Research Paper:

Multi-team Formation System for Collaborative Crowdsourcing

Ryota Yamamoto and Kazushi Okamoto

Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
E-mail: kazushi@uec.ac.jp
[Received 2000/00/00; accepted 2000/00/00]

For complex crowdsourcing tasks that require collaboration between multiple individuals, teams should be formed by considering both worker compatibility and expertise. Furthermore, the nature of crowdsourcing dictates the budget for tasks and workers' remuneration, and excessively large team sizes may reduce collaborative performance. To address these challenges, we propose a heuristic optimization algorithm that leverages social network information to simultaneously form teams with optimized worker compatibility for multiple tasks. In our approach, historical collaboration is represented as a social network, where the edge weights correspond to explicit ratings of worker compatibility. In a simulation experiment using synthetic data, we applied Gaussian process regression to examine the relationship between eight experimental parameters and evaluation values, thereby analyzing the output of the proposed algorithm. To generate the necessary data for regression, we ran the proposed algorithm with experimental parameters that were sequentially estimated using Bayesian optimization. Our experiments revealed that the evaluation values were extremely low when the team size limit, the degree mean of the social network, and the task budget were set to low values. The results also indicate that the proposed algorithm outperformed the hill-climbing method under almost all experimental conditions. In addition, the highest evaluation values were achieved when the simulated annealing temperature decrease rate was approximately 0.9, while smoothing the objective function proved ineffective.

Keywords: crowdsourcing, social network, combinatorial optimization, metaheuristics

1. Introduction

Crowdsourcing is a model in which individuals or organizations delegate tasks to a diverse group of workers recruited via digital platforms. In this model, it is essential to manage workers with diverse expertise, depending on the scale and nature of the tasks [1]. In particular, collaborative crowdsourcing, in which skilled workers form teams to tackle complex tasks such as software development and translation, has garnered significant attention. Collaborative teams are expected to have synergistic effects, such as increasing work motivation and complementing each other's expertise [2, 3].

To form efficient collaborative teams in crowdsourcing, compatibility is key to ensure easy communication. Crowd worker teams often consist of individuals from various cultures, workplaces, and time zones, making it challenging to foster belonging and commitment, which can reduce productivity [4]. In team formation, a lack of team communication generally leads to project failure [5], and team members who have prior experience working with many co-workers are often more communicative [6].

In this study, we address the problem of team formation with optimized team compatibility for crowdsourcing. Team compatibility refers to the strength of social network connections within the team and is quantified using a social network representing past collaborative experiences. The nodes in the graph represent workers, and the edges indicate whether past collaboration occurred. When forming a team for a complex task in crowdsourcing, it is important to consider the following factors:

- Skill levels: Skill-level requirements should be defined. The level required for each task depends on the type of skill, and each worker has different levels of expertise.
- Hiring costs and budget: The hiring cost of each worker and the budget for each task should be considered to form teams in a crowdsourcing system.
- Team size: Kenna and Berche [7] found that increasing team size beyond a certain point does not improve team quality. Therefore, team sizes should be capped.
- Worker workload: It is unrealistic for a worker to take on multiple complex tasks simultaneously.

Therefore, each worker should be limited to handling only one task at a time.

This study establishes these as the requirements for the team formation problem. Applying the team formation algorithm that generates a single team for each task sequentially, in turn, may result in teams that do not satisfy the requirements of tasks to be processed later, or in teams that are less compatible with each other. Therefore, an algorithm capable of concurrently forming teams for multiple tasks is required.

The problem of forming multiple teams for complex tasks concurrently through crowd workers with specialized abilities is referred to as the multi-team formation problem. The primary aim of this study is to represent this formation problem using mathematical formulas and to analyze the outputs through a straightforward approach. To solve this problem, we propose a heuristic algorithm that forms teams with optimized compatibility for multiple tasks concurrently while satisfying all requirements.

The properties of the outputs of the proposed algorithm were analyzed through simulation experiments, which revealed how the team's compatibility was influenced by eight simulation parameters and hyperparameters. Simulation experiments were performed using a synthetic dataset. Some information, such as the weights of social network edges, worker skill levels, hiring costs, and task skill requirements, was generated from a Poisson distribution, with its parameters included in the experimental parameters. In the experiments, the objective function, which outputs the team's compatibility for an input vector comprising each experimental parameter, was approximated through Gaussian process regression (GPR) with the observed simulation outcomes. In addition, Bayesian optimization was employed to determine the experimental conditions necessary for regression.

The main contributions of this study are summarized as follows:

- We formulate a crowdsourcing team formation problem that simultaneously considers skill requirements, budget, team size, and worker compatibility.
- To solve the problem, we propose a heuristic algorithm that combines a SAT solver to generate feasible initial solutions with simulated annealing for optimization.
- Through extensive simulations, we analyze the characteristics of the proposed model with respect to various parameters, and demonstrate that our algorithm consistently outperforms hill climbing as a baseline.

2. Related work

There have been two types of applied research on team formation: focusing on applications within organizations and in crowdsourcing. In team formation within general organizations, all workers are typically required to be assigned to one of the teams, whereas in crowdsourcing team formation, a proportion of workers are selected from a larger worker pool. In addition, many studies on crowdsourcing team formation have considered factors such as hiring cost and skill levels of workers.

