


Article

Location and Time Aware Multitask Allocation in Mobile Crowd-Sensing Based on Genetic Algorithm

Aridegbe A. Ipaye ¹, Zhigang Chen ^{1,*}, Muhammad Asim ¹ , Samia Allaoua Chelloug ^{2,*} , Lin Guo ¹, Ali M. A. Ibrahim ¹ and Ahmed A. Abd El-Latif ^{3,4,*} 

¹ School of Computer Science and Engineering, Central South University, Changsha 410083, China; aa.ipaye@hotmail.com (A.A.I.); asimpk@csu.edu.cn (M.A.); guolincsu@csu.edu.cn (L.G.); alimaher_912@hotmail.com (A.M.A.I.)

² Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

⁴ Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt

* Correspondence: czg@csu.edu.cn (Z.C.); SACHelloug@pnu.edu.sa (S.A.C.); aabdellatif@psu.edu.sa (A.A.A.E.-L.)

Abstract: Mobile crowd-sensing (MCS) is a well-known paradigm used for obtaining sensed data by using sensors found in smart devices. With the rise of more sensing tasks and workers in the MCS system, it is now essential to design an efficient approach for task allocation. Moreover, to ensure the completion of the tasks, it is necessary to incentivise the workers by rewarding them for participating in performing the sensing tasks. In this paper, we aim to assist workers in selecting multiple tasks while considering the time constraint of the worker and the requirements of the task. Furthermore, a pricing mechanism is adopted to determine each task budget, which is then used to determine the payment for the workers based on their willingness factor. This paper proves that the task-allocation is a non-deterministic polynomial (NP)-complete problem, which is difficult to solve by conventional optimization techniques. A worker multitask allocation-genetic algorithm (WMTA-GA) is proposed to solve this problem to maximize the workers welfare. Finally, theoretical analysis demonstrates the effectiveness of the proposed WMTA-GA. We observed that it performs better than the state-of-the-art algorithms in terms of average performance, workers welfare, and the number of assigned tasks.

Keywords: crowd-sensing; genetic algorithm; incentive mechanism; multitask allocation; time-sensitive



Citation: Ipaye, A.A.; Chen, Z.; Asim, M.; Chelloug, S.A.; Guo, L.; Ibrahim, A.M.A.; Abd El-Latif, A.A. Location and Time Aware Multitask Allocation in Mobile Crowd-Sensing Based on Genetic Algorithm. *Sensors* **2022**, *22*, 3013. <https://doi.org/10.3390/s22083013>

Academic Editors: Alicia García-Holgado and Brij B Gupta

Received: 27 February 2022

Accepted: 10 April 2022

Published: 14 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the mobile crowd-sensing (MCS) paradigm has gained attention from many researchers. The main advantage of MCS is that the deployment of several static sensors over a large geographical region is reduced and replaced by willing users with the required sensing equipment in their smart devices [1]. Furthermore, due to its efficiency, MCS is used in many applications, such as environmental monitoring, prediction of traffic congestion, and map updating [2].

The mobile crowd-sensing mechanism consists of three components: the service requester, platform and workers. The entity that requires the results of the sensing tasks is known as the sensing task requester. Participants are smartphone employees compensated for their contributions and have the requisite sensors to execute the sensing duties. The platform's purpose is to broadcast task requirements to available participants, such as sensing time, reward, and location, to recruit the best participants for each job, and ultimately to collect and deliver the sensed data to the platform [3,4].

In practice, many MCS systems require that workers consider the time-sensitivity of the tasks. Aware of this requirement, the research community has studied task allocation

for the MCS system. Several works focus on task allocation [5–8]; however, they mainly concentrate on single task assignment, which has been an issue since the time that the worker would have taken to perform multiple tasks is wasted on waiting for the next task allocation from the platform after finishing the previous one.

In actuality, the majority of qualified workers located in the sensing task's geographical region can only do the tasks during their off-work hours (e.g., half an hour lunch break at work or when returning home after school). Furthermore, each task can be assigned to several workers with the caveat that each worker's assigned tasks must be completed before the worker's working time is up. As a result, while scheduling the completion order of assigned tasks, each worker must consider their working time. Each task should be given to workers who arrive at the task's location and have adequate working time to complete the assignment. As a result, the MCS system's supremacy is ensured by performing the sensing tasks within the stipulated time range.

Another critical factor to be considered is the payment of the worker. These workers use their valuable time and resources to complete the sensing task and, therefore, they need to be rewarded to encourage them. Unfortunately, in [9–11], the researchers focused mainly on improving the utility of the platform and neglected the profit of the workers.

This paper addresses task allocation for the workers and maximization of the total workers welfare considering the strict time constraint. We propose a multitask allocation scheme for time-sensitive tasks based on a genetic algorithm to answer this question.

The main contributions of our paper are summarized as follows:

- We presented a task allocation approach in the MCS system under a given set of allocated task budgets and workers' available time, which is formulated as a workers welfare maximization problem.
- We proposed an algorithm called Worker Multi-task Allocation (WMTA-GA) that enables workers to select the sensing tasks that are convenient for them and will maximize their profit.
- Simulation shows that WMTA-GA obtains better performance, generates adequate workers welfare, and allocates more tasks to the workers than the existing approaches.

The remainder of the paper is organized as follows. The related works are discussed in Section 2, and the system model together with the problem statement are presented in Section 3. We concentrate on the pricing mechanism and look into the proposed algorithm in Sections 4 and 5, respectively. Section 6 focuses on the experimental design and discussion of the results. Finally, the paper is concluded in Section 7.

2. Related Work

2.1. Task Allocation

Task allocation is an essential issue in the MCS system as it determines the performance of the system. Therefore, many studies in task allocation approaches have been investigated. In [12], the authors aim to solve the task-allocation problem by employing a bilateral preference matching game between the service requester and the workers in the MCS system to satisfy both the service requester and the workers. In some cases, a sensing task needs to be allocated to more than one worker to increase the workers' utility. The authors in [5] tackle this problem by designing an approach that uses both a greedy and an equilibrium-based method. In [13], a new protocol for assigning tasks was introduced to measure the workers' utility when performing sensing tasks in a crowd-sensing opportunistic network [2]. In [14], the purpose was to use an analytical hierarchy process to select workers from the user set. The worker had the option of accepting the sensing task, and if it chose to do so, then its updated reputation was to be determined based on its performance. There was also an issue of unbalanced task allocation that needed to be resolved to improve the performance of MCS. The purpose of [15] was to solve the problem of online user recruitment with consideration of the budget and time constraints.

