Li et al. EURASIP Journal on Wireless Communications and Networking (2016) 2016:166
DOI 10.1186/s13638-016-0668-3

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Equilibrium is priceless: selfish task allocation for mobile crowdsourcing network

Qingyu Li[1], Panlong Yang[2,4*], Shaojie Tang[3], Maotian Zhang[2] and Xiaochen Fan[2]

Abstract

Recent years have witnessed major innovations in mobile crowdsourcing networks. For selfish participants, conventional methods resort to incentive mechanism design for resource utilization, which might overlook the inherent equilibrium property among mobile users. In contrast to these proposals, we investigate the problem that whether or not the selfish users could be enabled to endorse stable task sharing with balanced allocations without incentive mechanism designs. Before making a positive answer to this problem, we need to address the following challenge, i.e., users have to make their balancing decisions with only very limited and dynamic local load information, which could possibly incur longer convergence time and imbalanced task allocations. In tackling this difficulty, we propose two distributed selfish load balancing schemes, the *max-weight best response* policy for *strong information* scenario, where load information could be sufficiently collected; and the *proportional allocation* policy for *weak information* scenario. We make experimental studies to validate proposed schemes. In our simulation study with real trace data, the proposed schemes converge fast in many typical settings with fairly good balancing performance. As for data traces from RollerNet (Tournoux et al., The accordion phenomenon 2009), the performance of load balancing and convergence property are further validated.

Keywords: Selfish task allocation, Mobile crowdsourcing, Load balancing

1 Introduction

Recent years have witnessed the emergence of mobile crowdsourcing network, which is a human based, distributed, and task-driven ecosystem, such as Serendipity [1], Bikenet [2], and smartphone-based sensing applications [3–8]. Instead of improving the capacity and availability among users, these systems enable the unconscious cooperations for tasks, incorporating distributed mobile users in non-invasive manner, and achieving cloud-like service with loosely coupled mobile devices. Furthermore, mobile crowdsourcing networks could be proliferated to promote distributed cooperations among users.

To this end, the mobile data and task load could be effectively shared and allocated among users and enabled to utilize network resources more effectively and efficiently [1].

Unfortunately, status quo solutions heavily rely on incentive mechanism design [9, 10], where users are encouraged to participate the crowdsourcing practice for rewards. In contrast, hardly any of these solutions focus on the fair task allocation among users, i.e., encouraging more selfish users involved in crowdsourcing practice [11, 12].

Thus, balancing the working load among selfish users is desperately needed [11–13]. Indeed, an inevitable stalemate exists: If there is no effective load balancing scheme among participatory users, it is difficult to propose a powerful incentive mechanism design, because no user would like to cooperate under unfair allocations. Fully considering the selfish behavior among users would effectively help to break this stalemate. This paper investigates a simple but fundamental question: can we perform efficient cooperation among distributed crowdsourcing networks, fully considering the *selfish behaviors* among mobile users? Furthermore, can such cooperative design converge fast enough in dynamic and uncertain crowdsourcing network? Positive answers to these questions could enhance

*Correspondence: panlongyang@gmail.com
[2]College of Communications Engineering, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China
[4]University of Science and Technology of China, JinZhai Road Baohe District, Hefei, Anhui, 230026, People's Republic of China
Full list of author information is available at the end of the article

the gains of crowdsourcing network to a more applicable system.

In this work, we present an innovative task allocation scheme, where tasks are distributed among selfish mobile users. In that, the *selfish* behavior could be summarized in the sense that mobile users are attempting to optimize their own situation[1], i.e., trying to assign their tasks to the least loaded user, instead of trying to optimize the global situation. In general, Nash equilibrium among selfish users is a balanced state, such that there is no incentive among users for changing the status quo allocation. In other words, no user has an incentive to reallocate their tasks to other users.

Our algorithms show good balancing performance with fast enough convergence speed in both *strong* and *weak* information cases[2]. Surprisingly, even for the *weak information* case, proportional allocation scheme could achieve similar performance to that for strong information case. Specifically, for real trace data set [14], the algorithms could still show relatively good converge property. We further show that the Nash equilibrium is a fairly good balancing allocation for each user. Such that, our design could improve the cooperation efficiency and reduce the potential cost for future crowdsourcing network as well. Moreover, important factors are discussed and evaluated with spectrum of settings.