Crowdsourcing often involves tasks where individual workers operate independently, and many applications leverage the collective power of the crowd. Recent studies have addressed the team formation problem in social networks, which is similar to our mathematical problem. Furthermore, some studies have applied the team formation problem to crowdsourcing.

2.1. Crowdsourcing

Crowdsourcing offers cost-effective, high-quality results by securing labor from an unspecified number of people [1]. Numerous services and systems have emerged or evolved thanks to the development of the Internet. For example, reCHAPCHA is a crowdsourcing system for transcribing scanned book images to verify whether a user is a human being in specific situations, such as authentication. Foldit is a system that predicts protein structures by having people play an online puzzle game [8]. Amazon Mechanical Turk and Upwork are marketplaces that connect task outsourcers with willing workers.

2.2. Team formation problem on social networks

Many studies have tackled the team-formation problem by evaluating the compatibility of a team with a social network. Lappas et al. were the first to address the team-formation problem in social networks [9]. They modeled real-life relationships, such as those between bosses and co-workers, in social networks with edge weights as communication costs. In their study, the communication distance between workers was defined as the sum of the edge weights along the shortest paths in the social network. They also defined two objective functions for the team formation problem: (1) the diameter of the subgraph representing the largest distance between any two workers in the team; (2) the sum of the edge weights of the minimum Steiner tree formed on the subgraph by the team's nodes. Subsequent studies [10, 11] proposed a new measure called *density*. This measure is adopted in our study and is explained in the next section. Awal and Bharadwaj introduced a new measure of team compatibility, incorporating individual knowledge levels and the ability to work together as a group [12]. Juang et al. proposed a measure of team compatibility for the total distance from the project leader in a social network [13]. Selvarajah et al. introduced an index integrating explicit ratings among workers, worker profile proximity, and psychological compatibility [14].

Lappas et al. [9] included a requirement for team members to have certain skills in their problem formulation. Some studies used the number of workers with certain skills as a task requirement [10,15]. Other studies assigned skill levels to workers and imposed requirements on the total skill level within the team [16–18]. In addition, some studies considered worker remuneration for a project and incorporated a budget requirement into the team formation problem [19, 20]. Anagnostopoulos et al. proposed an algorithm that the number of tasks performed by an individual worker, with team compatibility as a constraint [21]. Conversely, Majumder et al. proposed an algorithm that optimizes team coordination while limiting the number of tasks a worker can undertake [6]. Rangapuram et al. proposed a team formation algorithm that limits team size to prevent team performance from degrading as the number of team members increases beyond a certain number [22].

Some studies have addressed the problem of limiting task duration [23] or considering geographic proximity [24]. In addition, some studies have proposed worker clustering algorithms: a study [25] reduces the computational complexity, while other studies [12,26] focus on evolutionary approaches.

2.3. Team formation on collaborative crowdsourcing

Table 1 presents a summary of previous studies on team formation problems in collaborative crowdsourcing. These studies have addressed the problem of forming teams of crowd workers within social networks.

Many studies on the team formation problem in crowdsourcing incorporate budget constraints due to the nature of crowdsourcing, yet few consider team compatibility. Rahman et al. addressed the problem of minimizing team coordination metrics, such as maximum distance and total distance within the team, assuming that the distances between all workers are quantified [27]. They partitioned teams to ensure that the number of members per team did not exceed a predefined limit. Sun et al. proposed an algorithm that first determines a centrality leader and then forms teams while limiting the total distance to that leader [28]. Yamamoto and Okamoto introduced a team formation system designed for projects with multiple tasks [29]. This system forms teams with optimized compatibility for each task while adhering to the same budget.

Yadav et al. proposed a method for simultaneously applying a team formation algorithm to multiple tasks [30]. In their approach, the objective function minimizes the total hiring costs of workers. Therefore, their problem formulation differs from our study, which focuses on optimizing team compatibility.

Some studies have investigated the following possibilities: crowd workers may incorrectly report information such as their skill levels and hiring costs [31, 32]; workers in a formed team may not always take on tasks [32]. Barnabo's method ensures fairness and diversity concerning various individual attributes, such as gender [33].

2.4. Quantifying compatibility

To quantify team compatibility, we must first quantify compatibility between individual workers. Rahman et al. quantified the compatibility of all worker combinations based on sociodemographic and psychological characteristics [27]. Several previous studies have represented the history of collaboration as a social network and have used a subgraph with team member nodes to quantify team compatibility. This study also defined team coordination in a similar manner.

Lappas et al. quantified the edge weights in a social network as the communication cost, which is calculated based on the percentage of collaborative projects between workers [9]. The higher the compatibility, the lower the communication cost. Validation using IMDb reveals a negative correlation between communication cost and team performance [34]. Moreover, in crowdsourcing systems, workers are matched arbitrarily, and the percentage of collaborations does not necessarily guarantee compatibility. Therefore, in this study, worker compatibility should be calculated based on explicit evaluations of one another.

