Task Trading for Crowdsourcing in Opportunistic Mobile Social Networks

Xiao Chen
Department of Computer Science, Texas State University, San Marcos, TX 78666
Email: xc10@txstate.edu

***Abstract*—With the explosive proliferation of mobile devices, *mobile crowdsourcing* has become a new paradigm involving a crowd of mobile users to collectively take large-scale tasks from requesters in mobile social networks (MSNs). In this paper, we study task allocation in crowdsourcing in Opportunistic Mobile Social Networks (OMSNs) which are formed opportunistically when people gather together at social events. Specifically, we aim to minimize the total working hours of the users to finish these tasks. Different from other algorithms, we hope to raise the efficiency of the whole network by task trading inspired by the comparative advantage in macroeconomy. We first prove that our defined problem is NP-hard and then propose a heuristic task trading algorithm TTA by which users can trade when they meet opportunistically. Simulation results comparing our proposed algorithm with the one without considering trading and the brute force algorithm to find the minimum total number of hours show that our proposed algorithm can substantially reduce the total number of hours to finish all the allocated tasks and is very close to the benchmark brute force algorithm.**

***Index Terms*—comparative advantage, crowdsourcing, opportunistic mobile social networks, trading**

I. Introduction

With the explosive population of mobile devices, *mobile crowdsourcing* has become a new paradigm involving a crowd of mobile users to collectively take large-scale tasks from requesters in mobile social networks (MSNs) [11]. It has stimulated a lot of research, including platform design [5], [13], user recruitment [10], [12], [14], task allocation algorithms [6], [18], [20], [22], and so on. Among them, user recruitment and task allocation are among the most important topics [6], [12], [14], [18], [20], [22] and are similar. User recruitment focuses more on the incentives to get the users while task allocation focuses more on distributing tasks to the users. In this paper, we work on task allocation. Furthermore, we discuss task allocation in a special kind of mobile social network which is formed opportunistically when people gather together at conferences, social events, campus activities, etc. We refer to it as *Opportunistic Mobile Social Network* (OMSN).

As we know, the majority of the current crowdsourcing implementation relies on centralized registries to recruit possible participants [16] and assumes the use of cellular networks to distribute tasks [14]. We discuss task allocation in a distributed and opportunistic environment such as OMSN because it is a promising solution [4] to alleviate the intense pressure on the existing cellular infrastructure caused by the soon-to-be Zettabytes of annual global IP traffic generated mostly from the mobile devices [2]. Opportunistic Mobile Social Networks such as Device-to-Device (D2D) communication network [15] allow direct communication between two mobile devices [9] without traversing the core cellular network. Under this trend, we plan to study task allocation in OMSNs.

More specifically, the goal of our task allocation problem is how to minimize the total number of working hours to finish all the tasks allocated to the users. Achieving this goal allows the users to obtain more tasks to earn more money. Many task allocation algorithms proposed for mobile crowdsourcing [6], [12], [14], [18], [20], [22] address the efficiency issue. However, to our best knowledge, they have one common feature. That is, once the tasks are allocated to the users, the users will work on their own without exchanging tasks with others. We believe the working efficiency of the users can be improved if they trade tasks with others when they meet opportunistically. Our idea is backed up by the trading law in macroeconomics proposed by economist David Ricardo [17]: it is beneficial for a country to trade with others as long as it has *comparative advantage* in producing some goods. A country has a comparative advantage at producing something if it can produce the product at lower cost than others. In our context, a user has comparative advantage in some task if he can finish it faster than anyone else. We first use an example to show how trading can reduce the total number of working hours and then come up with trade conditions when two users meet. We prove that finding the minimum total number of working hours for a set of allocated tasks in OMSNs is NP-hard. And then we propose a heuristic *task-trade algorithm* (TTA) based on comparative advantage to solve the problem. We evaluate the performance of our algorithm by comparing it with the *no-trade algorithm* (NTA) and the *brute-force algorithm* (BFA) to find the minimum total number of working hours in a brute force way using two online traces. Simulation results show that our proposed TTA algorithm can substantially reduce the total number of working hours compared with the one without trading by being very close to the benchmark brute force algorithm.

The differences of our work from others and the key contributions of our work are as follows:

- We define a task allocation problem in a distributed and opportunistic environment OMSN and prove it NP-hard.
- We propose a heuristic task trading algorithm TTA based on comparative advantage.
- We conduct simulations to compare TTA with the one

without trading NTA and the brute force algorithm BFA. Simulation results show that TTA substantially outperforms NTA by being very close to BFA.