The rest of the paper is organized as follows. We review the related work in Section 2. After that, we introduce our basic system model and formulate our investigate problem in Section 3. Section 4 introduces our proposed scheme with algorithmic descriptions. We present the experimental evaluation results in Section 5 and conclude the paper in Section 6.

2 Related work

Our work relates to the efficient data transfer schemes over disruption-tolerant networks or opportunistic networks [1, 15, 16]. The intermittent contacts between randomly roaming users are useful for data sharing, which has been explored and studied extensively in variety of network settings, from military warfare [17] to disaster recovery [18]. The opportunistic access will bring more chances for data sharing. Even further, the social relationship will enhance these opportunities with higher task execution efficiency and stability.

Selfish load balancing correlates to the *congestion game*, which is very similar to our concerned working scenario. The aforementioned results have not addressed how to find Nash equilibrium efficiently. For worst case analysis, the convergence time could be exponentially long [19–21]. While best and better response dynamics are a plausible model of selfish behavior, the associated algorithms typically require that migrations be done one-by-one, and another common assumption is that best responses are always selected.

Even-Dar and Mansour [22] consider concurrent and independent task migration policies, where tasks are allowed to migrate from the overloaded machines to the under-loaded resources. Such assumption would break the selfish behavior rule and could not achieve the Nash equilibrium. To this end, the ε-approximate Nash equilibrium could be leveraged to settle down this stalemate [20, 23]. A state x is called an ε-approximate Nash equilibrium, when for any $0 \le \varepsilon \le 1$, no task can decrease the load by more than factor of $(1-\varepsilon)$. In such a state, for every user i and every neighboring user j, the ε-approximate Nash equilibrium could be achieved, where the following inequality holds:

$$(1-\varepsilon) \cdot \frac{W(x_i)}{s_i} \le \frac{W(x_j)+1}{s_j} \qquad (1)$$

where $W(x_j)$ is the sum of task weights on machine *j*.

3 System model and problem formulation

3.1 System model

In our system model, the task could be a typical sensing or computing task. Users are considered as selfish agents, where actions are unilaterally played, and tasks are migrated concurrently. Note that, although there is a platform, it is hardly applicable to incorporate centralized scheduling due to large network overhead and delays [24, 25]. In our model, the decision should be made in distributed way, and no global knowledge is available to any mobile user.

As the crowdsourcing tasks are increasing over the time, the number of users should be constant. Although the users could leave and join the crowdsourcing network dynamically, the number of participatory users are limited. Thus, suppose that $m \gg n$, which is typically considered in previous studies [12, 20].

As shown in Fig. 1, we discuss the difference between traditional load balancing and selfish load balancing schemes. If the load among users are equal, it is called load balancing state. As depicted in Fig. 1a, the task load of each user is 3 and they have reached balancing state. However, in selfish load balancing scheme, every user tends to selfishly migrate task to the least loaded neighbors. In Fig. 1b, if user A migrates a task to B, where the weight of task equals to 1, the cost of this task is $2+1+1=4$, which is greater than the initial cost (2) in user A. If user A removes this task to user C, the cost of this task could be $2+2+1$, which is still greater than the initial cost in user A. Such that, user A would not migrate tasks at the moment. For user C, any task migration would lead to the increase of task cost. To this end, each user could not reduce the task cost by unilaterally migrating tasks to anyone, which leads to a *nash equilibrium* state.