However, for the allocation of time sensitive tasks, the above task allocation approaches will be ineffective. Therefore, the research in [16–20] focused on improving the MCS

system's reliability by ensuring that the sensing task is completed before the specified deadline. In [21], the authors proposed an approach such that the workers would spend less time performing the sensing task. This aim was achieved by partitioning each task into groups, and a travelling path for each worker was planned. The authors in [3,22,23] noted that most task-allocation research focused on assigning single tasks to workers. However, these workers usually had the capacity to complete multiple tasks. Due to this reason, a mechanism was introduced for the multitask allocation to maximize the platforms' utility. In [24], the users' preference based on content and context information was used to develop a worker-recruitment mechanism. The authors in [25] studied user preference and the effect it had on task assignment and workers' long-term utility. The authors in [7] studied a destination-aware task-allocation strategy to maximize the number of completed tasks and perform tasks before the deadline. In [26], the authors determined which sensing task to complete according to the incomplete task-popularity statistics information. Some papers focused on collecting the sensed data from the workers and delivering it to the platform. In [27], a mechanism was introduced to transmit sensed data to the data requester according to the time constraint. In [28], the researchers aimed to collect sensed data from workers that have high coverage properties of MCS fulfilled. The authors in [29] proposed a data-collection path-planning scheme that employed either cellular transmission or an opportunistic network. S. Akter et al. [30] investigated task allocation in time-sensitive MCS based on A-DQN. The task allocation problem was considered a travelling salesman problem, and the aim was to minimize the total cost of completing the tasks subject to time constraints.

2.2. Incentive Mechanism

It is important that the workers assigned to the sensing tasks are compensated, as these workers consume their computational resources and time, and are at risk of private information leakage. Therefore, an incentive mechanism must be presented to encourage workers to participate in the MCS system [31–34]. In [35], a truthful incentive approach was introduced, which used the workers' sensing performance and reputation to determine the payment they would receive after completing the allocated tasks. In [36,37], the tasks were allocated such that the social welfare was maximized to ensure that there is fairness in the system. In [38], the semi-Markov model was used to obtain the positions of the workers, which was then used to determine workers with the least distance and bidding price. The authors in [39] updated the reward of each task in each iteration to take advantage of the workers' interest in a particular task. However, this research on incentive mechanisms neglected the fact that maximizing the combined utility of the workers will encourage them to participate in time-sensitive tasks in MCS and increase the chances that they will be optimistic about performing the future time-sensitive tasks as well.

2.3. Summary

According to the research mentioned above, most authors failed to consider maximizing the workers welfare for time-sensitive tasks and its benefit for MCS. For instance, in [30], the authors proposed A-DQN to determine the task allocation subject to time constraints. However, they aimed to minimize the total reward given to the users selected to perform the sensing task. Therefore, by considering the task allocation subject to time constraints of the task and workers, this paper aims to maximize the total worker welfare by solving the (WMTA-GA) multitask allocation problem.

3. System Model and Problem Statement

In this section, the system model of the MCS system followed by the problem statement are presented. Table 1 shows the main notations used in this paper.

Table 1. Model Parameters .

System Model Parameter	
Worker Parameter	
W	Set of all workers
n	Number of workers
w_i	worker i
c_i	Sensing cost of worker w_i
p_i	Current position of worker w_i
wt_i	Available working time of worker w_i
P_i^t	Power level of worker w_i
v_i	Velocity of worker w_i
R_i	Payment given to worker w_i
$will_i$	Willingness of worker w_i
WF_{ij}	Willingness factor of worker w_i for task t_j
Wrt_{ij}	Response time of worker w_i for task t_j
P_i^r	Residual power of worker w_i
WT_i	Task t_j popularity
SP_{ij}	Specification of worker w_i for task t_j
$SP_{iT_t^{w_i}}$	Specification of worker w_i for $T_t^{w_i}$
Pr_i	Profile for worker i
Task Parameter	
T	Set of all tasks
m	Number of task
$will_j$	Willing threshold of task t_j
t_j	Task j
te_j	End time of task t_j
ts_j	Start time of task t_j
t_{d_j}	Sensing duration of task t_j
bt_j	Budget allocated to task t_j
Pr_j	Profile for task t_j
$T_t^{w_i}$	Set of interested tasks for worker w_i
TD_j	Task Demand for task t_j
Other	
B	Total budget
$D(w_i, t_j)$	Distance between worker w_i location and the task t_j position

3.1. System Model

Figure 1 depicts the MCS system model and the following definitions describe the components of the model:

Definition 1 (Requester). *The requester is an entity; which can be an individual or a company, that requires certain sensing tasks $T = \{t_1, t_2, t_3, \dots, t_n\}$ to be performed. The requester communicates with the platform about its required sensing tasks and provides a budget for the set of tasks to be performed.*

Definition 2 (Task). *Each task $t_j \in T$ is located in a specific geographical position p_j . The profile of task t_j is denoted by $Pr_j = \{p_j, bt_j, te_j, ts_j, will_j\}$ which consists of budget bt_j , sensing-duration(te_j, ts_j), and willing threshold $will_j$.*

Definition 3 (Platform). The main responsibility of the platform is to assign tasks to the workers having the necessary sensor component and who are willing to perform the specific task. When a worker w_i has available working time, it informs the platform, and the worker will select the set of tasks that they can perform. All the sensed data are then collected by the platform and sent to the corresponding requester.

Definition 4 (Worker). The worker $w_i \in W$ is a smart-device user with enough energy that participates in performing sensing tasks. A worker can participate in a sensing task only if the task's sensing duration fits into the worker's available working time wt_i and the worker's willingness factor for the task is above the threshold. Each worker is associated with a specification for each task t_j which is denoted by $SP_{ij} = \{p_i, wt_i, c_{ij}, v_i\}$ which consists of current position p_i , available working time wt_i , sensing cost c_{ij} , and velocity v_i . The goal of workers is to select tasks that will maximize their utility. The worker w_i then includes these selected tasks to its set of interested tasks $T_t^{w_i} = \{t_1, t_2, t_3, \dots, t_{f_j}\}$. After selecting these tasks, the worker sends to the platform its profile $Pr_i = \{T_t^{w_i}, SP_{iT_t^{w_i}}\}$.