The rest of the paper is organized as follows: Section II references the related works; Section III defines the problem we want to solve; Section IV presents our solution to the problem; Section V describes the simulations comparing our solution with its variations; and Section VI is the conclusion.

II. Related Works

There has been a lot of research on user recruitment or task allocation problems in mobile crowdsourcing [6], [10], [12], [14], [18], [20], [22]. The various algorithms proposed by the majority of these works assume a centralized registry and use the cellular network resources for communication. For example, M. Cheung et al. in [6] consider a mobile crowdsourcing platform that posts task information and propose an algorithm to help the users plan their task selections on their own. Z. He et al. in [12] propose a participant recruitment strategy for vehicle-based crowdsourcing based on predicted vehicle trajectory. S. He et al. in [18] study the problem of allocating location dependent tasks and design a local ratio based algorithm to solve it. M. Xiao et al. in [20] and Q. Zhao et al. in [22] study the task allocation issues by formulating them as online scheduling problems.

Not many papers in the literature discuss distributed task allocation over opportunistic networks. We are aware of two papers under this thread. M. Karaliopoulos et al. [14] study user recruitment over opportunistic networks. Their focus is on how the data collected by the users are transferred over the opportunistic networks to the campaign organizer. G. S. Tuncay et al. [10] consider both user recruitment and data collection using the opportunistic network. The goal of their recruitment protocols is to find users who can cover the sensing areas based on their previous geographical locations.

So far as we know, none of the above works considers task trading which we believe can make the whole crowdsourcing system more efficient. And this is what we will work on next.

III. Problem Definition

In this section, we first show the benefit of trading and then define our problem.

A. Benefits of Trading

First, we use an example, which we refer to as example 1, to explain the benefits of trading. Suppose there are two mobile users A and B and two types of tasks T_1 and T_2. The units of the two tasks obtained by A and B from the requester are lists in Table I. And the number of hours for each one of them to finish each task is shown in Table II. Here, A has got 300 units of T_1 and 400 units of T_2 and B has got 500 units of T_1 and 100 units of T_2. For simplicity, we use a *task vector* in the form of (# of T_1, # of T_2, $\cdots$, # of T_n) to represent the units of the tasks obtained by a user. So, A's task vector is (300, 400) and B's task vector is (500, 100). Similarly, we define an *efficiency vector* as (hours to finish T_1, hours to finish T_2, $\cdots$, hours to finish T_n) to represent the number of hours to finish each task by a user. So A's efficiency vector is (1, 2), meaning it takes A one hour to finish T_1 and two hours to finish T_2. Accordingly, B's efficiency vector is (3, 4).

	A	B
T1	300	500
T2	400	100

TABLE I
Tasks A and B have obtained

	H_A	H_B
T1	1	3
T2	2	4

TABLE II
The number of hours for A and B to finish each task

If they do not trade, it takes A $300\times1+200\times2 = 700$ hours and B $500 \times 3 + 100 \times 4 = 1900$ hours to finish their tasks, respectively. The total number of working hours is 2600. Now let us look at the situation if they trade. For example, A takes 100 T_1 from B. In return, B will take 75 T_2 from A because 100 T_1 is worth 75 T_2 in B due to the fact that the number of hours to finish T_1 and T_2 is four and three in B. After trading, A's task vector becomes (400, 125) and B's vector becomes (400, 175). Then their respective working hours will be $400\times1+125\times2 = 625$ and $400\times3+175\times4 = 1900$. Here, B's working hours remains the same but A's working hours is reduced. Therefore, the total number of working hours has been decreased from 2600 to 2525. Though B does not benefit from this trading, it may benefit from the trading with other users. From trading, we can see that the total number of hours for A and B to finish their tasks can be reduced. Inspired by this example, we have the following problem formulation.

B. Problem Definition

We consider a mobile crowdsourcing environment OMSN which has a group of users and m types of tasks $\{T_1, T_2, \cdots, T_m\}$. Each user has a task vector (# of T_1, # of T_2, $\cdots$, # of T_m) that represents the units of the tasks he has obtained from the requester and an efficiency vector (hours to finish T_1, hours to finish T_2, $\cdots$, hours to finish T_m) that describes the number of hours he can finish each task. When two users meet opportunistically, they can exchange their vectors and trade if possible. Our goal is to minimize the total number of working hours for the users to finish these tasks by taking advantage of task trading.