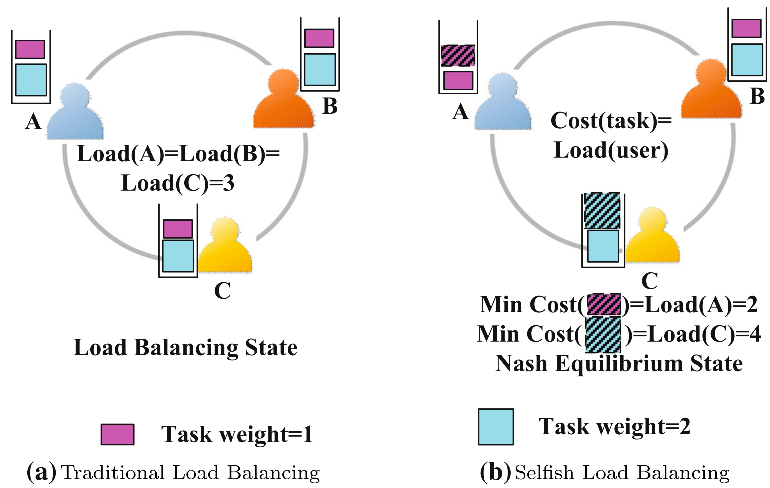


Fig. 1 Difference between traditional load balancing scheme and selfish load balancing scheme

3.2 Problem formulation

Our main objective is to propose an efficient task loading algorithm for each selfish mobile user. Additionally, there are two design requirements:

- The "$\varepsilon - Nash$" condition, *i.e.*, for any $0 \leq \varepsilon \leq 1$, no task can decrease the load by more than factor of $(1 - \varepsilon)$.
- The algorithm should converge fast enough, since the mobile crowdsourcing network only provides intermittent connections.

Typically, we consider two working scenarios:

- First, we suppose mobile users have sufficient time and bandwidth, for sharing the task load information among all neighbor nodes. As depicted in Fig. 2a, users could use WiFi connections and have relatively long contact window to share data, such working scenario is typical for crowdsourcing network [1].
- Second, we consider the weak connection case, where the contact window is short or the transmission bandwidth is limited. For mobile crowdsourcing network, this concern is very necessary as such weak communication opportunities are very common. Although the data sharing is not sufficient, contacting with small number of users could be allowed, which is illustrated in Fig. 2b.

4 Algorithm description

4.1 Max-weight best response (strong information case)

We investigate typical case for strong information, where users could have their neighbors' load information in each round. In other words, people could compare the load of

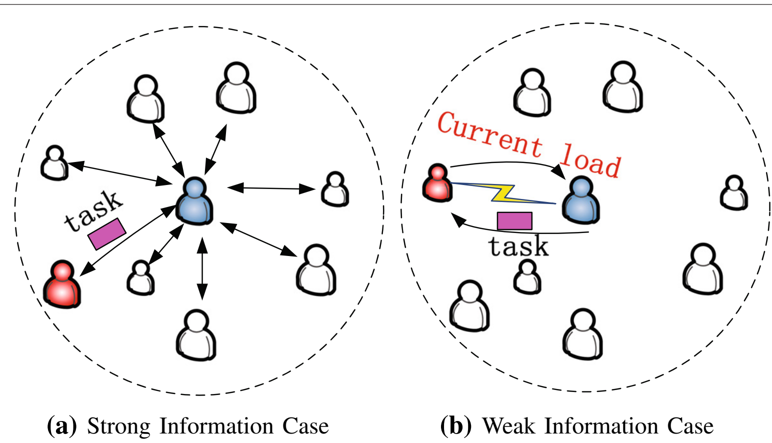


Fig. 2 Two typical working scenarios for information collection in crowdsourcing networks

among users accurately in communication range to make decision wisely.

We leverage the basic principle that users could reach a satisfied state *if and only if* the task is placed on the leasted loaded user among neighbors [23]. Such that, a *best response* never decreases the minimum load among users [23]. To this end, a satisfied user could be unsatisfied only for one reason: i.e., the load of the user holding this task increases because another task moves in. Inspired by this intuition, we proposed an adaptive algorithm: *max-weight best response*.

Algorithm 1: Max-weight best response

```
input : Task T, User U, Weight W, Speed S,
        Round interval t, Round times R,
        Initial user load L = {l_1, l_2, ... l_n},
        Encounter(i, j), {i, j ∈ U, i ≠ j}.
output: User load: L, Gap in each round: G

for round = 1 : R do
    for task i in descending weight order do
        Let u(i) be the current located user of task i for
        u(j) ∈ N/u(i) do
            if
            Encounter(u(i), u(j)) ∈ [t∗(round − 1), t∗
            round] then 𝒩 ← j else continue
        end
        if 𝒩 ≠ ∅, Find the least loaded neighbor q then
            l_{u(i)}^{round}:current load of user u(i) l_{jj}^{round}:
            current load of user jj
            if l_{u(i)}/s_{u(i)} > (l_q + W(i))/s_q then
                Move task i from user u(i) to q
                u(i) ← q
                L ← {l_1, l_2, ... l_q + w_i, ..., l_{u(i)} −
                w_i, ..., l_n}
            end
        end
    end
end
```

