International Journal of Internet, Broadcasting and Communication Vol.12 No.4 102-108 (2020)
http://dx.doi.org/10.7236/IJIBC.2020.12.4.102

IJIBC 20-4-13

A hybrid tabu search algorithm for Task Allocation in Mobile Crowd-sensing

Shathee Akter, Seokhoon Yoon*

Ph.D Candidate, Department of Electrical and Computer Engineering, University of Ulsan, Ulsan, Korea
Professor, Department of Electrical and Computer Engineering, University of Ulsan, Ulsan, Korea
erittrashathee@gmail.com, seokhoonyoon@ulsan.ac.kr

Abstract

One of the key features of a mobile crowd-sensing (MCS) system is task allocation, which aims to recruit workers efficiently to carry out the tasks. Due to various constraints of the tasks (such as specific sensor requirement and a probabilistic guarantee of task completion) and workers heterogeneity, the task allocation become challenging. This assignment problem becomes more intractable because of the deadline of the tasks and a lot of possible task completion order or moving path of workers since a worker may perform multiple tasks and need to physically visit the tasks venues to complete the tasks. Therefore, in this paper, a hybrid search algorithm for task allocation called HST is proposed to address the problem, which employ a traveling salesman problem heuristic to find the task completion order. HST is developed based on the tabu search algorithm and exploits the premature convergence avoiding concepts from the genetic algorithm and simulated annealing. The experimental results verify that our proposed scheme outperforms the existing methods while satisfying given constraints.

Keywords: *task allocation, traveling salesman problem, tabu search, genetic algorithm, mobile crowd-sensing*

1. Introduction

A new sensing platform called mobile crowd-sensing (MCS) has recently received a great amount of attention, which exploits the sensing capability of the rapidly increasing smart devices, including smart vehicles, phones, and wearable devices, to perform sensing tasks and share local knowledge in their surroundings environment [2]. A typical MCS system has an MCS platform where tasks requester publishes various sensing tasks, and the platform recruits a set of workers to perform the tasks from the registered workers who are ordinary citizens with smart devices. In addition, the workers are remunerated for performing the tasks. Thus, the success of an MCS application mostly depends on the proper task-worker assignments. Note that the MCS system has benefitted many MCS applications such as intelligent transportation system, environmental monitoring, and information mapping, and so on.

Manuscript Received: August. 19, 2020 / Revised: August. 21, 2020 / Accepted: August. 27, 2020
Corresponding Author: seokhoonyoon@ulsan.ac.kr
Tel: +82-52-259-1403, Fax: +82-52-259-1687
Professor, Department of Computer Engineering, University of Ulsan, Korea

There have been numerous studies on task allocation problem in MCS, where tasks are mostly located in specific locations, and workers need to visit the locations to perform the tasks. Hence, it is more resource-efficient to assign multiple tasks to a worker considering that the workers resource may not be rich always and the task-venue correlations. However, assigning multiple tasks to a worker necessitates finding the task completion order since the task's response time (the time needed for a worker to go to the task location) varies depending on the order. Response times of the tasks are critical for emergency tasks that need to be performed within the given deadline—for example, identifying gas leakage and hazardous material. Furthermore, tasks and workers can have heterogenous requirements and specifications, respectively, which require additional system-level parameters to be taken into consideration. Though our existing paper [3] has proposed a task allocation strategy while taking path planning into account, we verify that the proposed method outperforms the existing one through various simulations.

Therefore, in this paper, we propose a hybrid search algorithm for task allocation (HST) with the aim of minimizing the total distance moved by all workers while taking different types of workers, such as unmanned aerial vehicle (UAV) and human workers, and tasks various requirements (such as deadline, required sensor, and reliability (i.e., the probability at least one worker will complete the task)) into account. The proposed algorithm combines the merits of the three different algorithms, i.e., tabu search, genetic algorithm, and simulated annealing, to find the near-optimal task-worker assignment.

The rest of the paper is organized as follows: Section 2 presents the system model and problem formulation. The HST algorithm is described in section 3 and the simulation results are presented in Section 4. Section 5 concludes the paper.