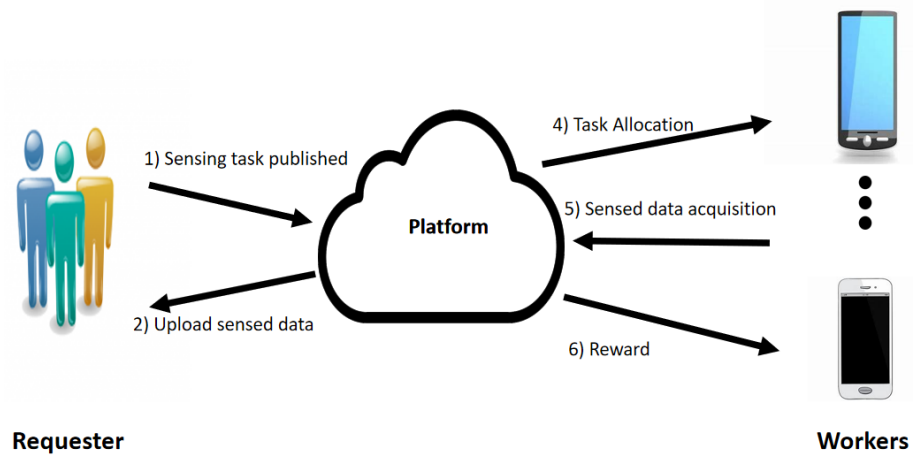


Figure 1. MCS System Model.

3.2. Problem Statement

From the description of the system model, the task-allocation problem is formulated as follows:

$$\max \sum_{i=1}^n (R_i \cdot a_i - c_i) \quad (1)$$

s.t.

$$\sum_{j=1}^m D(w_i, t_j) \leq v_i \cdot wt_i, \forall t_j \in T_t^{w_i} \quad (2)$$

$$D(w_i, t_j) \leq v_i \cdot (te_j - ts_j), t_j \in T_t^{w_i} \quad (3)$$

$$WF_{ij} \geq will_j, \forall t_j \in T \quad (4)$$

$$a_{ij} = 1 \text{ or } a_{ij} = 0, \forall w_i \in W, \forall t_j \in T \quad (5)$$

$$\sum_{i=1}^n a_{ij} \leq 1, \forall t_j \in T \quad (6)$$

where R_i denotes the payment given to w_i , the distance between the worker's location and the location of task $t_j \in T_t^{w_i}$ is denoted by $D(w_i, t_j, T_t^{w_i})$, velocity is indicated by v and a_{ij} represents the task-assignment: if task t_j has been assigned to worker w_i then $a_{ij} = 1$ otherwise $a_{ij} = 0$.

Equation (2) indicates that the time that the worker w_i takes to move from their current location to the last task in $T_t^{w_i}$ should not exceed the working time wt_i . This constraint is to ensure that the worker can perform all the allocated tasks it has accepted from the platform

within its working time. Constraint Equation (3) ensures that each task is completed within the sensing duration, Equation (4) ensures workers with a high willingness factor are selected to perform task t_j , and Equation (5) ensures that each task is allocated to only one suitable worker.

The willingness-factor WF_{ij} predicts the performance of workers for the assigned sensing tasks. The WF_{ij} is obtained by considering worker w_i response-time Wrt_{ij} (i.e., the time the worker begins a sensing task) and the residual power of the workers' smart device p_i^r . For example, when the worker's commencing time for a task is the same as the task's start time, then the response time of the worker is 0. This reflects positively on the workers' willingness to perform the task, and in this scenario, the willingness-factor is 1. From the above analysis, the willingness-factor is measured as follows:

$$WF_{ij} = \begin{cases} \min \left(\left[\frac{p_i^r}{p_i^r} \cdot \frac{\left(\sum_{i=1}^n \frac{Wrt_{ij}}{n} \right)}{Wrt_{ij}} \right], 1 \right), & Wrt_{ij} \neq 0 \\ 1, & Wrt_{ij} = 0 \end{cases}$$

The willingness factor is then normalized as follows,

$$NWF_{ij} = (WF_{ij}) / \left(\sum_{j=1}^m WF_{ij} \right) \quad (7)$$

and worker w_i is then rewarded with the following:

$$R_i = \left(\sum_{j=1}^m NWF_{ij} \right) \cdot bt_j \quad (8)$$

where bt_j is the budget allocated to task t_j . The calculation of the task budget is discussed in Section 4.

Theorem 1. *The multitask allocation problem is NP-complete.*

Proof. The multitask allocation problem is a combinatorial optimization problem that has a large solution space. We assume that there is Q number of tasks and s number of workers, and each worker can perform Q number of tasks. Since each worker can perform 2^Q of assigning tasks and for all s number of workers, there will be 2^{sQ} assigning method. In the worst case, there will be $2^{sQ} \cdot Q!$ assignment method if each worker develops $Q!$ ways of completing the tasks, increasing the solution space which implies that the proposed multitask problem is NP-complete. \square

4. Pricing Mechanism

In this section, a pricing mechanism is introduced to determine the budget allocation for each task.

The task budgets are allocated to each task according to the task demand TD_j which is a reflection of how important a task is in the MCS system. The budget is obtained by the weighted combination of the task-popularity W_{T_i} and the task-duration t_{d_j} which is defined by the following

$$t_{d_j} = t_{e_j} - t_{s_j} \quad (9)$$

The task popularity reflects how frequently workers visit a certain location and is indicated by the predicted number of workers in a given task location. The task demand for task t_j is determined as follows:

$$TD_j = (w_1 * W_{T_i}) + (w_2 * t_{d_j}) \quad (10)$$

where w_1 and w_2 are the weights satisfying ($w_1 + w_2 = 1$), the values which are determined by the MCS platform. However, in this paper, we assume that the sensing duration is more important than task popularity, and therefore we set the weights as $w_1 = 0.2$ and $w_2 = 0.8$.

The task demand is then normalized as follows,

$$NTD_j = (TD_j) / (\sum_{j=1}^m TD_j) \quad (11)$$

and therefore given the total budget B of all tasks, the budget allocated to task t_j is given by,

$$bt_j = (NTD_j) \cdot B \quad (12)$$

5. Multi-Task Allocation Algorithm

From the previous section, it was concluded that the task allocation solution space was too large, and traditional combinatorial optimization algorithms will be inefficient in handling the multitask allocation problem. In this section, we introduce the genetic algorithm (GA) [40] to solve the multitask allocation problem. GA has been applied to address various optimization problems in various research fields [41–44].