The proposed algorithm works as follows: Given tasks $\mathbf{T} = \{1, 2, \ldots, m\}$ and task weight $\mathbf{W} = \{w_1, w_2, \ldots w_m\}$ accordingly, mobile users $\mathbf{U} = \{1, 2, \ldots n\}$ could be enabled with diverse computing speed: $\mathbf{S} = \{s_1, s_2, \ldots s_n\}$. First, we rank the task weights in descending order. The unsatisfied user holding the task with highest weight should be considered firstly. If one would always allocate the task of minimum weight among the unsatisfied users, then it will need an exponential number of best response steps to reach a pure Nash equilibrium [23, 26].

Let $u(i)$ denote the user holding task i, for other user $u(j)$, if there is an encounter record between user $u(i)$ and $u(j)$ in current round interval, we include $u(j)$ in the neighbor set of $u(i)$. As we have mentioned in the very first of this section, *if and only if* the task is handed over to the least loaded user, the dynamic equilibrium could be realized eventually. Under this discipline, we use the polling method to get the least loaded neighbor $u_{\min}$ and compare the work load of $u(i)$ and $u_{\min}$.

4.2 Proportional allocation (weak information case)

Since the network is distributed, intermittent, and unpredictable, it is difficult to get the global information for task load [27]. As depicted in Fig. 2b, users can only get weak local information with limited number of neighbors. For this typical scenario, we put forward *proportional allocation*, where tasks are migrated in parallel for each round.

Algorithm 2: Proportional allocation

```
input : Task T, User U, Weight W, Speed S,
        Round interval t, Round times R,
        Initial user load L = {l_1, l_2, ... l_n},
        Encounter(i, j), {i, j ∈ U, i ≠ j}.
output: User load: L, Gap in each round: G

for round = 1 : R do
    for each task in parallel do
        for j ∈ N/u(i) do
            if
            Encounter(u(i), j) ∈ [t ∗ (round − 1), t ∗
            round] then 𝒩 ← j
        end
        if 𝒩 ≠ ∅, randomly choose neighbor k then
            if l_{u(i)}/s_{u(i)} > l_k/s_k then
                P ← 1 − l_k/l_{u(i)}
                l_k ← l_k + W(i) ∗ P
                l_{u(i)} ← l_{u(i)} − W(i) ∗ P
                u(i) ← k
            end
        end
    end
end
```

Let $l_{u(i)}$ be the current load of $u(i)$, where $u(i)$ denotes the user holding task i. Next, according to the weak local information constraint, the user $u(i)$ would randomly choose a neighbor k to compare its task load. Due to the selfishness behavior, if the load on user k is smaller than that of user $u(i)$, task i could be migrated from user $u(i)$ to user k with probability $P = 1 - l_k/l_{u(i)}$. Different from Algorithm for *strong information* case, users could make such mixed strategies to ensure self-profit, where the probability distribution over different choices could be leveraged to achieve Nash equilibrium within limited steps.

5 Performance analysis and evaluation

5.1 Simulation tests

5.1.1 Load balancing performance

We focus on the load balancing problem caused by users' selfishness. First of all, we evaluate the load balancing performance with the converged task load. We set different task weight values to each task, such that the tasks could be migrated among users. If the task weights are randomly selected from a larger range, the diversities among users would be more significant.

In this evaluation, the duration of task sharing round is set to 20 s, and the inter-contact frequency λ is uniformly distributed in range[3] $[10, 30]$. In Fig. 3a, the task weight range is set to $[1, 50]$. Similarly, in Fig. 3b, c, the task weight range is set to $[1, 100]$ and $[1, 500]$, respectively. As depicted in the three sub-figures in Fig.3, *proportional allocation* and *max-weight best response* could achieve balanced load among users. Surprisingly, although *proportional allocation* scheme only has weak information, the balancing performance is comparable to that of *max-weight best response* scheme. The reason is, in our simulations, the proportional allocations are executed in parallel, which could make task migration with sufficient times. Thus, the task load could be effectively compared among users. Such that, it could still achieve similar performance in strong information scheme, because mobility would help the contacts among users, and provide opportunities for task load information sharing.