3. Proposed model

This study mathematically defines the multi-team formation problem and proposes a model to solve this problem.

3.1. Preliminaries

3.1.1. Data model

We assume a set of n workers $U = \{u_1, u_2, \cdots, u_n\}$ and a set of m tasks $P = \{p_1, p_2, \cdots, p_m\}$. Each worker $u \in U$ has skill levels $[s_1, s_2, \cdots, s_l]$ and a hiring cost c. Similarly, each task $p \in P$ requires skill levels $[t_1, t_2, \cdots, t_l]$ and a budget b. All of these values are nonnegative integers. The l skills encompass all the required skills, and if a worker u lacks the k-th skill, then s_k for the worker is 0. For all tasks, m teams $\{G_1, G_2, \cdots, G_m\} \subset 2^U$ are generated by the proposed multi-team formation algorithm.

This study defines a social network representing the relationships among n workers. The vertex set of the social network is U, and the set of edges is E. In crowdsourcing systems, an edge is added between workers as they collaborate, with the edge weight (a nonnegative integer), determined by their mutual evaluations. Higher edge weights indicate greater compatibility. The weight of an edge is denoted by the function w. For instance, $w(u, v)$ represents the weight of the edge connecting workers u and v if edge $(u, v) \in E$; otherwise, if edge $(u, v) \notin E$, $w(u, v) = 0$.

Table 1. Studies of team formation problem on collaborative crowdsourcing

	Skill level	Budget	Team size	Multi-task	Team compatibility
Rahman et al. [27]	✓	✓	✓		✓
Sun et al. [28]		✓			✓
Yamamoto & Okamoto [29]	✓	✓		✓	✓
Yadav et al. [30]	✓	✓		✓	
Liu et al. [31]		✓			
Wang et al. [32]		✓			
Barnabo et al. [33]		✓			

3.1.2. Objective function

In previous studies [9, 19, 20, 35], the following four communication cost functions defined for team G have proposed as objective functions for the team formation problem:

- CC-Diameter(G): The communication cost of the team is defined as the diameter of its subgraph,

$$\text{CC-Diameter}(G) = \max_{\forall u,v \in G} d(u,v).$$

- CC-Steiner(G): The communication cost of the team is defined as the total edge weight of the minimum Steiner tree on its subgraph.

- CC-SD(G): The communication cost of the team is defined as the total distance between every pair of workers in the team,

$$\text{CC-SD}(G) = \sum_{\forall u,v \in G} d(u,v).$$

- CC-LD(G): The communication cost of the team is defined the sum of the distances between from each team member to the leader u_L,

$$\text{CC-LD}(G) = \sum_{\forall u_i \in G, u_i \neq u_L} d(u_i, u_L).$$

The distance function $d(u,v)$ for workers u and v in the social network is defined as the sum of the edge weights on the shortest paths between them.

The communication cost functions using the distance on the social network, such as CC-Diameter, CC-Steiner, and CC-LD, assume that shorter distances indicate higher compatibility among workers. This implies that even if workers have never collaborated, those who are close in the social network are assumed to be more compatible. However, we found no studies that support correlation between worker compatibility and social network distance in crowdsourcing contexts. In addition, these functions assume that all workers in the team are connected; this assumption may not hold in sparse graphs. Therefore, we adopt the density function $f(G)$, which can be computed directly from available edge weights and remains applicable even when the social network is sparse or partially disconnected.

We define a function f for any team G as

$$f(G) = \frac{\sum_{\forall u,v \in G} w(u,v)}{\|G\|},$$

which serves as a measure of team compatibility and is referred to as *density*. In this study, we employ this measure as the objective function of the multi-team formation problem. This objective function can be calculated only from the weights of the existing edges, even if the subgraph induced by of the team's nodes is disconnected.

3.2. Problem definition

This study addresses the problem of concurrently forming teams with optimized compatibility for multiple tasks. The measure of compatibility within a team is given by the *density*. Since multiple teams are formed, the objective function is defined as the sum of the *densities* of all teams. For each task, the total skill level of the team must exceed the required skill level, and the total employment cost must be within the budget. Each team must have no more than K members. Additionally, no worker may be assigned to more than one team. The mathematical formulation is given as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} \frac{\sum_{\forall i_1, i_2 \in [1,n]} w(u_{i_1}, u_{i_2}) x_{i_1 j} x_{i_2 j}}{\sum_{i=1}^{n} x_{ij}} \\
\text{subject to} \quad & \sum_{i=1}^{n} s_{ik} x_{ij} \geq t_{jk}, \quad \forall k \in [1,l],\ \forall j \in [1,m], \\
& \sum_{i=1}^{n} c_i x_{ij} \leq b_j, \quad \forall j \in [1,m], \\
& \sum_{i=1}^{n} x_{ij} \leq K, \quad \forall j \in [1,m], \\
& \sum_{j=1}^{m} x_{ij} \leq 1, \quad \forall i \in [1,n], \\
& x_{ij} \in \{0,1\}.
\end{aligned}$$

Note that x_{ij} denotes whether worker u_i is assigned to team G_j.