IV. Our Solution

In this section, we provide answer to our defined problem. The solution has two parts. The first part is to answer whether or not two users should trade, and the second part is how to trade to minimize the total working hours. In how to trade, we first discuss the trade conditions for two tasks and then propose a task-trade algorithm TTA to trade multiple tasks.

A. Trade or not

In reality, many capable people can finish tasks faster than others. Intuitively they will not benefit from trading. But this is not true. In example 1, A is more efficient than B by finishing each task faster. In other words, A has *absolute advantage* over B in every task. But after a close look, A can reduce its working hours by trading. The explanation lies in the *comparative advantage*, one of the most powerful yet counter-intuitive insights developed by David Ricardo [17] in macroeconomics. More explicitly, two users can trade as long as each one has comparative advantage over the other in some task. In example 1, A has comparative advantage over B in T_1 because A can finish one T_1 in $1/2 = 0.5$ of the time to finish T_2 and B can finish one T_1 in $3/4 = 0.75$ of the time to finish T_2. We can simply say that one T_1 is worth 0.5 T_2 in A and 0.75 T_2 in B, respectively. Similarly, B has comparative advantage over A in T_2 because one T_2 is worth $4/3\, T_1 = 1.33$ T_1 in B and $2/1\; T_1 = 2\; T_1$ in A. Now we have the following theorem regarding comparative advantage.

Theorem 1. *Assume there are two users A and B and two tasks T_1 and T_2. User A's efficiency vector is (p, q) meaning that it takes A p hours to finish one unit of T_1 and q hours to finish one unit of T_2, respectively and B's efficiency vector is (s, r). Parameters $p, q, s, r > 0$. If user A has comparative advantage over B in T_1, that is, $\frac{p}{q} \le \frac{s}{r}$, then B must have comparative advantage over A in T_2.*

Proof. From the given conditions, B can finish one T_2 in $\frac{r}{s}$ of the time to finish T_1 and A can finish one T_2 in $\frac{q}{p}$ of the time to finish T_1. Since $\frac{p}{q} \le \frac{s}{r}$, then $\frac{r}{s} \le \frac{q}{p}$ must be true. Therefore, B has comparative advantage over A in T_2. □

From Theorem (1), we conclude that two users should trade as long as one user has a comparative advantage in some task. The only situation that they do not need to trade is when both users are equally efficient in the two tasks. That is, $p = s$ and $q = r$. In this special case, you can still trade but it makes no difference. In the general case, each user carries multiple tasks. Before trading, they should first identify the common tasks and then find out each other's efficiency in these tasks. In the following two subsections, we discuss how to trade to minimize the total working hours. In the first subsection, we look for the trade conditions for two tasks. And in the second subsection, we propose an algorithm to trade multiple tasks.

B. Trade with two tasks

In this subsection, we work on the trade conditions to trade two tasks. We first use a concrete example to provide us with some insights and then formalize the conditions for two users to trade two tasks to minimize the total working hours.

1) Trade conditions in an example: Let us take a look at example 1 again. There are two cases here: user A initiates trading or user B initiates trading.

We first look at the case of A initiating trading shown in Table III. The table presents A's tasks and its time H_A to finish them, B's tasks and its time H_B to finish them, and the total number of hours H_{AB} for both A and B to finish these tasks. In H_{AB}, A is put before B to indicate that it initiates the trading. The top rows of A and B show the original tasks obtained from the requester. That is, A has a task vector of (300, 400) and B has a task vector of (500, 100). If they do not trade, the number of hours for A and for B to finish these tasks are $300 \times 1 + 400 \times 2 = 1100$ and $500 \times 3 + 100 \times 4 = 1900$, respectively. And the total number of hours H_{AB} is 3000. Now suppose A initiates trading and since it has comparative advantage in T_1 and B has comparative advantage in T_2, A should take some units of T_1 from B and B should take some units of T_2 from A in return. If A takes 100 T_1 from B, then B should take $100 * 3/4 = 75$ T_2 from A. Then shown in row 2, A's task vector becomes (400, 325) and B's task vector becomes (400, 175). In this case, H_A is reduced to 1050 and H_{AB} is reduced to 2950. After that, A can continue taking more units of T_1 from B and B can continue taking the corresponding units of T_2 from A based on their efficiency vectors. We have several observations from Table III. First, with A taking more units of T_1 from B, A's working hours H_A and H_{AB} are decreasing. Second, the maximum units of T_1 A can take is bounded by whichever the total units of T_1 obtained by B and the total units of T_2 obtained by A goes to zero first after trading. In this example, the units of T_1 in B goes to zero first. This is the termination point of the trading initiated by A. So A can take a maximum of 500 T_1 from B and reduces H_A to 850 and thereafter reduces H_{AB} to 2750.