2. Problem Definition

In this paper, we consider an MCS scenario where two types of workers [3], i.e., unmanned aerial vehicles (UAVs) and human workers, are registered to the MCS platform. The workers set is denoted by $\mathbb{W}$ and $|\mathbb{W}| = |\mathbb{W}^1| + |\mathbb{W}^2|$. $\mathbb{W}^1$ denotes the set of UAV workers and $\mathbb{W}^2$ denotes the set of human workers. Each worker $i (i \in \mathbb{W})$ is associated with a location l_i, moving speed M_i, available sensors set $\mathbb{S}_i$, and the reputation level r_i, which is calculated using the previous task completion history of the worker, i.e., the ratio of the tasks completed by the worker among the total assigned tasks. Each UAV (i.e., $v_i = 1$) is associated with the extra specification flight time limitation e_i. We use j to denote a task with location l_j, required sensor s_j, and required reliability q_j (i.e., the probability that at least one worker will complete the task) where $j \in \mathbb{T}$ and $\mathbb{T}$ is the tasks set.

Problem 1: *Given tasks and workers set $\mathbb{T}$ and $\mathbb{W}$, respectively, find a task-worker assignment such that each task is performed before the deadline θ, each UAVs total task completion time is within the given flight time e_i, assigned workers available sensors set $\mathbb{S}_i$ contains the required sensor s_j, and the probability that at least one worker will complete the task among the selected workers should be higher than the required reliability q_j. The objective of this paper is to find an assignment, which minimizes the total distance moved by the workers, i.e.,*

$$\min \sum_{i \in \mathbb{W}} dist(\mathbb{F}_i) \tag{1}$$

Subjected to:

$$\frac{dist(F_i)}{M_i} - t_i \leq \theta, \qquad \forall i \in \mathbb{W} \tag{2}$$

$$\frac{dist(F_i)}{M_i} \leq e_i, \qquad \forall i \in \mathbb{W}^2 \tag{3}$$

$$\{s_j\} \cap \{\mathbb{S}_i\} \neq \emptyset, \qquad \forall j \in \mathbb{T} \tag{4}$$

$$1 - \prod_{i \in \mathbb{B}_j} (1 - r_i) \geq q_j, \qquad \forall j \in \mathbb{T} \tag{5}$$

where $dist(\mathbb{F}_i)$ is the function to calculate the total length of path moved by the worker i and $\mathbb{F}_i$ is the set of ordered tasks that assigned to the worker. M_i, t_i are the moving speed and the time need to go back to the current location of the worker from the last task on the path of worker i, respectively. $\mathbb{B}_j$ is the set of workers that assigned to task j.

3. Algorithm

In this section, a hybrid search algorithm for task allocation is proposed, which is based on the tabu search and uses the idea of offspring generation from genetic algorithm (GA) [5] and random start at a new point from simulated annealing [6]; hence the name. The tabu search is a metaheuristic designed for solving challenging problems in the optimization filed by Glover [7]. The basic features of the tabu search comprised of solution representation, fitness, neighborhood, tabu list, and tenancy period [8]. We use the matrix structure, i.e., the task-worker assignment matrix S, to represent the solution where the dimension of the matrix is $|\mathbb{T}| \times |\mathbb{W}|$ and the elements can only have binary value, i.e., either 0 or 1. The value of an element is zero when the task is not assigned to the worker; otherwise, one. The fitness of an assignment matrix or a solution $f(S)$ is calculated as follows:

$$f(S) = \sum_{i \in \mathbb{W}} td_i + \alpha + \beta \tag{6}$$

where td_i is the total distance moved by the worker i, α denotes the total number of workers violated the deadline constraint and β denotes the number of tasks that remained unassigned respectively in each solution S. The fitness of each solution is obtained by using the sub-problem algorithm in [3], which uses a traveling salesman problem (TSP) heuristic to find the worker's moving path.

In each iteration, a set of candidate solutions or the neighbors of the current best solution is generated. The candidate solutions differ from each other by one element and not in the tabu list. Among the newly generated candidate solutions, the solution with the best fitness, i.e., lower total moving distance, is treated as a new current best solution. To avoid getting stuck at suboptimal regions, a tabu list is introduced in tabu search to prevent the selection of the already taken "moves" or solutions. Specifically, the tabu list is a short-term memory that remembers the moves from the previous searches and disables them for a certain time or iterations. The time or number of iterations moves are disabled (i.e., tabu) is known as the tenancy period. In HST, the tabu list contains the element from the assignment matrix, i.e., changing an element is considered as a move.

