formation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1441–1442, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[15] H. Konishi and D. Ray. Coalition formation as a dynamic process. *Journal of Economic Theory*, 110(1):1 – 41, 2003.

[16] S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 1–8, New York, NY, USA, 2003. ACM.

[17] A. Krause and D. Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014.

[18] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[19] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.

[20] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2010.

[21] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM conference on Electronic Commerce (EC)*, 2001.

[22] P. Michiardi and R. Molva. Analysis of coalition formation and cooperation strategies in mobile ad hoc networks. *Ad Hoc Networks*, 3:2005, 2005.

[23] M. Minoux. *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, chapter Accelerated greedy algorithms for maximizing submodular set functions, pages 234–243. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[24] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Conference on Neural Information Processing Systems (NIPS)*, 2013.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1):265–294, 1978.

[26] P. Ögren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[27] P. Reyes and A. Silva. Maximum coverage and maximum connected covering in social networks with partial topology information. In *Computer Communications Workshops*, pages 825–830, 2014.

[28] I. Rochlin, D. Sarne, and M. Mash. Joint search with self-interested agents and the failure of cooperation enhancers. *Artif. Intell.*, 214:45–65, 2014.

[29] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1):209 – 238, 1999.

[30] D. Sarne and S. Kraus. Cooperative exploration in the electronic marketplace. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2005.

[31] M. W. Schmidt, E. van den Berg, M. P. Friedlander, and K. P. Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Conference on Statistics in Artificial Intelligence (AISTATS)*, 2009.

[32] T. D. Seeley. *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press, 2009.

[33] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1):165–200, 1998.

[34] M. J. Stealey, A. Singh, M. A. Batalin, B. Jordan, and W. J. Kaiser. NIMS-AQ: A novel system for autonomous sensing of aquatic environments. In *International Conference on Robotics and Automation (ICRA)*, 2008.

[35] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[36] D. J. Sumpter. *Collective animal behavior*. Princeton University Press, 2010.

[37] H. Takamura and M. Okumura. Text summarization model based on maximum coverage problem and its variant. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 781–789, 2009.

[38] M. Tsvetovat, K. Sycara, Y. Chen, and J. Ying. Customer coalitions in electronic markets. In *Agent-Mediated Electronic Commerce III*, pages 121–138. Springer, 2000.

[39] F. Wu, S. Zilberstein, and X. Chen. Online planning for ad hoc autonomous agent teams. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, pages 439–445. AAAI Press, 2011.

[40] J. Yamamoto and K. Sycara. A stable and efficient buyer coalition formation scheme for e-marketplaces. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 576–583, New York, NY, USA, 2001. ACM.