T_1	T_2	H_A
300	400	1100
400	325	1050
500	250	1000
600	175	950
700	100	900
800	25	850

A

T_1	T_2	H_B
500	100	1900
400	175	1900
300	250	1900
200	325	1900
100	400	1900
0	475	1900

B

H_{AB}
3000
2950
2900
2850
2800
2750

AB

TABLE III
A INITIATES TRADING AND TAKES T_1 FROM B

T_1	T_2	H_A
300	400	1100
450	325	1100
600	250	1100
750	175	1100
800	**150**	**1100**

A

T_1	T_2	H_B
500	100	1900
350	175	1750
200	250	1600
50	325	1450
0	**350**	**1400**

B

H_{BA}
3000
2850
2700
2550
2500

BA

TABLE IV
B INITIATES TRADING AND TAKES T_2 FROM A

Now we look at the case of B initiating the trading shown in Table IV. The table shows the similar content. B is put before A in H_{BA} to indicate that B initiates the trading. The first rows of A and B show their initial task vectors. In row 2, B takes 75 T_2 from A and in return A takes $75*2/1 = 150$ T_1 from B. Thus, A's task vector becomes (450, 325) and B's task vector becomes (350, 175). Now B is able to reduce H_B to 1750 and H_{BA} to 2850. Then B can continue taking T_2 from A and A can continue taking the corresponding units of T_1 from B until either B is running out of T_1 or A is running out of T_2. In this case, B runs out of T_1 first again. At the termination point, the total number of working hours H_{BA} is 2500, which is less than H_{AB} in A initiating the trading. In this example, this is the best trade condition. That is, B initiates trading and takes 250 T_2 from A and A takes 500 T_1 in return from B to reach a minimum of 2500 working hours.

From this example, we can get some insights for the trade conditions to minimize the total working hours. First, whoever initiates trading can reduce its working hours. Second, an initiator can achieve the minimum working hours when he takes the maximum units of the task he has comparative advantage in from the other user. Third, for the initiator, the maximum units of the task he can take (the termination condition) depends on the original units of tasks obtained by the users and their efficiency vectors. Fourth, who should be the initiator depends on whoever can achieve the minimum total number of hours. And finally, each user is motivated to trade even if he cannot benefit from one trade because he can benefit from another trade with another user. The proofs of these conclusions are straightforward. So we skip them to save space. Next, we formalize the trade conditions.

2) Formalize trade conditions: Assume there are two users A and B and two tasks T_1 and T_2. Users A and B have efficiency vectors (p, q) and (s, r), respectively. Suppose A has comparative advantage in T_1 over B and B has comparative advantage in T_2 over A. So, $\frac{p}{q} \leq \frac{s}{r}$. Initially, their task vectors are (t_{A1}, t_{A2}) and (t_{B1}, t_{B2}), respectively.

If A initiates trading and takes x units of T_1 from B, B will take $\frac{xs}{r}$ units of T_2 from A in return. Then, their task vectors become:

$$A : (t_{A1} + x, t_{A2} - \frac{xs}{r}) \quad B : (t_{B1} - x, t_{B2} + \frac{xs}{r})$$

And the total number of hours to finish these tasks is:

$$H_{AB} = (t_{A1}+x)p + (t_{A2} - \frac{xs}{r})q + (t_{B1} - x)s + (t_{B2} + \frac{xs}{r})r$$

If B initiates the trading and takes y units of T_2 from A, A will take $\frac{yq}{p}$ units of T_1 from B in return. Then, their task vectors become:

$$A : (t_{A1} + \frac{yq}{p}, t_{A2} - y) \quad B : (t_{B1} - \frac{yq}{p}, t_{B2} + y)$$

And the total number of hours to finish these tasks is:

$$H_{BA} = (t_{A1} + \frac{yq}{p})p + (t_{A2} - y)q + (t_{B1} - \frac{yq}{p})s + (t_{B2} + y)r$$

These parameters need to satisfy the following conditions:

$$\text{minimize}(H_{AB}, H_{BA}) \tag{1}$$

subject to

$$x \leq t_{B1}, \quad y \leq t_{A2} \tag{2}$$

$$x \leq \frac{r}{s} \cdot t_{A2}, \quad y \leq \frac{p}{q} \cdot t_{B1} \tag{3}$$

$$\frac{p}{p+q} \leq \frac{s}{s+r}, \quad \frac{r}{s+r} \leq \frac{q}{p+q} \tag{4}$$