Genetic Algorithm for Multitask Allocation

In order to solve the studied problem, WMTA-GA adopted GA for multitask allocation as presented in Algorithm 1. First, the population is initialized with K chromosomes (line 1). In each iteration, the best chromosome is selected based on fitness (lines 2–9). The fitness presented in line 4 represents the objective problem that is used to evaluate each chromosome's performance. The current best chromosome is updated in line 5, and the selection operator, crossover and mutation operator, and re-establish operator are introduced in (lines 6–8), respectively. The while loop continues until all the workers have been assigned to a specified set of tasks. Finally, the global best chromosome is updated in line 11. The initial population is generated from Algorithm 2, which uses the greedy algorithm. The re-establishing operator is presented in Algorithm 3.

Algorithm 1 WMTA-GA.

Input: Worker set W and Task set T

Output: Worker set S_W assigned to the task $t_j, \forall t_j \in T$

- 1: Pop_{size}, max_G, G
 - 2: Initialize population initial G with K chromosomes
 - 3: **while** $G \leq max_G$ **do**
 - 4: Evaluate the fitness of all chromosomes in Pop_G
 - 5: Update the current best chromosome
 - 6: $Pop'_G \leftarrow$ Selector operator on Pop_G
 - 7: $Pop''_G \leftarrow$ Crossover and mutation operator on Pop'_G to produce update N chromosomes
 - 8: $Pop_{G+1} \leftarrow$ Re-establish operator on Pop''_G
 - 9: $G \leftarrow G + 1$
 - 10: **end while**
 - 11: **return** best chromosome (task-worker allocation)
-

Algorithm 2 Initial population.**Input:** Worker set W and Task set T **Output:** Chromosome C (initial task-worker allocation)

```

1: Candidate worker set  $WS \leftarrow W$ 
2: Unassigned task set  $TS \leftarrow T$ 
3: while  $WS \geq 0$  do
4:   Randomly select a worker  $w \in WS$ 
5:   Set counter  $q = 0$ 
6:   while  $q \leq$  number of tasks do
7:     Randomly select a task  $t_j \in TS$ 
8:      $C \leftarrow$  append  $t_j$  to the end of the genetic segment allocated for  $w$ 
9:     if  $C$  is a supported chromosome then
10:       $TS \leftarrow TS - t_j$ 
11:     end if
12:      $q \leftarrow q + 1$ 
13:   end while
14:    $WS \leftarrow WS - w$ 
15: end while

```

Algorithm 3 Re-establishing Operator.**Input:** A unsupported chromosome C **Output:** A unsupported chromosome C_n

```

1: Case 1: Unsupported genetic segment
2: for genetic segment  $C(w) \in C$  do
3:   if gene  $g(i) \in C(w)$  violates constraints Equations (2)–(4)
4:      $C(w) \leftarrow$  select subset of genes in the genetic segment that satisfy the aforementioned
       constraints
5:   end if
6: end for
7: Case 2: Repeated task allocation
8: for each task  $t_j$  do
9:    $RGS \leftarrow$  all genetic segments containing the same task  $t_j$ 
10:   $GSB \leftarrow$  the genetic segment  $\in RGS$  with the maximum fitness
11:  for the remaining genetic segment in  $RGS$  do
12:    Remove task  $t_j$ 
13:  end for
14: end for
15: for genetic segment  $C(w) \in C$  do
16:  Update worker  $w$  position and remaining working time
17:   $T_a \leftarrow$  the tasks that  $w$  can perform with remaining available time
18:  while  $T_a \geq 0$  do
19:    Randomly select a task  $t'_j$  from  $T_a$ 
20:     $C'(w) \leftarrow C(w) \cap t'_j$ 
21:    if  $C'(w)$  is a supported chromosome then
22:       $C(w) \leftarrow C(w) \cap C'(w)$ 
23:    end if
24:     $T_a \leftarrow T_a - t'_j$ 
25:  end while
26: end for

```

1. Chromosome Representation and Fitness Evaluation

The main aim of WMTA-GA is to find a solution that indicates task-worker allocation and task completion order. Since the number of chromosome solutions is uncertain, we use a matrix structure to represent them. Each chromosome consists of b genetic segments representing the registered workers. The genetic segments are identified

using the index of their position in the chromosome. Each genetic segment is divided into genes that represent the tasks assigned to the worker. According to the constraints Equations (2)–(6), we classify the chromosomes into supported chromosomes meaning that the multitask allocation satisfies the constraints and unsupported chromosomes when the constraints are not satisfied.

Since each chromosome represents all the workers and their assigned tasks, the whole population represents the set of workers and their assigned tasks. The fitness function was applied to the chromosomes in a population to evaluate the quality of the population. Since the objective function $R_i = (\sum_{j=1}^m NWF_{ij}) \cdot bt_j$ is to maximize the workers welfare, then it is used as an indicator of each chromosome's fitness level (i.e., the amount of workers welfare generated by a chromosome is directly related to its fitness level). Therefore, the fitness function of a chromosome C_t is calculated as:

$$Fit(C_t) = \sum_{i=1}^n R_i \cdot a_i - c_i \quad (13)$$

This implies that for a population in a given generation defined as $G = \{C_1, C_2, C_3, \dots, C_l\}$, the task assignment strategy of chromosome C_t ($1 \leq t \leq l$) is the row vector $A_t = \{a_1, a_2, a_3, \dots, a_p\}$, where a_i ($1 \leq i \leq p$) can be either "1" if the task is allocated to the worker, or otherwise a_i is "0".

2. Selection Operator

The selector operator aims to allow the chromosomes with the higher fitness level to proceed to the next generation while taking into consideration the population diversity; however, chromosomes with lower fitness values can contain acceptable genetic segment properties. To guarantee that top-quality chromosomes are accepted into the next generation, we firstly select elitist chromosomes to ensure that chromosomes with high fitness levels do not disappear. To obtain this type of chromosome, we arranged the parent population chromosomes in descending order of fitness, and randomly picked out a specified number of chromosomes (e.g., first to fifth) and passed it on to the next generation. Afterward, tournament selection [3] is applied to the remaining chromosomes. In each generation, a certain number of chromosomes are compared and the chromosome with the highest fitness level proceeds to the next generation. This continues until the required amount is obtained.