3.3. Proposed multi-team formation algorithm

The problem of finding a maximal *density* subgraph whose vertex set size is below a constant K is known to be NP-hard, and its instance set is a subset of the instance set of our defined problem. Therefore, the problem described above is also NP-hard.

In this study, we propose a heuristic algorithm for the multiteam formation problem. The basic strategy of the proposed algorithm is *Simulated Annealing* (SA). Due to the nature of SA, it is necessary to determine in advance whether a satisfactory solution exists; if it does, one such solution should be derived. For this purpose, a Boolean SATisfiability problem (SAT) solver was employed. Since SAT is NP-complete, the feasibility check has worst-case exponential complexity in the problem size. In practice, larger n, m, higher skill requirements t_{jk}, and smaller budget b_j or team size limit K tend to make the search more difficult. As noted in Section 4.2, we consider two failure cases as "no solution": (i) timeout beyond the cutoff, and (ii) encoding failure due to memory or file-size limits.

3.3.1. Simulated annealing

In the proposed algorithm with SA, the probability of transitioning to a neighboring solution with a lower evaluation value decreases over time. In addition, if the current solution is G and a neighboring solution G' (with $f(G') < f(G)$) is selected during the neighborhood search, the probability of transition to G' is given by

$$\exp\left(\frac{f(G') - f(G)}{T}\right).$$

The temperature T is updated over time according to the equation

$$T' = \alpha T. \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

It is noteworthy that α is a hyperparameter of the proposed algorithm. In our implementation, the initial temperature was set to 10. The stopping criterion was defined by the overall time limit L, where annealing procedure terminates once the allocated time per run is exeeded (details are provided in Section 3.3.3).

In the proposed algorithm, the neighborhood of a team G is defined as the set of candidate solutions that satisfy the constraints on skill levels, budget, and team size. These neighborhoods are determined from the following sets of teams:

- N_1: The set of teams formed by removing one worker from team G and adding one worker who does not belong to any team.
- N_2: The set of teams formed by removing two workers from team G and adding one worker who does not belong to any team.
- N_3: The set of teams formed by removing one worker from team G and adding two workers who do not belong to any team.

To select a new solution, the proposed algorithm first randomly selects one candidate from each of the three sets, and then randomly chooses one among these candidates. When the size of G is equal to K, only candidates from N_1 and N_2 are considered.

To obtain a set of candidate solutions from the neighborhoods, we must verify that every candidate team meets all the requirements. This verification process has a time complexity of $O(n^2)$, making it computationally expensive and impractical. However, since the neighborhood search ultimately requires only one valid solution, the proposed algorithm randomly selects candidate teams until it finds one that satisfies all the requirements. This procedure is applied to each of N_1, N_2, and N_3, resulting in three types of neighborhood solutions.

3.3.2. Smoothing for evaluation value

When evaluating solutions using the *density* measure, the evaluation function returns zero for any solution in which no workers in the team are adjacent on the social network. Consequently, smoothly improving the solution via a nearest neighbor search may be impossible. Therefore, during the algorithm's execution, a virtual edge is assigned between nonadjacent workers u and v with a weight. This is defined as

$$w(u,v) = \begin{cases} w(u,v) & (u,v) \in E \\ \beta \cdot \dfrac{1}{p} \cdot \dfrac{i}{5}, \quad \forall i \in \{0,1,\cdots,5\} & (u,v) \notin E, \end{cases} \quad (2)$$

where β is a hyperparameter, p is the number of edges in the shortest path between u and v in the social network, and i is a nonnegative integer. The proposed algorithm executes the SA procedure a total of six times, with each run assumed to take the same amount of time. The counter i is initially set to five and is decremented by one after each iteration of this process.

3.3.3. Main function

In addition to the variables defined in this section, the main function of the proposed algorithm accepts as inputs the maximum team size K, the hyperparameters in Eq. (1) and (2), the overall time limit for the algorithm L, and initial solutions produced by the SAT solver $\{G_1, G_2, \cdots, G_m\}$. The pseudocode for the main function is provided in Algorithm 1 and the explanation is as follows:

- Line 1–3: The initial solutions G obtained from the SAT solver are set as provisional best solutions. G' only stores the best-so-far solutions and does not affect the behavior of the algorithm.
- Line 4–22: The SA procedure is executed six times. The temperature is reset at the beginning of each run.
- Line 5: Assign the initial temperature.

- Line 6–21: For each run, repeat the neighborhood search while lowering the temperature until one-sixth of the total time limit has elapsed.
- Line 7–19: At each temperature level, repeat the neighborhood search for one six-hundredth of the total time limit.
- Line 20: After the above time limit has elapsed, lower the temperature.
- Line 8–13: For each team, conduct the neighborhood search.
- Line 9: For each team, obtain one neighboring solution. A neighboring solution is a team where one or two workers are replaced, possibly changing the team size.
- Line 10–12: If a random number drawn from (0, 1) is smaller than the right-hand side of the line 10, the current solution is replaced by the neighboring solution. The higher the temperature, the more likely the algorithm is to accept a worse solution.
- Line 14–18: Update the best solution. If the sum of $f(G_j)$ exceeds that of the best-so-far solution, replace it with the current solution.