Algorithm 1 Hybrid search algorithm for task allocation (HST)

```
input: Output of the TGreedy algorithm
output: Workers set for each task
 1: $z \leftarrow$ initial solution
 2: $y \leftarrow$ initial solution
 3: tabu_list $\leftarrow$ []
 4: Set $K_{max}$
 5: for $K = 0$ through $K_{max}$ do
 6:     $T \leftarrow \frac{K_{max}}{K+1}$
 7:     Neighborhood $\leftarrow$ getNeighbors($z$)
 8:     $z \leftarrow$ Neighborhood [0]
 9:     for $c$ in Neighborhood do
10:         if $c$ is not in tabu_list and $f$ (c) $< f(z)$ then
11:             $z \leftarrow c$
12:         end if
13:     end for
14:     if there are no local improvement for a certain period then
15:         perform crossover and mutation operations and obtain new solution $s_{new}$
16:         $z \leftarrow s_{new}$
17:         repeat from $7 - 13$
18:     end if
19:     if $f(z) < f$ $(y)$ then
20:         $y \leftarrow z$
21:     else if random(0, 1) $< e^{\frac{f(z)-f(y)}{T}}$ then
22:         $y \leftarrow z$
23:     end if
24:     tabu_list.push($z$)
25:     if tabu_list.size $>$ maxTabulistSize then
26:         tabu_list.removeFirst()
27:     end if
28: end for
```

However, it is possible that there is no improving move available from the current best solution leading to getting stuck into the strict local optima. Thus, to enable the algorithm to start from a new point, the crossover and mutation operations from the GA are used to create a new solution, and the newly created solution is set as the current best solution. In GA, the crossover operation is used to generate an offspring by combining the parent's genetic information, whereas mutation maintains the genetic diversity, respectively. In this algorithm, the uniform crossover [9] is considered, and the parents for the operation is selected sequentially in pair from the neighborhood of the current best solution. The mutation operation is performed by randomly changing the value of one element from the candidate solution.

The proposed problem is very complex and may contain several local optima; hence, to further enhance the performance, the idea of selecting a worse solution with some probability is also employed in HST, which enables the algorithm to start at a random point and avoid being trapped into local optima. The probability of selecting a worse solution as the current best solution decreases as time goes by. The pseudo-code for HST is presented in Algorithm 1.

4. Result and Analysis

In this section, the experimental settings and the results are presented. To simulate the tasks and workers locations, we extract GPS records, i.e., location information, of 7 km $\times$ 7 km area from the roma/taxi [10] dataset. This dataset contains GPS trajectories of 320 taxis from February 2014 to March 2014. The location of tasks and workers are randomly chosen from the extracted GPS position information of the first day. The performance of the HST is evaluated for various number of tasks [8, **10**,12,14,16], workers [6, **10**, 14, 18, 22], deadlines [5, 6, **7**, 8, 10], and required reliabilities [**0.9**, 0.92, 0.94, 0.96, 0.98], respectively using the performance matrix total moving distance, where the values in bold font are default values. The workers reliability level is set within the range between [0.6, 1]. Furthermore, we compare our algorithm with three other methods; TGreedy and MGATA from [3] and a basic tabu search algorithm called TST [1]. Note that the initial solution of HST, TST, and MGATA is the result of the TGreedy algorithm.

Fig. 1 depicts the performance of the four algorithms (TGreedy, TST, MGATA, and HST) for different number of tasks. It can be seen from the figure that the total moving distance increases when the number of tasks varies from 8 to 16. This is because of the total number of places that need to be visited by the workers increases. Furthermore, the proposed method outperforms the other three approaches when the number of tasks increases. For example, when the number of tasks is 12, the total distance moved by the workers in HST is around 14 km, whereas they have to move around 16 km, 22km, and 26 km in case of the MGATA, TST, and TGreedy, respectively. HST performs better than MGATA and TST because it combines more local optima escaping techniques than the other two approaches. Performance of the TGreedy is lowest among them because of its greedy selection of the workers at the beginning, which leads to the sub-optimal solution.