3. Crossover and Mutation Operator

In the crossover operation, a new generation of sophisticated chromosomes is generated by recombining the parent chromosomes through crossover operation. Initially, two-parent chromosomes from the elite set of chromosomes and a single chromosome from the remaining chromosomes are selected. Then the fitness value of the genetic segments of the chromosomes (elite and normal chromosomes that are to be combined by crossover operation) are compared using their fitness level, and the superior genetic segment is passed to the new chromosome. In the process of performing this operation, the entire population is evolved and the diversity of the population is improved.

It is important to realize that the new chromosome generated from the crossover operation might not follow the constraint conditions and therefore become an unsupported chromosome. To avoid local optimum and enhance the population diversity, the mutation operation is applied to the chromosome.

For the chromosome mutation to occur, we select two genes from two genetic segments in the same chromosome and swapped them to generate mutated chromosomes. However, during the mutation process, the status of some chromosomes may change and become unsupported.

4. Re-establishing Operator

After the crossover and mutation operation, some supported chromosomes may convert to unsupported chromosomes and for that reason, we introduce a re-establishing operator. Initially, we confirm whether each genetic segment satisfies all the con-

straints. If a constraint is violated, then we select a subset of genes in the corresponding genetic segment that satisfies the constraints. Secondly, if a task is allocated more than once in a particular chromosome, we use the idea of the survival of the fittest to determine which of the tasks is to be retained. Finally, we consider the fact that some workers may have remaining time to perform some unallocated sensing tasks and determine which chromosome should be selected for this task such that the constraint remains satisfied.

6. Experimental Results

6.1. Experimental Settings

In this section, we introduce the experimental design used to analyze the proposed WMTA-GA algorithm. In the experiment, we assumed that both the number of sensing tasks and the number of workers range from 60 to 200. The maximum number of iterations (i.e., the maximum generation max_G) and the population size is set to 150 and 20, respectively. The worker's working time ranges from 1 to 50 (min), and the response time of the task ranges from 1 to 20 (min). The sensing area is divided into 100 locations, and the position of both the worker and task are randomly selected from one of the locations. We assume that the velocity of the worker is 50 km/h and the total budget allocated to the platform is 15,000. The willing threshold for each task is randomly selected from the range [0.1 to 0.7]. The sensing cost of the workers is within the range [100, 400]. The experiment was conducted using MatLab 2016a.

6.2. Performance Metric

We evaluate the approaches' performances by studying how various parameters (number of tasks and number of workers) affect the following metrics:

1. **Workers Welfare:** This metric is an indicator of how well an algorithm performs based on fulfilling the condition of maximizing all the workers' profit. The higher this profit, the better the performance of the algorithm.
2. **Number of Allocated Tasks:** This indicates how many tasks are allocated to the selected workers. When the number of allocated tasks is high, it shows that the large number of workers can complete the tasks and this improves the reliability of the platform.
3. **Average Performance w.r.t Varying Number of Tasks:** The average worker's welfare subject to the number of tasks in each algorithm is evaluated. This ratio assists in evaluating the performance of the algorithms.
4. **Average Performance w.r.t Varying Number of Workers:** The performance of each algorithm is compared by evaluating each of their average workers welfare with respect to the number of workers. A high ratio indicates a high-performing algorithm.

6.3. Baseline Approaches

1. **Greedy Worker Payoff-based Task Allocation (GWP):** In [17], the GWP approach was used to greedily select tasks with the highest reward in each iteration. We modified this algorithm to maximize the workers welfare, whereby each worker selects the tasks with the biggest reward and these workers are assigned to their selected tasks if the worker-task allocation satisfies the validity conditions. If the conditions are satisfied, then we exchange the location of the worker with the location of the current task, therefore changing the new initial location of the worker. This process continues until each worker's working time is finished.
2. **Greedy Worker Reward-Distance Ratio-GA (GWR-GA):** The second benchmark algorithm is the modified greedy payoff/distance-based task selection algorithm (G.P.D.A.) [17]. This approach has been modified to maximize the workers welfare by employing GA tasks with a high payoff/distance ratio to be assigned to workers, and GA is employed to obtain an optimal chromosome (i.e., worker-task allocation).

6.4. Result Analysis

In this section, the results obtained from WMTA-GA, GWR-GA and GWP are interpreted and analyzed.

- Impact of Varying the Number of Tasks

In Figure 2, we observe that the number of tasks is directly proportional to the workers welfare and that WMTA-GA outperforms GWR-GA and GWP. For instance, when the number of tasks was 100 and 140, the workers welfare for WMTA-GA was 3950 and 5605, respectively, which is much higher than the welfare obtained by GWR-GA (3742 and 5144) and GWP(3175 and 3863).

As shown in Figure 3, the ratio of the assigned tasks in WMTA-GA, GWR-GA, and GWP declines significantly with the increase in the number of tasks. However, when the number of tasks is 60 and 200, we see that the WMTA-GA recorded the highest ratio of assigned tasks, which are 0.6 and 0.26, respectively. Table 2 illustrates the performances of the three approaches and indicates that WMTA-GA performs better in comparison to GWP and GWR-GA. This is confirmed by their average utility value where WMTA-GA is 73.24 which is 13% and 68% higher than GWR-GA and GWP, respectively.

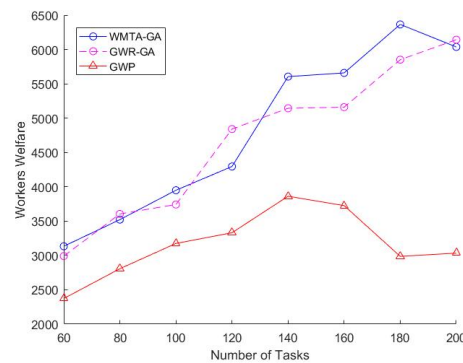


Figure 2. Impact of Varying Number of Tasks on Worker Welfare.

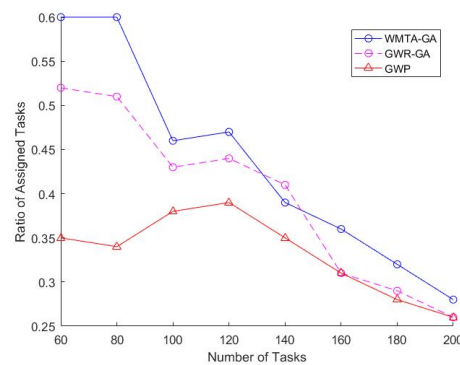


Figure 3. Impact of Varying Number of Tasks on Ratio of Assigned Tasks.