Optimization is performed for each value of i in Eq. (2), and this process is executed six times in total; therefore, the time limit per run is set to $L/6$. The temperature in the SA procedure is updated 100 times per optimization, so time allocated for the neighborhood search at each temperature is set to $L/600$. Neighborhood searches are conducted separately for each team, and then each solution is updated. The probability of updating a solution extracted from the neighborhoods is given by

$$\exp\left(\frac{f(G_j'') - f(G_j)}{T}\right).$$

A value $rand(0, 1)$ is drawn from a uniform distribution in over the interval $[0, 1]$. Once all solution updates have been completed, the best solutions are selected and updated.

Algorithm 1 Main function

Input: wokers set U, edges set E, tasks set P, max team size K, hyperparameters α, β, time limit L, initial solutions $\{G_1, G_2, \cdots, G_m\}$
Output: optimized solutions $\{G_1', G_2', \cdots, G_m'\}$

```
1: for all j ← [1, m] do
2:     G'_j ← G_j
3: end for
4: for all i ← [5, 4, ··· , 0] do
5:     T ← 10
6:     repeat
7:         repeat
8:             for all j ← [1, m] do
9:                 G''_j ← a solution extracted from neighborhood
10:                if rand(0, 1) < exp((f(G''_j) − f(G_j))/T) then
11:                    G_j ← G''_j
12:                end if
13:            end for
14:            if Σ_{j=1}^m f(G_j) > Σ_{j=1}^m f(G'_j) then
15:                for all j ← [1, m] do
16:                    G'_j ← G_j
17:                end for
18:            end if
19:        until time exceeds L/600
20:        T ← α · T
21:    until time exceeds L/6
22: end for
```

4. Experimental evaluation

In this study, we conducted a simulation experiment, with three main objectives:

1. To determine the relationship between the experimental parameters and the evaluation values of the proposed algorithm's outputs. In particular, we aim to identify which parameters have a strong influence and how they depend on one another. This information can help establish the system conditions (e.g., the number of registered workers) under which the proposed algorithm is justified.
2. To investigate the relationship between the hyperparameters and the evaluation values in the proposed algorithm. In other words, we explore how the hyperparameters should be adjusted under various experimental conditions.
3. To validate whether the SA procedure and the smoothing technique positively contribute to the optimization performance of the proposed algorithm.

4.1. The synthetic dataset

We could not find real-world datasets on team formation using crowdsourcing. Therefore, in this study, we employ a stochastically generated synthetic dataset. Table 2 lists the experimental parameters used to generate the synthetic dataset. For each combination of experimental parameters, an optimization problem was generated with the following conditions:

- Social network generation: A social network is generated using the LFR benchmark [36], with n representing the number of workers and k denoting the degree mean of each node.
- Edge weight assignment: Each edge in the social network is assigned an integer weight between 1 and 5, randomly sampled from a Poisson distribution with a mean of 3.

Table 2. Experimental parameters

Parameter	Range	Description
n	$[100, 1000]$	The number of workers
m	$[1, 10]$	The number of tasks
K	$[1, 50]$	The limit of team size
k	$[5, 30]$	The degree mean
m_{SN}	$[2, 10]$	The mean of required skill numbers
m_{SL}	$[5, 45]$	The mean of required skill levels
b'	$[0, 500]$	The additional budget
L	$\{300, 450, 600\}$	The time limit of the algorithm [sec]

Table 3. GpyOpt parameter settings

Parameter name	Value
acquisition_type	LCB
acquisition_weight	5
kernel	Matern52 (ARD=True)
f	None
de_duplication	True

- Skill types and worker skills: This experiment assumes 20 skill types. The number of skills possessed by each worker is randomly sampled from a Poisson distribution with a mean of 5.
- Worker skill levels: Each worker's skill level, an integer between 1 and 9, is randomly sampled from a Poisson distribution with a mean of 3.
- Employment cost: A worker's employment cost is randomly sampled from a Poisson distribution with the mean equal to the sum of the worker's skill levels.
- Task skill requirements: For each of the m tasks, the number of required skills is randomly sampled from a Poisson distribution with a mean of m_{SN}, ensuring that the mean of the required skill counts across all tasks is as close as possible to m_{SN}.
- Required skill levels: The required levels of these skills are randomly sampled from a Poisson distribution with a mean of m_{SL}. Similarly, the mean of all skill levels is as close as possible to m_{SL}.
- Task budget: The budget for a task is defined as the sum of its required skill levels for each task plus a non-negative integer b'; that is, the budget for the task j is given by $\sum_{k=1}^{20} t_{jk} + b'$.

In this experiment, an optimization problem is generated from a combination of experimental parameters, and solving that problem yields a single evaluation value. In this way, a set of experimental parameter-evaluation value pairs can be obtained.