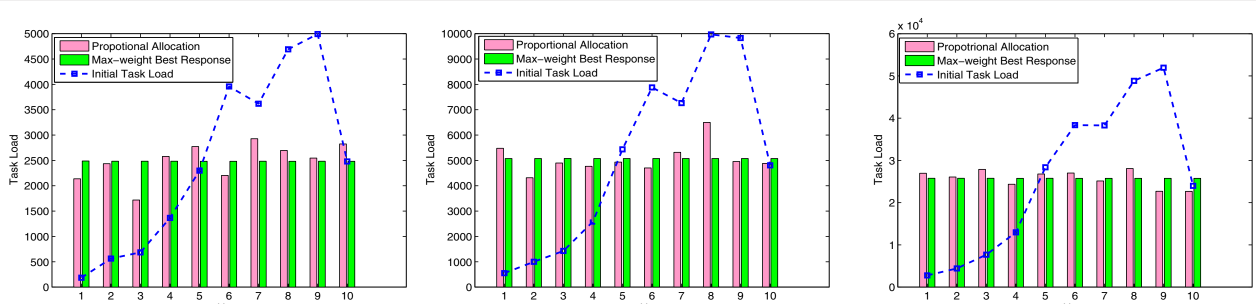
Note that, for *proportional allocation* scheme, there are small portion of unbalanced load in some users, which have been shown in Fig. 3b, c. The reason is, with the increasing range of task weight, some tasks with large task weight could not migrate to other users, since no user would admit the migration for such a big job for some rounds in our simulation.

5.1.2 Convergence property

The convergence property is fundamentally important to load balancing scheme. Specifically, in our study, we need the algorithm with ensured convergence property and fast enough convergence speed.

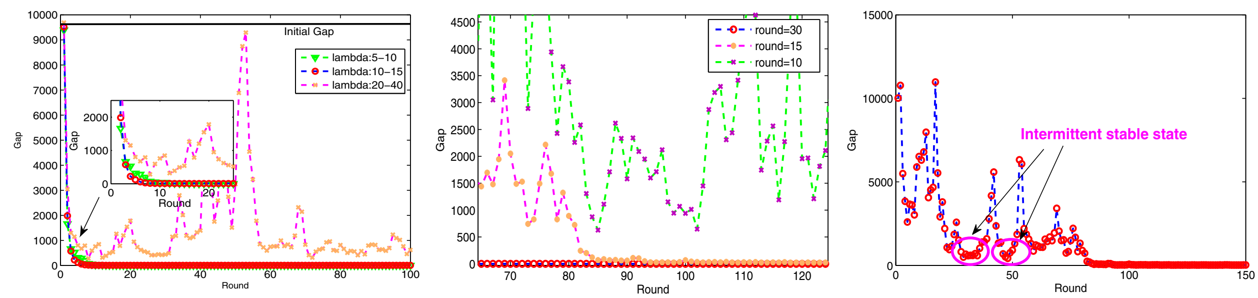
Basically, we evaluate the convergence speed of the *proportional allocation* scheme. Note that the convergence time relates to the duration of each task allocation round. This parameter correlates to the mobility, i.e., the inter-contact frequency, λ. As depicted in Fig. 4a, we set different values for λ, i.e., and the ranges for random selection are set to $[5, 10]$, $[15, 20]$, $[20, 40]$, respectively. The length of each round is set to 20. When the range value for λ is set to $[5, 10]$, $[15, 20]$, we find that, the convergence property is good. However, when it is set to $[20, 40]$, the convergence property does not hold anymore. The reason is, when the contact interval becomes larger, users would have less opportunities for task migration. Thus, the assumption for sufficient data sharing in each round does not hold. Users with more contact opportunities could unfairly offload tasks.