Condition (1) states our goal to minimize the total number of hours to finish all the tasks obtained by the users through the comparison of the cases of A initiating the trading and B initiating the trading. Conditions in (2) say that the units of a task a taker with comparative advantage in can take from the giver is less or equal to the units of the task initially obtained by the giver. Conditions in (3) talk about the corresponding return a taker can get. The return should not exceed the original units of the task the taker does not have comparative advantage in. And conditions in (4) show the comparative advantage of A in T_1 and B in T_2.

With the requirements formalized, we can derive the trade conditions next. To find out whether H_{AB} or H_{BA} is smaller, we consider $H_{AB} - H_{BA}$. After simplification, we get

Cases	Parameter relationship	Trade conditions
$x = t_{B1}, y = t_{A2}$	$p \cdot t_{B1} \leq r \cdot t_{A2}$	$B^* \leftarrow t_{A2}(T2)$ $A \leftarrow \frac{q}{p} t_{A2}(T1)$
	$p \cdot t_{B1} \geq r \cdot t_{A2}$	$A^* \leftarrow t_{B1}(T1)$ $B \leftarrow \frac{s}{r} t_{B1}(T2)$
$x = t_{B1}, y = \frac{p}{q} t_{B1}$	$q \leq r$	$B^* \leftarrow \frac{p}{q} t_{B1}(T_2)$ $A \leftarrow t_{B1}(T_1)$
	$q \geq r$	$A^* \leftarrow t_{B1}(T_1)$ $B \leftarrow \frac{s}{r} t_{B1}(T_2)$
$x = \frac{r}{s} t_{A2}, y = t_{A2}$	$p \leq s$	$B^* \leftarrow t_{A2}(T_2)$ $A \leftarrow \frac{q}{p} t_{A2}(T_1)$
	$p \geq s$	$A^* \leftarrow \frac{r}{s} t_{A2}(T_1)$ $B \leftarrow t_{A2}(T_2)$
$x = \frac{r}{s} t_{A2}, y = \frac{p}{q} t_{B1}$	$q \cdot t_{A2} \leq s \cdot t_{B1}$	$B^* \leftarrow \frac{p}{q} t_{B1}(T2)$ $A \leftarrow t_{B1}(T_1)$
	$q \cdot t_{A2} \geq s \cdot t_{B1}$	$A^* \leftarrow \frac{r}{s} t_{A2}(T1)$ $B \leftarrow t_{A2}(T_2)$

TABLE V
TRADE CONDITIONS BASED ON FOUR CASES AND PARAMETER RELATIONSHIPS

$$H_{AB} - H_{BA} = xp - yt - \frac{sqx}{r} + \frac{sqy}{p} \tag{5}$$

As we know from the above insights, to minimize the total working hours, a taker should take as many units of the task he has comparative advantage in from the giver as possible and at the same time, the number of units should satisfy conditions (2-4). So $x = \min(t_{B1}, \frac{r}{s} \cdot t_{A2})$ and $y = \min(t_{A2}, \frac{p}{q} \cdot t_{B1})$. Based on the data given, there are four cases listed in Table V depending on the values of x and y. Under each case, there are two subcases leading to different trade conditions. A trade condition in the format of $U^* \leftarrow x(T_i)$ means user U takes x units of T_i from the other user. The star on the user denotes that this user is the initiator of the trading. The trade conditions are in pairs because when one user takes some units of the task he has comparative advantage in from the other user, the other user will take some units of the task he has comparative advantage in from this user in return.

Next we use the first case to explain how we get the trade conditions in Table V. In the first case, $x = \min(t_{B1}, \frac{r}{s} \cdot t_{A2}) = t_{B1}$ and $y = \min(t_{A2}, \frac{p}{q} \cdot t_{B1}) = t_{A2}$. We plug the values of x and y into expression (5). After simplification, we get