Table 2. Average performance w.r.t Varying Number of Tasks.

Workers' Welfare Average Performance Based on the Number of Tasks			
	WMTA-GA	GWR-GA	GWP
WMTA-GA	-	2.91	73.24
GWR-GA	-3	-	68
GWP	42	-40.60	-

- **Impact of Varying Number of Workers**

We investigate how varying the number of workers affects worker welfare, the number of task allocations, and the average performance of the workers. We set the range of workers from 60 to 90 and fixed the number of tasks to 200. In Figure 4, we observe that as the number of workers increases, the workers' welfare increases as well. The workers' welfare in WMTA-GA steadily increases compared to the other approaches. For example, when the number of workers ranges from 60 to 80, WMTA-GA's worker welfare is 3100 and 5300.

In Figure 5, we see that WMTA-GA provides the highest number of assigned tasks as compared to the other algorithms. When the number of workers is 60 and 80, WMTA-GA achieves 56 and 77 number of the assigned tasks.

The number of the assigned tasks increases as more tasks are successfully allocated, as observed in Figure 5. The WMTA-GA approach had the highest number of assigned tasks compared to other approaches for $n = 60$ to 90 , with the highest recorded number of assigned tasks of 84 at $n = 90$. GWR-GA follows the same trend as WMTA-GA because it achieved a similar result to WMTA-GA. However, GWP had the lowest overall number of assigned tasks and peaked at 68 assigned tasks at $n = 70$.

We observed from Table 3 that WMTA-GA has the best average performance as compared to GWR-GA and GWP, which is indicated by their average utility values. WMTA-GA's average utility value is 17.26% more than GWP, and 1.98% more than GWR-GA. The second-best performing algorithm is GWR-GA, with an average utility value higher than GWP by 15 percent. Compared to WMTA-GA and GWR-GA, the GWP had a minor favourable result, with an average performance of -15% and -13.3% for WMTA-GA and GWR-GA, respectively.

Table 3. Average performance w.r.t Varying Number of Workers.

Workers Welfare Average Performance Based on Number of Workers			
	WMTA-GA	GWR-GA	GWP
WMTA-GA	-	1.98	17.26
GWR-GA	-2	-	15
GWP	-15	-13.03	-

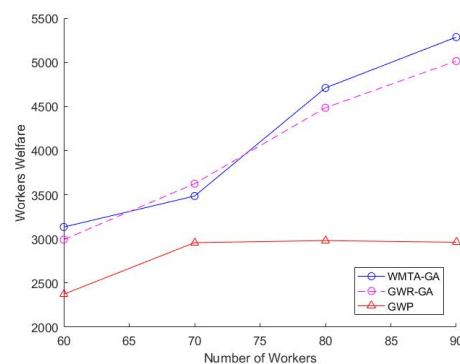


Figure 4. Impact of Varying Number of Workers on Worker Welfare.

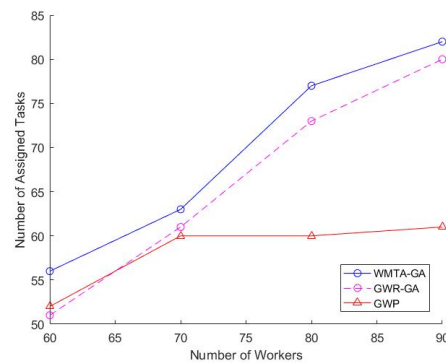


Figure 5. Impact of Varying Number of Workers on Number of Assigned Tasks.

6.5. Discussion

In Section 6.3, WMTA-GA achieves the overall best results for the workers welfare, ratio of assigned tasks and number of assigned tasks. The explanation for this is that the workers using the WMTA-GA approach have a higher chance of being correctly assigned to sensing tasks with greater reward due to the re-establishing operator. The function of the re-establishing operator is to convert the unsupported chromosome (i.e., task-worker allocation) to the supported chromosome that fulfils the MCS constraints Equations (2)–(4). For instance, in Figure 2, WMTA-GA recorded high worker welfare as a value of m increases, while GWR-GA followed a similar trend but with lower values for the workers welfare. On the other hand, GWP, which uses the greedy algorithm, had the lowest worker welfare throughout the graph. In Figure 3, the ratio of assigned tasks declined for all approaches as the number of tasks increased. This result is due to the resources of the workers in the MCS system being limited, so newly introduced tasks will be less likely to be assigned as the workers will have exhausted their resources from the previous set of tasks. Nevertheless, the WMTA-GA displayed a better ratio of assigned tasks than the remaining approaches due to the assistance of the re-establishing operator.

The advantages of using the re-establishing operator for the various number of workers are presented in Figures 4 and 5. In Figure 4, WMTA-GA accomplished the highest steady increase in worker welfare for the varying number of workers. The stronger performance of WMTA-GA can be attributed to the following reasons: (1) more tasks are being successfully assigned to the workers in the MCS system, and, as a result, the selected workers gain more profit. Due to similar reasons, WMTA-GA in Figure 5 experienced the highest number of assigned tasks for the range $m = (60 \text{ to } 90)$ and (2). The average percent of supported and unsupported chromosomes in each iteration is 30% and 70%, respectively. The unsupported chromosomes are then reduced and converted into supported chromosomes by using the re-establishing operator.

Therefore, we observed that the approaches that do not utilize a re-establishing operator will have lower performance due to the unsupported chromosome or task allocation. Therefore, we can conclude that WMTA-GA is superior to GWR-GA and GWP.

7. Conclusions

In this paper, we have studied a multitask allocation in a mobile crowd-sensing environment. The problem is formulated as an optimization problem. We first proved that the problem is NP-complete, which is hard to be solved by traditional methods. Then, we proposed a worker multitask allocation-genetic algorithm (WMTA-GA) to solve this problem. WMTA-GA is used to optimize the workers' welfare such that the multiple sensing task is allocated to a worker with the condition that the time and willingness constraints are fulfilled. Specifically, these conditions include that the allocated tasks are performed during the working time of the worker, and the worker's willingness factor for each assigned task is above the tasks' willingness threshold. For each selected worker, we utilized the task budget and the worker's willingness factor to determine the payment for

the worker. Simulation results show that the proposed WMTA-GA approach outperforms other approaches in terms of average performance, worker's welfare, and ratio and number of assigned tasks.