4.2. Experimental methodology

This experiment examines the relationship between the experimental parameters listed in Table 2 and the evaluation values of the teams formed using the proposed algorithm. Since there eight parameters, performing a grid searching would be time-consuming. Therefore, this study approximates the distribution of experimental parameter pairs and their corresponding evaluation values using GPR. Bayesian optimization is employed to select the next set of experimental parameters, with the resulting evaluations serving as input for further optimization. By iteration this process, a set of experimental parameter-evaluation value pairs can be efficiently obtained. In this experiment, the GPR and Bayesian optimization modules were implemented using *Gpy* and *GpyOpt*, respectively; the parameters used for *GpyOpt* are listed in Table 3. When the acquisition type is LCB, a larger acquisition weight biases the selection of experimental parameters toward exploration, while a smaller weight favors optimization. In order to improve the accuracy of the regression, the acquisition weight was set to a relatively high value (the default is two).

In the SAT solver, the constraint satisfaction problem is solved before the generated synthetic data are input into the proposed algorithm. In this study, we employed *sugar* [37] as a solver that encodes the constraint satisfaction problem to a Boolean CNF formula, which is then solved using the external SAT solver. For the external solver, *minisat*[1] was chosen for this experiment. The program to run *sugar* was implemented using the officially distributed *sugar-java-api*. Solving SAT problems with a large search space but a small solution space is extremely time-consuming. Therefore, if the computation takes more than one hour, we assume that no solution exists. In addition, if the variable space or the number of constraints becomes excessively large, these cases may exceed the upper limit of the file size to be encoded, making it impossible to run the solver. Such experimental parameters are treated as cases with no solution.

After the solver generates an initial solution, the proposed algorithm is executed. A grid search is conducted over the hyperparameters α and β; specifically, the algorithm is run for all hyperparameter pairs with $\alpha \in \{0.8, 0.85, 0.9, 0.95\}$ and $\beta \in \{0.5, 1, 1.5, 2\}$. The best evaluation results from these 16 combinations are then selected for Bayesian optimization. A grid search

1. Niklas Eén, Niklas Sörensson: "The MiniSat Page," http://minisat.se/, (2025-1-2)

over the hyperparameters is performed, yielding feasible solutions from 16 distinct runs for each set of experimental parameters. By performing regression on each of these outputs and comparing the results, we can determine which experimental parameters are key and how the hyperparameters should be adjusted. In this comparative experiment, the output results are compared with those obtained from the hill-climbing method to demonstrate the effectiveness of SA. Furthermore, to confirm the effectiveness of smoothing, the output results for the case $\beta = 0$ are included in the comparison. The proposed algorithm was implemented in C++, and all algorithms were executed on a computer with dual Intel Xeon Gold 6132 2.60GHz CPU and 192GB RAM.

5. Result and discussion

Bayesian optimization yielded results for 4,340 experimental conditions – i.e., unique combinations of experimental parameters – of which 1,921 produced feasible optimization problems. Although regression can be performed using both feasible and infeasible solutions, we conducted regression using only feasible solutions.

5.1. Analysis of experimental parameters

5.1.1. Fixed single parameter cases

We analyze the relationship between each experimental parameter and its corresponding evaluation value. For the parameter vector $\boldsymbol{p} = (n, m, K, k, m_{SN}, m_{SL}, b', L)$, one variable is held fixed while the remaining variables are randomly set within the ranges defined in Table 2. For these randomly set variables, 100,000 samples are generated (with replacement). For each combination of a fixed value and a generated sample, we compute $\boldsymbol{\theta} = g(\boldsymbol{p}|P, \boldsymbol{y})$, where $\boldsymbol{\theta} = (\mu, \sigma^2)$ denotes the mean and variance of the estimated evaluation value, g is a GPR function, $P \in \mathbb{N}^{1921\times 8}$ is the matrix of previously observed evaluation parameters corresponding to the 1,921 feasible optimization problems, and $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{1921}$ is the vector of evaluation value associated with P. This procedure is repeated for all 100,000 samples, and by averaging the results, the mean and variance corresponding to a given fixed value are computed. The fixed variable is then varied one step at a time over the range specified in Table 2. Fig. 1 illustrates the resulting mean and variance; the blue line represents the mean evaluation value, while the light-blue shaded area indicates the 95% confidence interval.

According to Fig. 1, we found the following insights:

- (a), (d), (g): The graphs for the number of workers n, the degree mean k, and the additional budget b' exhibit a monotonically increasing trend. The confidence intervals widen when n and b' are small, possibly due to the scarcity of feasible solutions for these parameters.
- (c): The graph for the limit of team size K increases monotonically up to 30 but tends to decrease beyond 40, likely because the disadvantage of a larger search space becomes more significant when K is extremely high.
- (b), (f): The graphs for the number of tasks m and the mean of the required skill levels m_{SL} generally decrease monotonically.
- (e): The graph for the mean of the required skill numbers m_{SN} does not exhibit a significant change in the evaluation values. In this experiment, 20 skill types were assumed, with workers possessing an average of three skills, suggesting that workers complement each other well in meeting the required skill levels.
- (h): The graph for the time limit of the algorithm L also shows minor variations, and in most cases, the annealing method converges earlier to a near-optimal solution.