This observation motivates us to make a further study. That is, we need to investigate the relationship between the duration of each round and the inter-contact interval, i.e., the λ. As shown in Fig. 4b, when the duration of each sharing round is set to 30, and the inter-contact frequency λ is set uniformly with interval $[10, 20]$. The *proportional allocation* scheme could perfectly converge. When the round length is set to 15, there is a dynamic process, and the algorithm could achieve convergence state after 80 rounds again. Notably, when the round length is set to 10, there is no significant convergence state shown in our evaluations. The reason is, due to the extremely short round length, users could not migrate their tasks sufficiently, i.e., from dynamic allocation to stable Nash equilibrium, where the balanced task allocation could not be achieved. To this end, the task sharing rounds should be larger or approximate to that of the inter-contact



(a) Task weight randomly from 1 to 50 **(b)** Task weight randomly from 1 to 100 **(c)** Task weight randomly from 1 to 500

Fig. 3 Load balancing performance comparisons between *proportional allocation* and *best response*, when the task weights are concerned



(a) Impact of inter-contact rate, λ **(b)** Impact of round interval **(c)** Dynamics and stable states in convergence

Fig. 4 Analysing convergence characters in different inter-contact rates and round intervals

frequency, such that fast and stable convergence state could be achieved.

These dynamic features motivate us to take another concern, i.e., the dynamics during convergence process. We set round length to 15 and set the inter-contact rate λ with interval [10, 30]. We make evaluations on the convergence property of *proportional allocation*. As shown in Fig. 4c, there are two intermittent stable states, where a relatively balanced tasks assignment could be achieved. The gap at these states are very low, say, less than 500. Comparing to the initial value, 5000, it is still a good result. Note that in those states, if the Nash equilibrium condition could be relaxed, we can achieve faster convergence results.

Also, it is worth noting that we have not shown the convergence property of the *max-weight best response* method, because it could get immediate convergence property once task weights are ordered and tasks are allocated sequentially, which has been proved in [23].

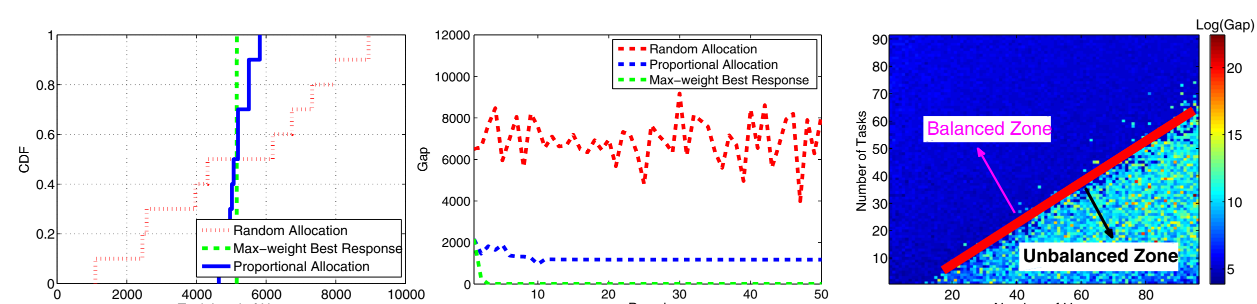
5.1.3 *Comparison with random allocation*

In order to show the performance of our algorithms, we compare them with 'Random Allocation'. As depicted in Fig. 5a, both algorithms we proposed perform better than 'Random Allocation'. Especially for Gap evaluation, as depicted in Fig. 5b, 'Random Allocation' can hardly converge but fluctuate. Whereas, 'Proportional Allocation' finally reaches equilibrium through a period of up-down and "max-weight best response" acts almost perfect because tasks are ordered by weight.

5.2 Balancing among users and tasks

As we have aforementioned in Section 3, the task number m is assumed to be much larger than the user number n considering the crowdsourcing network fact. To further explore the impact of proportion of task number and user number, we run the "proportional allocation" for different settings of m and n.

As shown in Fig. 5c, the gap for each task allocations under different number of tasks and users are colored with



(a) Load distribution with different algorithms **(b)** Gap performance with random allocation **(c)** Number of tasks vs. users, when the task load Gap is evaluated

Fig. 5 Load distribution and gap analysis data values. Note that the value is processed with log-normal function, so as to see the differences more clearly. we can clearly find two zones. One is the balanced zone, achieving balancing tasks, and the other zone is unbalanced. We draw a thick red line to separate these two zones.