$$(tp - sq)(\frac{t_{B1}}{r} - \frac{t_{A2}}{p}) \tag{6}$$

At the same time, from conditions in (4), we get $\frac{r}{q} \leq \frac{s+r}{p+q} \leq \frac{s}{p}$. Thus, $tp \leq sq$. So whether expression (5) is greater equal to zero depends on whether $\frac{t_{B1}}{r} - \frac{t_{A2}}{p}$ is less equal to zero. If $\frac{t_{B1}}{r} - \frac{t_{A2}}{p} \leq 0$, that is, $pt_{B1} \leq rt_{A2}$, $H_{AB} \geq H_{BA}$ will be true, which means B should initiate trading to achieve the minimum total number of hours. So B should take t_{A2} units of T_2 from A and A should take $\frac{q}{p} \cdot t_{A2}$ units of T_1 from B in return. If $\frac{t_{B1}}{r} - \frac{t_{A2}}{p} \geq 0$, in other words, $pt_{B1} \geq rt_{A2}$, $H_{AB} \leq H_{BA}$ will be true, which means A should initiate trading and take t_{B1} units of T_1 from B and B should take $\frac{s}{r} \cdot t_{B1}$ units of T_2 from A in return. The next three cases can be derived in the same way. After combining the identical trade conditions and writing the cases just in terms of the efficiency and task vectors, Table V is converted into Table VI.

Cases	Trade conditions
$t_{B1} \le \frac{r}{s} t_{A2}, t_{A2} \le \frac{p}{q} t_{B1}, p \cdot t_{B1} \le r \cdot t_{A2}$ or $\frac{r}{s} t_{A2} \le t_{B1}, t_{A2} \le \frac{p}{q} t_{B1}, p \le s$	$B^* \leftarrow t_{A2}(T_2)$ $A \leftarrow \frac{q}{p} t_{A2}(T_1)$
$t_{B1} \le \frac{r}{s} t_{A2}, t_{A2} \le \frac{p}{q} t_{B1}, p \cdot t_{B1} \ge r \cdot t_{A2}$ or $t_{B1} \le \frac{r}{s} t_{A2}, \frac{p}{q} t_{B1} \le t_{A2}, q \ge r$	$A^* \leftarrow t_{B1}(T_1)$ $B \leftarrow \frac{s}{r} t_{B1}(T_2)$
$t_{B1} \le \frac{r}{s} t_{A2}, \frac{p}{q} t_{B1} \le t_{A2}, q \le r$ or $\frac{r}{s} t_{A2} \le t_{B1}, \frac{p}{q} t_{B1} \le t_{A2}, q \cdot t_{A2} \le s \cdot t_{B1}$	$B^* \leftarrow \frac{p}{q} t_{B1}(T_2)$ $A \leftarrow t_{B1}(T_1)$
$\frac{r}{s} t_{A2} \le t_{B1}, t_{A2} \le \frac{p}{q} t_{B1}, p \ge s$ or $\frac{r}{s} t_{A2} \le t_{B1}, \frac{p}{q} t_{B1} \le t_{A2}, q \cdot t_{A2} \ge s \cdot t_{B1}$	$A^* \leftarrow \frac{r}{s} t_{A2}(T_1)$ $B \leftarrow t_{A2}(T_2)$

TABLE VI
TRADE CONDITIONS

C. *Trade with Multiple Tasks*

In this subsection, we present a solution to our defined problem which may involve multiple tasks. The trading of multiple tasks can be treated as a sequence of trading between two tasks whose minimum number of hours can be achieved using the trade conditions above. We first prove that our defined problem is NP-hard and then propose a heuristic algorithm based on the trade conditions.

Theorem 2. *In a mobile crowdsourcing environment OMSN, there are a group of users and m types of tasks $\{T_1, T_2, \cdots, T_m\}$. Each user has an efficiency vector and a task vector. When two users meet opportunistically, they trade the tasks they have comparative advantage in. Then given a list of such tradable tasks and the minimum number of hours they can achieve to trade a pair of tasks, finding the minimum total number of hours to finish all these tasks is NP-hard.*

Proof sketch. We provide here a proof sketch due to limited space. This problem resembles the travelling sales man problem [3] (TSP). Here each task is a city and the minimum number of hours to trade a pair of tasks is the distance between two cities. The trade sequence to achieve the minimum total number of hours to finish all these tasks is the minimum possible route to visit the cities. Furthermore, in our problem, a task can be visited multiple times because it can be traded with multiple other tasks. And once a task is traded with another task, the user's task vector is updated. So our problem is more complicated than TSP. Since TSP is NP-hard, our problem is NP-hard. □