Author Contributions: Conceptualization, A.A.I., Z.C., M.A., L.G., A.M.A.I. and A.A.A.E.-L.; methodology, A.A.I., Z.C., M.A., L.G.; software, A.A.I., Z.C., M.A., L.G., validation, A.A.I., Z.C., M.A., S.A.C., L.G., A.M.A.I. and A.A.A.E.-L.; formal analysis, A.A.I., Z.C.; investigation, A.A.I., Z.C.; resources, A.A.I., Z.C., M.A., A.M.A.I.; data curation, A.A.I., Z.C., M.A., A.M.A.I.; writing—original draft preparation, A.A.I., Z.C., M.A.; writing—review and editing, A.A.I., Z.C., M.A. and A.A.A.E.-L.; visualization, A.A.I., Z.C., M.A., S.A.C.; supervision, Z.C. and A.A.A.E.-L.; project administration, A.A.I., Z.C., M.A.; funding acquisition, Z.C., All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62102456 and Grant 71633006, and in part by the project funded by the China Postdoctoral Science Foundation under Grant 2021TQ0369. This work was also supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R239), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Ahmed A. Abd El-Latif acknowledges the financial support of EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, Z.; Cao, J.; Liu, X. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, 26 April–1 May 2015; pp. 2542–2550. [[CrossRef](#)]
2. Karaliopoulos, M.; Telelis, O.; Koutsopoulos, I. User recruitment for mobile crowdsensing over opportunistic networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, 26 April–1 May 2015; pp. 2254–2262. [[CrossRef](#)]
3. Li, X.; Zhang, X. Multi-Task Allocation Under Time Constraints in Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1494–1510. [[CrossRef](#)]
4. Asim, M.; Mashwani, W.K.; El-Latif, A.A. Energy and Task Completion Time Minimization Algorithm for UAVs-Empowered MEC system. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100698. [[CrossRef](#)]
5. Zhao, Y.; Guo, J.; Chen, X.; Hao, J.; Zhou, X.; Zheng, K. Coalition-based Task Assignment in Spatial Crowdsourcing. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; IEEE Computer Society: Los Alamitos, CA, USA, 2021; pp. 241–252. [[CrossRef](#)]
6. Wang, J.; Wang, F.; Wang, Y.; Wang, L.; Qiu, Z.; Zhang, D.; Guo, B.; Lv, Q. HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing. *IEEE Trans. Mob. Comput.* **2020**, *19*, 598–611. [[CrossRef](#)]
7. Zhao, Y.; Zheng, K.; Li, Y.; Su, H.; Liu, J.; Zhou, X. Destination-Aware Task Assignment in Spatial Crowdsourcing: A Worker Decomposition Approach. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 2336–2350. [[CrossRef](#)]
8. Wei, X.; Wang, Y.; Tan, J.; Gao, S. Data Quality Aware Task Allocation with Budget Constraint in Mobile Crowdsensing. *IEEE Access* **2018**, *6*, 48010–48020. [[CrossRef](#)]
9. Chen, Y.; Li, Z.; Yang, B.; Naia, K.; Li, K. A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing. *Future Gener. Comput. Syst.* **2020**, *108*, 273–287. [[CrossRef](#)]
10. Dai, M.; Su, Z.; Wang, Y.; Xu, Q. Contract Theory Based Incentive Scheme for Mobile Crowd Sensing Networks. In Proceedings of the 2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT), Tangier, Morocco, 20–22 June 2018; pp. 1–5.
11. Singla, A.; Krause, A. *Truthful Incentives in Crowdsourcing Tasks Using Regret Minimization Mechanisms*; Association for Computing Machinery: New York, NY, USA, 2013. [[CrossRef](#)]
12. Yucel, F.; Yuksel, M.; Bulut, E. Coverage-Aware Stable Task Assignment in Opportunistic Mobile Crowdsensing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3831–3845. [[CrossRef](#)]
13. Yucel, F.; Bulut, E. *Location-Dependent Task Assignment for Opportunistic Mobile Crowdsensing*; IEEE Press: Piscataway, NJ, USA, 2020. [[CrossRef](#)]