According to Fig. 5c, we summarize the following properties. First, with increasing number of tasks, the chances falling in balanced zone would be large accordingly. While with the increasing number of users, the chances falling in unbalanced zone would be large. To this end, when more tasks are presented with less users, the balancing performance would possibly be good. Second, when the number of users and tasks are all very large, the algorithm would achieve unbalanced performance. In tackling with this defect, we need to balance the users and tasks. That is, we need to mitigate some users, e.g., users with less processing ability and less contact frequency, such that the balanced allocations could be achieved.

6 Conclusions

We propose a fast converging load balancing algorithm for crowdsourcing network, fully considering the selfish behavior among users. In our concern, two major network scenarios, users with strong and weak load information, are fully considered. We propose *max-weight best response* policy and simple distributed allocation scheme with probabilistic assignment. We find that these simple schemes are effective in achieving selfish load balancing. The fundamental reason is, although users are intermittently connected, the user diversity also increases, which will speedup convergence process in sense of selfish behavior considerations. Such that, the mobility provides sufficient diversity among users, which makes the task offloading policy more robust. Evaluations have been extensively processed for validating the effectiveness and efficiency of the proposed scheme.

In future work, social relationship can be explored and exploited for semi-cooperative design, where balanced task offloading could be performed among friends, and unfamiliar mobile users, and plays an important role for task execution efficiency. Finally, we plan to apply our methods to realistic applications, such as processing pictures to translate words within crowds of participants.

Endnotes

[1] In our concern, the situation is the sense of task load

[2] For strong information scenario, the local information could be sufficiently collected; while for weak information case, only local and limited information is available.

[3] The dimension for λ is hertz (Hz). We omit Hz for brevity in the following iterations.

Acknowledgements

This research is partially supported by NSF China under Grant Nos. 61232018, 61272487, and 61502520, 61402009 and PAPD, CICAEET.

Competing interests

First, we have developed a convincible system for workload balancing in mobile crowdsourcing system. In that, users could effectively leverage the equilibrium in selfish load balancing, where relatively balanced property could be available in mobile crowdsourcing networks.
Second, with extensive simulation study and real trace data evaluation, our schemes are validated in both strong and weak information scenario. Meeting interval and contact durations are further evaluated with instructive suggestions when real network protocol design is concerned.

Author details

[1] China Satellite Maritime Tracking and Control Department, JiangYin, Jiangsu 214431, China. [2] College of Communications Engineering, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China. [3] University of Texas at Dallas, Dallas, TX 75062-4736, USA. [4] University of Science and Technology of China, JinZhai Road Baohe District, Hefei, Anhui, 230026, People's Republic of China.