Since our problem is NP-hard, there is no polynomial time solution unless $P = NP$. So we propose a heuristic task trading algorithm (TTA) to solve the problem. The TTA algorithm is shown in Fig. 1. We assume that each user x has an efficiency vector and a task vector. When two users x and y meet, they first find the tradable tasks where they have comparative advantage. These tradable tasks must appear in pairs according to Theorem 1. For each pair of such tradable tasks, x and y trade according to the trade conditions in Table VI. For the trade sequence in the tradable tasks, it is related to the user efficiency and task vectors. The intuition is that we should first trade two tasks with many units and with a big difference in task efficiency. In that case, tasks can be performed by the most efficient users and more tasks can be exchanged. After trading, the task vectors of x and y are updated. The TTA algorithm will continue as long as there are tradable tasks in the network. After some time, the tasks will be concentrated in the hands of the most efficient users to finish these tasks and the whole efficiency of the network will be greatly improved.

Algorithm TTA: Task-Trade Algorithm

Require: Each user x has an efficiency vector and a task vector
while there are tradable tasks **do**
  /* On contact between a user x and a user y */
  x and y identify tradable tasks where they have comparative advantage;
  for all the tradable tasks **do**
    pick a pair of tasks with a large number of units and the biggest difference in efficiency
    x and y trade the two tasks according to the trade conditions in Table VI;
    the task vectors of x and y are updated
  end for
end while

Fig. 1. The task-trade algorithm

V. SIMULATIONS

In this section, we evaluate the performance of our proposed scheme TTA by comparing it with the one without considering trading and the brute force algorithm to find the minimum number of hours to finish the tasks. To obtain the opportunistic meetings of users, we used the two real traces posted on the Crawdad website [1], namely the Infocom [19] trace and the upb/hyccups [7] trace. We wrote a customized simulator in Matlab to apply the traces to the algorithms.

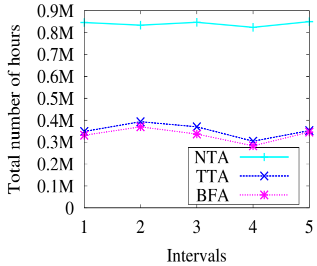
A. *Traces Used*

1) The Infocom trace (v. 5/29/2009) [19]: The Infocom trace has been widely used to test routing algorithms in mobile social networks [8], [21]. The trace recorded Infocom 2006 attenders' encounter history using Bluetooth small devices (iMotes) for 4 days at the conference.

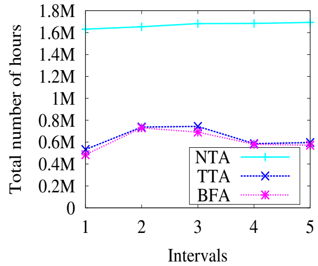
2) The upb/hyccups trace (v. 10/17/2016) [7]: The upb/hyccups trace was collected at the University Politehnica of Bucharest, using an application entitled HYCCUPS Tracer. It gathered information about a device's encounters with other nodes or with wireless access points. The duration of the tracing experiment was 63 days and had 42 participants.

B. *Algorithms Compared*

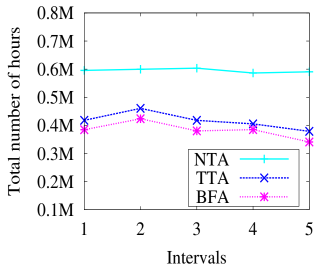
1) *The Task-Trade Algorithm* (TTA): our proposed heuristic algorithm where two users trade when they meet opportunistically.
2) *The Brute-Force Algorithm* (BFA): the algorithm to achieve the minimum total number of hours to finish all the tasks when users meet and trade. This algorithm is implemented using dynamic programming and is used as a benchmark to test our proposed algorithm.



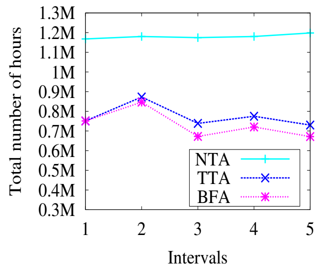
(a) Infocom trace - 5 task types



(b) Infocom trace - 10 task types



(c) upb/hyccups trace - 5 task types



(d) upb/hyccups trace - 10 task types

Fig. 2. Comparison of the algorithms using Infocom and upb/hyccups traces

3) *The No-Trade Algorithm* (NTA): the algorithm where users do not trade when they meet.

C. Simulation Setup

In our simulations, we divided each trace time into five intervals. For each user in each interval, we randomly generated his efficiency and task vectors. Each element in the efficiency vector was randomly picked in the range of $[1, 10]$ and each element in the task vector was randomly generated in the range of $[0, 1000]$. We tried five and ten types of tasks. We used the *total number of hours* to evaluate the performance of our algorithms. We ran each algorithm 1000 times and averaged the results of the evaluation metric.