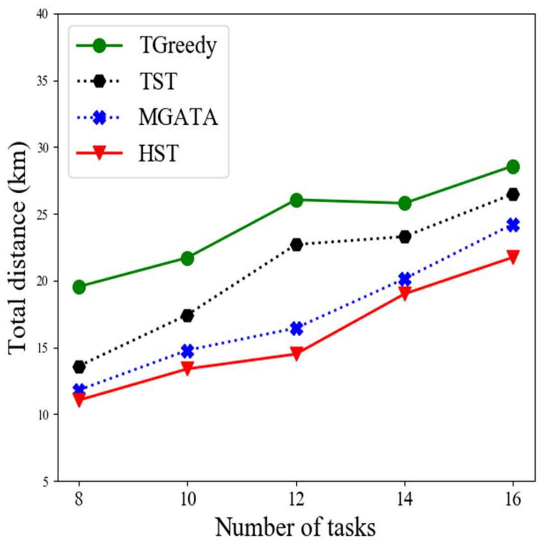

Figure 1. Effect of the different number of tasks

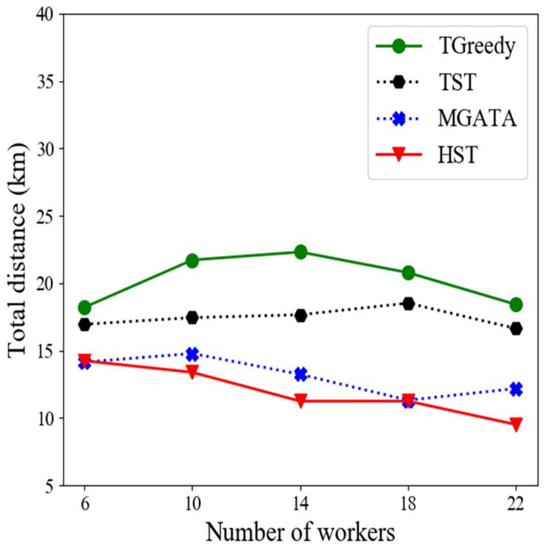

Figure 2. Effect of the different number of workers

Fig. 2 shows the effect of the different number of workers where the total moving distance decreases with the increase in the total number of workers. When the number of workers increases, the newly added workers pool contains better workers to select than the previously selected workers, hence the decreasing trend. Furthermore, it can be seen from the figure that sometimes HST and MGATA achieves the same result as both algorithms locally searches for the best solution. However, in terms of overall results, HST shows better performance. On the other hand, in case of TST and TGreedy, the total distance increases when number workers increase instead of decreasing. This is because when the number of workers increases, the solution space become large containing several local opima and both algorithms fall into local optima.