Received: 1 April 2016 Accepted: 5 July 2016
Published online: 13 July 2016

References

1. C Shi, V Lakafosis, MH Ammar, EW Zegura, in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*. Serendipity: enabling remote computing among intermittently connected mobile devices (ACM, 2012), pp. 145–154
2. SB Eisenman, E Miluzzo, ND Lane, RA Peterson, GS Ahn, AT Campbell, Bikenet: a mobile sensing system for cyclist experience mapping. ACM Trans. Sensor Netw. (TOSN). **6**(1), 6 (2009)
3. JA Burke, D Estrin, M Hansen, A Parker, N Ramanathan, S Reddy, MB Srivastava, Participatory sensing. Center for Embedded Network Sensing. (2006)
4. ND Lane, E Miluzzo, H Lu, D Peebles, T Choudhury, AT Campbell, A survey of mobile phone sensing. IEEE Commun. Mag. **48**(9), 140–150 (2010)
5. Z Xu, Y Liu, N Yen, L Mei, X Luo, X Wei, C Hu, Crowdsourcing based description of urban emergency events using social media big data. IEEE Transactions on Cloud Computing. **99**, 1–1 (2016). doi:10.1109/TCC.2016.2517638
6. Z Xu, Y Liu, J Xuan, H Chen, L Mei, Crowdsourcing based social media data analysis of urban emergency events. Multimedia Tools and Applications, 1–18 (2015)
7. Z Xu, H Zhang, V Sugumaran, K-KR Choo, L Mei, Y Zhu, Participatory sensing-based semantic and spatial analysis of urban emergency events using mobile social media. EURASIP Journal on Wireless Communications and Networking (2016)
8. Z Xu, H Zhang, C Hu, L Mei, J Xuan, K-KR Choo, V Sugumaran, Y Zhu, Building knowledge base of urban emergency events based on crowdsourcing of social media. Concurrency and Computation: Practice and Experience (2016). http://dx.doi.org/10.1002/cpe.3780
9. P Ipeirotis, F Provost, J Wang, in *Proceedings of the ACM SIGKDD workshop on human computation*. Quality management on amazon mechanical turk (ACM, 2010), pp. 64–67
10. D Yang, G Xue, X Fang, J Tang, in *Proceedings of the 18th annual international conference on Mobile computing and networking*. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing (ACM, 2012), pp. 173–184
11. T Luo, C-K Tham, in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*. Fairness and social welfare in incentivizing participatory sensing (IEEE, 2012), pp. 425–433
12. I Koutsopoulos, in *Infocom, 2013 proceedings ieee*. Optimal incentive-driven design of participatory sensing systems (IEEE, 2013), pp. 1402–1410

13. O Scekic, H-L Truong, S Dustdar, Incentives and rewarding in social computing. Communications of the ACM. **56**(6), 72–82 (2013)
14. PU Tournoux, J Leguay, F Benbadis, V Conan, MD De Amorim, J Whitbeck, in *INFOCOM 2009, IEEE*. The accordion phenomenon: Analysis, characterization, and impact on dtn routing (IEEE, 2009), pp. 1116–1124
15. Z Fu, X Sun, Q Liu, L Zhou, J Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans. Commun. **E98-B**, 190–200 (2015)
16. Z Fu, K Ren, J Shu, X Sun, F Huang, Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Transactions on Parallel and Distributed Systems. **99**, 1–1 (2015). doi:10.1109/TPDS.2015.2506573
17. P Marshall, in *MILCOM 2008-2008 IEEE Military Communications Conference*. DARPA progress towards affordable, dense, and content focused tactical edge networks (IEEE, 2008), pp. 1–7
18. K Fall, G Iannaccone, J Kannan, F Silveira, N Taft, in *ISCRAM*. A disruption-tolerant architecture for secure and efficient disaster response communications (Citeseer, 2010)
19. S Fischer, H Räcke, B Vöcking, Fast convergence to Wardrop equilibria by adaptive sampling methods. SIAM Journal on Computing. **39**(8), 3700–3735 (2010)
20. P Berenbrink, T Friedetzky, LA Goldberg, PW Goldberg, Z Hu, R Martin, Distributed selfish load balancing. SIAM Journal on Computing. **37**(4), 1163–1181 (2007)
21. P Berenbrink, M Hoefer, T Sauerwald, Distributed selfish load balancing on networks. ACM Transactions on Algorithms (TALG). **11**(1), 2 (2014)
22. E Even-Dar, A Kesselman, Y Mansour, in *International Colloquium on Automata, Languages, and Programming*. Convergence time to Nash equilibria, vol. 2719 (Springer, 2003), pp. 502–513
23. N Nisan, T Roughgarden, E Tardos, VV Vasirani, *Algorithmic game theory*, vol. 1. (Cambridge University Press Cambridge, 2007)
24. S Xie, Y Wang, Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. Wireless Person. Commun. **78**, 231–246 (2014)
25. T Ma, J Zhou, M Tang, Y Tian, A Al-Dhelaan, M Al-Rodhaan, S Lee, Social network and tag sources based augmenting collaborative recommender system. IEICE Trans. Inform. Syst. **E98-D**, 902–910 (2015)
26. P Guo, J Wang, B Li, S Lee, A variable threshold-value authentication architecture for wireless mesh networks. J. Int. Technol. **15**, 929–936 (2014)
27. Y Ren, J Shen, J Wang, J Han, S Lee, Mutual verifiable provable data auditing in public cloud storage. J. Int. Technol. **16**, 317–323 (2015)