D. Simulation Results

The simulation results using the Infocom trace and the upb/hyccups trace are shown in Fig. 2 (a)(b) and Fig. 2 (c)(d), respectively. The horizontal axis represents the trace intervals and the vertical axis represents the total number of hours for the users to finish all the tasks. The notation "M" means 10^6. In both traces with both five and ten types of tasks, we can see that trading can substantially reduce the total number of hours to finish the tasks obtained by the users, which confirms the benefits of trading using comparative advantage. In addition, our proposed algorithm TTA shows its competency by being very close to the benchmark brute force algorithm.

VI. Conclusion

In this paper, we have worked on task allocation in crowdsourcing in OMSNs. Different from other works, users can trade tasks when they meet opportunistically. We have proved that our defined problem is NP-hard and proposed a heuristic task-trade algorithm TTA. Simulation results have confirmed that TTA can substantially reduce the total number of working hours compared with the one without trading and it is very close to the benchmark brute force algorithm. In this paper, we have mainly worked on the trade conditions. In the future, we will continue optimizing the trading sequence to further bring down the total working hours.

Acknowledgments

This research was supported in part by NSF under CNS1305302 and ACI1440637.

References

[1] Crawdad - A Community Resource for Archiving Wireless Data At Dartmouth. http://www.crawdad.org/.

[2] The Zettabyte Era: Trends and Analysis. http://www.cisco.com/c/en/us/solutions.

[3] Travelling salesman problem. https://en.wikipedia.org/wiki/Travelling_salesman_problem.

[4] A. Asadi, Q. Wang, and V. Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communication Surveys & Tutorials*, 16(4), 2014.

[5] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola. Fostering participaction in smart cities: A geo-social crowdsensing platform. *IEEE Communications Magazine*, 51:112–119, 2013.

[6] M. H. Cheung, R. Southwell, F. Hou, and J. Huang. Distributed timesensitive task selection in mobile crowdsensing. In *ACM MobiHoc*, 2015.

[7] R. I. Ciobanu and C. Dobre. CRAWDAD dataset upb/hyccups (v. 20161017). http://crawdad.org/upb/hyccups/20161017,https://doi.org/10.15783/C7TG7K, October 2016.

[8] X. Deng, L. Chang, J. Tao, J. Pan, and J. Wang. Social profile-based multicast routing scheme for delay-tolerant networks. In *IEEE ICC*, pages 1857–1861, 2013.

[9] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: vehicular content delivery using wifi. In *Proc. ACM MobiCom*, 2008.

[10] A. Helmy G. S. Tuncay, G. Benincasa. Participant recruitment and data collection framework for opportunistic sensing: a comparative analysis. In *The 8th ACM MobiCom workshop on Challenged networks*, 2013.

[11] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 49:32–39, 2011.

[12] Z. He, J. Cao, and X. Liu. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *IEEE INFOCOM*, 2015.

[13] X. Hu, T. H. S. Chu, H. C. B. Chan, and V. C. Leung. Vita: A crowdsensing-oriented mobile cyber-physical system. *IEEE Transactions on Emerging Topics in Computing*, 1:148–165, 2013.

[14] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos. User recruitment for mobile crowdsensing over opportunistic networks. In *IEEE INFOCOM*, 2015.

[15] J. Liu, N. Kato, J. Ma, and N. Kadowaki. Device-to-device communication in lte-advanced networks: A survey. *IEEE Communication Surveys & Tutorials*, 16(4), 2014.

[16] S. Reddy, D. Estrin, and M. Srivastava. Recruitment Framework for Participatory Sensing Data Collections. In *International Conference on Pervasive Computing*, 2010.

[17] D. Ricardo. *On the Principles of Political Economy and Taxation (1 ed.)*. London: John Murray, 1817.

[18] J. Zhang S. He, D.-H. Shin and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *IEEE INFOCOM*, 2014.

[19] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom2006(v.2009-05-29). http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006, 2009.

[20] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu. Multi-task assignment for crowdsensing in mobile social networks. In *IEEE INFOCOM*, 2015.

[21] Y. Xu and X Chen. Social-similarity-based multicast algorithm in impromptu mobile social networks. In *IEEE Globecom*, 2014.

[22] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In *IEEE INFOCOM*, 2014.