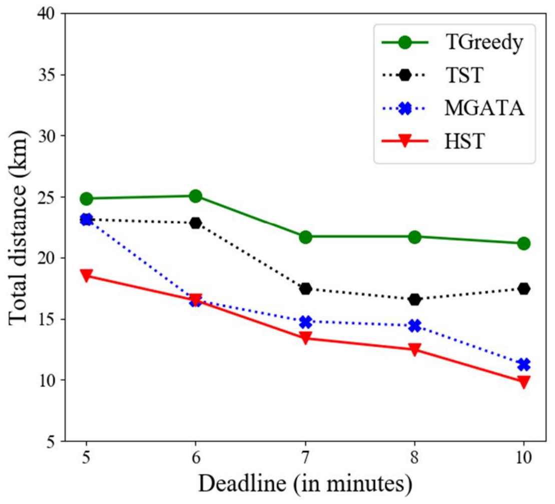

Figure 3. Effect of the different deadlines

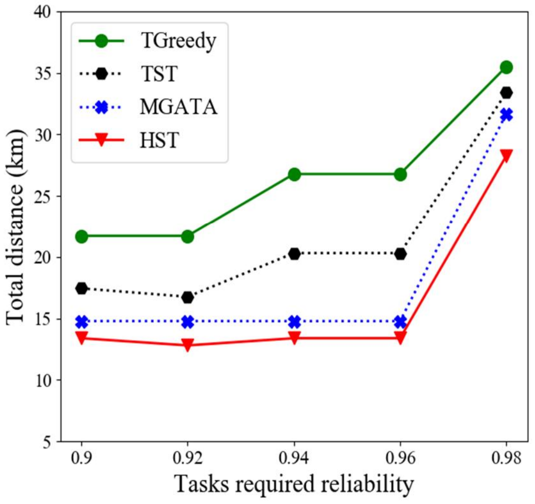

Figure 4. Effect of the different required reliabilities

In Fig. 3, the effect of the different deadlines [5, 6, 7, 8, and 10 minutes] is illustrated. The total moving distance obtained by all algorithms decreases when the deadline increases because the larger deadline enables more workers to perform the tasks within the given time. For example, workers who are located 8 minutes away from some tasks become enable to perform the tasks when the deadline is 8 minutes, which increases the opportunity to select better workers and leads to the lower total moving distance.

In Fig.4, the performance of the proposed method is shown under different required reliability of tasks. As the reliability requirements of the tasks increases, the total moving distance increases because tasks need to recruit more workers to satisfy the higher required reliability leading workers to visit more task venues. All the algorithms show an upward surge when the required reliability is 0.98 because the average number of tasks a worker has to performs increases, hence the total moving distance increases.

4. Conclusion

In this work, a deadline-constrained and location-dependent task allocation algorithm is proposed for MCS systems where a worker can perform multiple tasks, and a task also can be performed by many workers to ensure reliability. Furthermore, a worker needs to visit the task venues intentionally to complete the tasks, and the trajectory of the worker affects the response time of the task. Hence, along with the task assignment problem, the worker's moving path also needs to be taken into account. Therefore, we develop a hybrid search algorithm for task allocation with the objective of minimizing the total distance moved by the workers and finding the workers moving path while taking tasks deadline, sensor and reliability requirements into account. Simulation results under different parameters show that the proposed method achieves better performance than the greedy solution and other heuristics.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2019R1F1A1058147).

References

[1] S. Akter and S. Yoon, “Location-aware Task Assignment and Routing in Mobile Crowd Sensing,” *paper presented to The 11th International Conference on ICT Convergence (ICTC 2020),* Jeju, Korea, October 21, 2020.

[2] B. Guo et al., "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm ", *ACM Computing Surveys*, vol. 48, no. 1, pp. 1-31, August 2015.
DOI: https://doi.org/10.1145/2794400

[3] S. Akter and S. Yoon, "DaTask: A Decomposition-Based Deadline-Aware Task Assignment and Workers' Path-Planning in Mobile Crowd-Sensing", *IEEE Access*, vol. 8, pp. 49920-49932, March 2020.
DOI: https://doi.org/10.1109/ACCESS.2020.2980143

[4] G. Pataki, "Teaching Integer Programming Formulations Using The Traveling Salesman Problem", *SIAM Review*, Vol. 45, No. 1, pp. 116–123, 2003.
DOI: https://doi.org/10.1137/S00361445023685

[5] D. Whitley, "A genetic algorithm tutorial", *Statistics and Computing*, vol. 4, no. 2, pp. 65-85, June 1994.
DOI: https://doi.org/10.1007/BF00175354

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, Vol. 220, No. 4598, pp. 671-680, May 1983.
DOI: https://doi.org/10.1126/science.220.4598.671

[7] F. Glover, "Tabu Search: part I", *ORSA Journal on Computing*, Vol. 1, pp. 190-206, August 1989.
DOI: https://doi.org/10.1287/ijoc.1.3.190

[8] H. Youssef, S. M. Sait, and H. Adiche, "Evolutionary algorithms, simulated annealing and tabu search: a comparative study", *Engineering Applications of Artificial Intelligence,* Vol.14, No. 2, pp. 167-181, April 2001.
DOI: https://doi.org/10.1016/S0952-1976(00)00065-8

[9] A.J. Umbarkar1 and P.D.Sheth, "Crossover operators in genetic algorithm: A review", *ICTACT Journal on Soft Computing,* Vol. 6, No. 1, October, 2015.
DOI: https://doi.org/10.21917/ijsc.2015.0150

[10] L. Bracciale et al., CRAWDAD dataset roma/taxi (v. 2014-07-17), Jul 2014. https://crawdad.org/roma/taxi/20140717