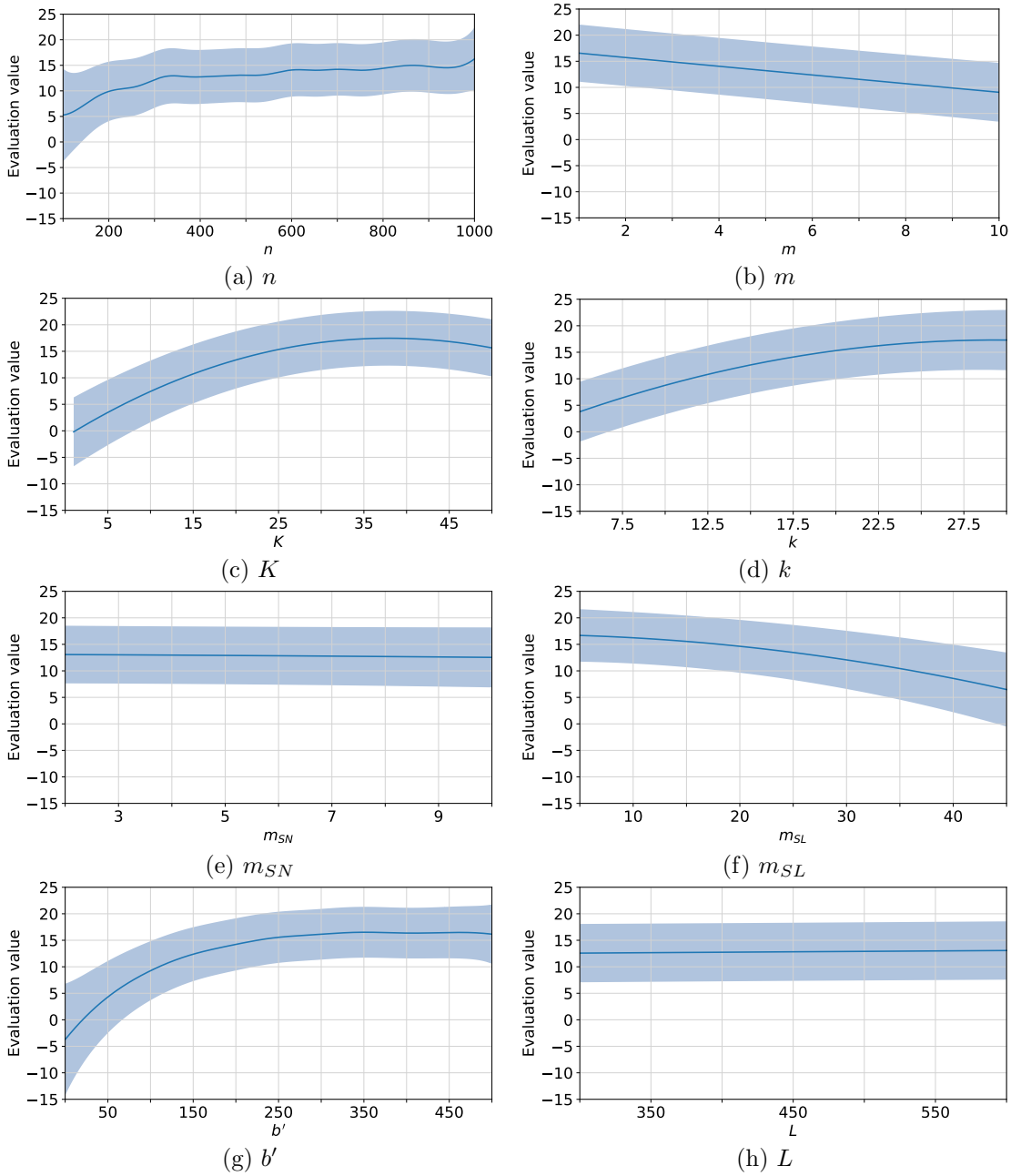
5.1.2. Fixed two parameters cases

In the previous section, we analyzed the effect of each experimental parameter on the estimated evaluation value by observing how the values change when one parameter is varied while all other parameters are marginalized. However, in practice, these parameters are not always independent; the evaluation values may become extremely low depending on the combination of the parameter values. For example, Fig. 2 shows that the evaluation values estimated by GPR for parameter n vary with different values of m, with particularly significant fluctuations observed when n is small. Note that, except for m and n, all estimated evaluation values in Fig. 2 were computed using the same method as in Fig. 1.

To analyze the independence between any two parameters, p_1 and p_2, we introduce the following metrics:

- The *variation* due to p_1 at a fixed p_2 is defined as the difference between the maximum and minimum estimated evaluation values obtained by varying p_1. For example, in Fig. 2, p_1 and p_2 correspond to m and n, respectively.
- The *maximum variation* due to p_1 is defined as the largest variation across all values of p_2.
- The *average variation* due to p_1 is defined as the difference between the maximum and minimum mean evaluation values, as indicated by the blue line in Fig. 1.