14. Xiong, J.; Chen, X.; Tian, Y.; Ma, R.; Chen, L.; Yao, Z. MAIM: A Novel Incentive Mechanism Based on Multi-Attribute User Selection in Mobile Crowdsensing. *IEEE Access* **2018**, *6*, 65384–65396. [[CrossRef](#)]
15. Liu, W.; Yang, Y.; Wang, E.; Wu, J. Dynamic User Recruitment with Truthful Pricing for Mobile CrowdSensing. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1113–1122. [[CrossRef](#)]
16. Huang, Y.; Chen, H.; Ma, G.; Lin, K.; Ni, Z.; Yan, N.; Wang, Z. OPAT: Optimized Allocation of Time-Dependent Tasks for Mobile Crowdsensing. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2476–2485. [[CrossRef](#)]
17. Wang, Z.; Tan, R.; Hu, J.; Zhao, J.; Wang, Q.; Xia, F.; Niu, X. Heterogeneous Incentive Mechanism for Time-sensitive and Location-dependent Crowdsensing Networks with Random Arrivals. *Comput. Netw.* **2017**, *131*, 96–109. [[CrossRef](#)]
18. Akter, S.; Yoon, S. DaTask: A Decomposition-Based Deadline-Aware Task Assignment and Workers' Path-Planning in Mobile Crowd-Sensing. *IEEE Access* **2020**, *8*, 49920–49932. [[CrossRef](#)]
19. Cheung, M.H.; Hou, F.; Huang, J. Delay-Sensitive Mobile Crowdsensing: Algorithm Design and Economics. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2761–2774. [[CrossRef](#)]
20. Liu, W.; Yang, Y.; Wang, E.; Han, Z.; Wang, X. Prediction Based User Selection in Time-Sensitive Mobile Crowdsensing. In Proceedings of the 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), San Diego, CA, USA, 12–14 June 2017; pp. 1–9. [[CrossRef](#)]
21. Wang, H.; Zhao, D.; Ma, H.; Ding, L. MB-GVNS: Memetic Based Bidirectional General Variable Neighborhood Search for Time-Sensitive Task Allocation in Mobile Crowd Sensing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2219–2229. [[CrossRef](#)]
22. Guo, B.; Liu, Y.; Wu, W.; Yu, Z.; Han, Q. ActiveCrowd: A Framework for Optimized Multitask Allocation in Mobile Crowdsensing Systems. *IEEE Trans. Hum.-Mach. Syst.* **2017**, *47*, 392–403. [[CrossRef](#)]
23. Yang, S.; Qi, X.; Wu, F.; Gao, X.; Chen, G. Bandit User Selection Algorithm for Budgeted and Time-Limited Mobile Crowdsensing. In Proceedings of the GLOBECOM 2017–2017 IEEE Global Communications Conference, Singapore, 4–6 December 2017; pp. 1–6. [[CrossRef](#)]
24. Wang, Z.; Zhao, J.; Hu, J.; Zhu, T.; Wang, Q.; Ren, J.; Li, C. Towards Personalized Task-Oriented Worker Recruitment in Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *20*, 2080–2093. [[CrossRef](#)]
25. Yucel, F.; Bulut, E. Time-dependent Stable Task Assignment in Participatory Mobile Crowdsensing. In Proceedings of the 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, 16–19 November 2020; pp. 433–436. [[CrossRef](#)]
26. Cheung, M.H.; Hou, F.; Huang, J.; Southwell, R. Distributed Time-Sensitive Task Selection in Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *20*, 2172–2185. [[CrossRef](#)]
27. Kortoçi, P.; Mehrabi, A.; Joe-Wong, C.; Di Francesco, M. Incentivizing Opportunistic Data Collection for Time-Sensitive IoT Applications. In Proceedings of the 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Rome, Italy, 6–9 July 2021; pp. 1–9. [[CrossRef](#)]
28. Yang, J.; Fu, L.; Yang, B.; Xu, J. Participant Service Quality Aware Data Collecting Mechanism with High Coverage for Mobile Crowdsensing. *IEEE Access* **2020**, *8*, 10628–10639. [[CrossRef](#)]
29. Majeed, D.M.; Zhang, L.; Shi, K. Optimal Data Collection for Mobile Crowdsensing over Integrated Cellular and Opportunistic Networks. *IEEE Access* **2020**, *8*, 157270–157283. [[CrossRef](#)]
30. Akter, S.; Dao, T.N.; Yoon, S. Time-Constrained Task Allocation and Worker Routing in Mobile Crowd-Sensing Using a Decomposition Technique and Deep Q-Learning. *IEEE Access* **2021**, *9*, 95808–95822. [[CrossRef](#)]
31. Xu, J.; Fu, J.R.; Yang, D.; Xu, L.J.; Wang, L.; Li, T. FIMI: A Constant Frugal Incentive Mechanism for Time Window Coverage in Mobile Crowdsensing. *J. Comput. Sci. Technol.* **2017**, *32*, 919–935. [[CrossRef](#)]
32. Sun, X.; Yang, X.; Wang, C.; Wang, J. A Novel User Selection Strategy with Incentive Mechanism Based on Time Window in Mobile Crowdsensing. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 1–13. [[CrossRef](#)]
33. Gao, H.; Liu, C.H.; Tang, J.; Yang, D.; Hui, P.; Wang, W. Online Quality-Aware Incentive Mechanism for Mobile Crowd Sensing with Extra Bonus. *IEEE Trans. Mob. Comput.* **2019**, *18*, 2589–2603. [[CrossRef](#)]
34. Wang, Y.; Cai, Z.; Zhan, Z.H.; Gong, Y.J.; Tong, X. An Optimization and Auction-Based Incentive Mechanism to Maximize Social Welfare for Mobile Crowdsourcing. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 414–429. [[CrossRef](#)]
35. Xu, C.; Si, Y.; Zhu, L.; Zhang, C.; Sharif, K.; Zhang, C. Pay as How You Behave: A Truthful Incentive Mechanism for Mobile Crowdsensing. *IEEE Internet Things J.* **2019**, *6*, 10053–10063. [[CrossRef](#)]
36. Yang, S.; Jiang, W.; Duan, J.; Huang, Z.; Lu, J. Max-Min Fairness Multi-task Allocation in Mobile Crowdsensing. In *Machine Learning for Cyber Security*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 164–179.
37. Liu, H.; Jiang, W.; Yang, S.; Lu, J.; Ning, D. Multi-term Multi-task Allocation for Mobile Crowdsensing with Weighted Max-Min Fairness. In *Machine Learning for Cyber Security*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 393–404.
38. Wang, Z.; Zhu, J.; Li, D. Prediction Based Reverse Auction Incentive Mechanism for Mobile Crowdsensing System. In *Combinatorial Optimization and Applications*; Springer: Cham, Switzerland, 2019; pp. 541–552. [[CrossRef](#)]
39. Wang, Z.; Hu, J.; Zhao, J.; Yang, D.; Chen, H.; Wang, Q. Pay On-Demand: Dynamic Incentive and Task Selection for Location-Dependent Mobile Crowdsensing Systems. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–5 July 2018; pp. 611–621. [[CrossRef](#)]
40. Holland, J. Erratum: Genetic Algorithms and the Optimal Allocation of Trials. *SIAM J. Comput.* **1974**, *3*, 326. [[CrossRef](#)]

41. Asim, M.; Wang, Y.; Wang, K.; Huang, P.Q. A Review on Computational Intelligence Techniques in Cloud and Edge Computing. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 742–763. [[CrossRef](#)]
42. Asim, M.; Mashwani, W.K.; Belhaouari, S.B.; Hassan, S. A Novel Genetic Trajectory Planning Algorithm with Variable Population Size for Multi-UAV-Assisted Mobile Edge Computing System. *IEEE Access* **2021**, *9*, 125569–125579. [[CrossRef](#)]
43. Asim, M.; Mashwani, W.K.; Habib, S.; Belhaouari, S.B. An evolutionary trajectory planning algorithm for multi-UAV-assisted MEC system. *Soft Comput.* **2021**. [[CrossRef](#)]
44. Asim, M.; Abd El-Latif, A.A. Intelligent computational methods for multi-unmanned aerial vehicle-enabled autonomous mobile edge computing systems. *ISA Trans.* **2021**. [[CrossRef](#)]