In this analysis, except for p_1 and p_2, the remaining parameters are marginalized using the same method as in Fig. 1. If p_1 and p_2 are independent, the variation in p_1 (measured at any fixed p_2) remains nearly constant,



(a) n (b) m (c) K (d) k (e) m_{SN} (f) m_{SL} (g) b' (h) L

Fig. 1. Estimated evaluation values for each experimental parameter

roughly matching the average variation in p_1. Therefore, a significant difference between the maximum variation in p_1 (as p_2 varies) and its average variation suggests a potential interaction between p_1 and p_2. Table 4 lists, for all pairs of parameters, the differences between the maximum and average variations. In particular, the entry in the row corresponding to p_2 and the column corresponding to p_1 represents the difference between the maximum variation in p_1 (as p_2 varies) and the average variations in p_1.

In Table 4, the parameter pairs with particularly high values are (K, k), (K, b'), and (k, b'). In other words, the parameters K, k, and b' are not independent. Heatmaps of the estimated evaluation values for these pairs are presented in Fig. 3 (a), (b), and (c). Fig. 3 shows that when one of the parameters K, k, or b' is small, the evaluation values remain low regardless of the other parameter values. This occurs because a small K or b' limits the number of teams that can be formed, and a small k results in a naturally sparse graph, which in turn leads to low evaluation values. Therefore, these results suggest that when the team size limit, degree mean, and additional budget are low, the proposed algorithm either fails to form teams that meet the requirements or produces teams with low compatibility.

Table 4 indicates that, among the parameter pairs (excluding K, k, and b'), the highest values are observed for (n, m) and (n, m_{SL}). The corresponding heatmaps are shown in Fig. 3 (d) and (e). According to these heatmaps, when n is small and either m or m_{SL} is large,

Table 4. Difference between maximum and average variations for each parameter pair (row: p_2, column: p_1)

	n	m	K	k	m_{SN}	m_{SL}	b'	L
n	0.0000	5.4968	3.9187	2.2635	1.7379	1.4265	3.2013	0.4146
m	4.0000	0.0000	4.2647	3.7021	0.3676	1.2027	3.8679	0.0033
K	3.9641	5.3501	0.0000	7.7454	1.1540	2.7514	8.6033	0.1750
k	1.3749	3.7234	6.4868	0.0000	1.4495	1.0462	6.5436	0.0284
m_{SN}	0.8613	0.3460	0.9020	1.4347	0.0000	0.8693	1.2565	0.0011
m_{SL}	2.2936	1.1853	2.1980	1.0195	0.8535	0.0000	2.2059	0.0056
b'	1.8463	3.8112	11.789	4.9661	1.4273	3.2042	0.0000	0.7432
L	0.0478	0.0024	0.1396	0.0012	0.0117	-0.0089	0.2501	0.0000

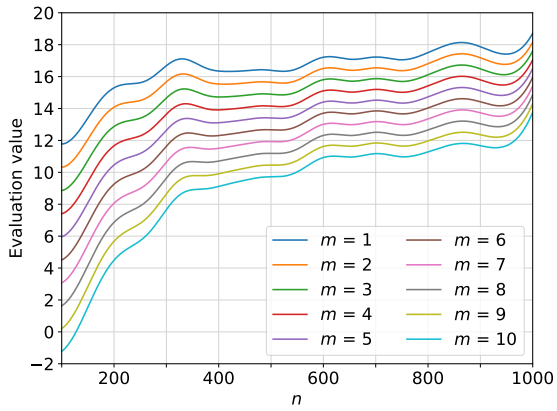


Fig. 2. Estimated evaluation values for parameter n vs. m

the evaluation value decreases substantially. This occurs because a large number of tasks or tasks with high skill-level requirements require more workers.

5.2. Analysis of hyperparameters

In this experiment, for each of the 1,921 feasible optimization problems, a grid search over the hyperparameters yielded 16 distinct outcomes, each comprising a solution and its corresponding evaluation value. Subsequently, either α or β is held fixed, and the four evaluation values corresponding to the other parameter are averaged. Finally, as in Fig. 1, one experimental parameter is varied while the remaining parameters are randomly sampled.

5.2.1. Fixed α cases

The results of the α comparison are shown in Fig. 4. The vertical and horizontal axes represent evaluation values and experimental parameters, respectively. In addition, the hill-climbing method was applied to every problems that produced a feasible solution, and the estimated evaluation values based on GPR are plotted in Fig. 4.

Fig. 4 shows that, overall, there is no significant difference in evaluation values among $\alpha = 0.8$, $\alpha = 0.85$, and $\alpha = 0.9$. The best performance was observed for $\alpha = 0.9$, while the worst was seen at observed for $\alpha = 0.95$. Alternatively, if the temperature is lowered too slowly or remains high for too long – resulting in a prolonged high probability of degradation – a near-optimal solution becomes difficult to achieve. This discrepancy is especially pronounced when both the number of workers and the number of tasks are large, possibly because a larger solution space requires more time to improve the solution; if the period during which alterations are likely is extended, there will be insufficient time to reach the optimal solution. Moreover, the shorter the algorithm's time limit, the greater the discrepancy between the results for $\alpha = 0.95$ and those for the other α values. Compared to the hill-climbing method, the proposed algorithm generally produced higher evaluation values. On average, the proposed algorithm achieved evaluation value improvements of at least 16.5%, 10.2%, 24.6%, 9.41%, 12.2%, 11.6%, and 6.84% for Fig. 4 (a)–(g), respectively.

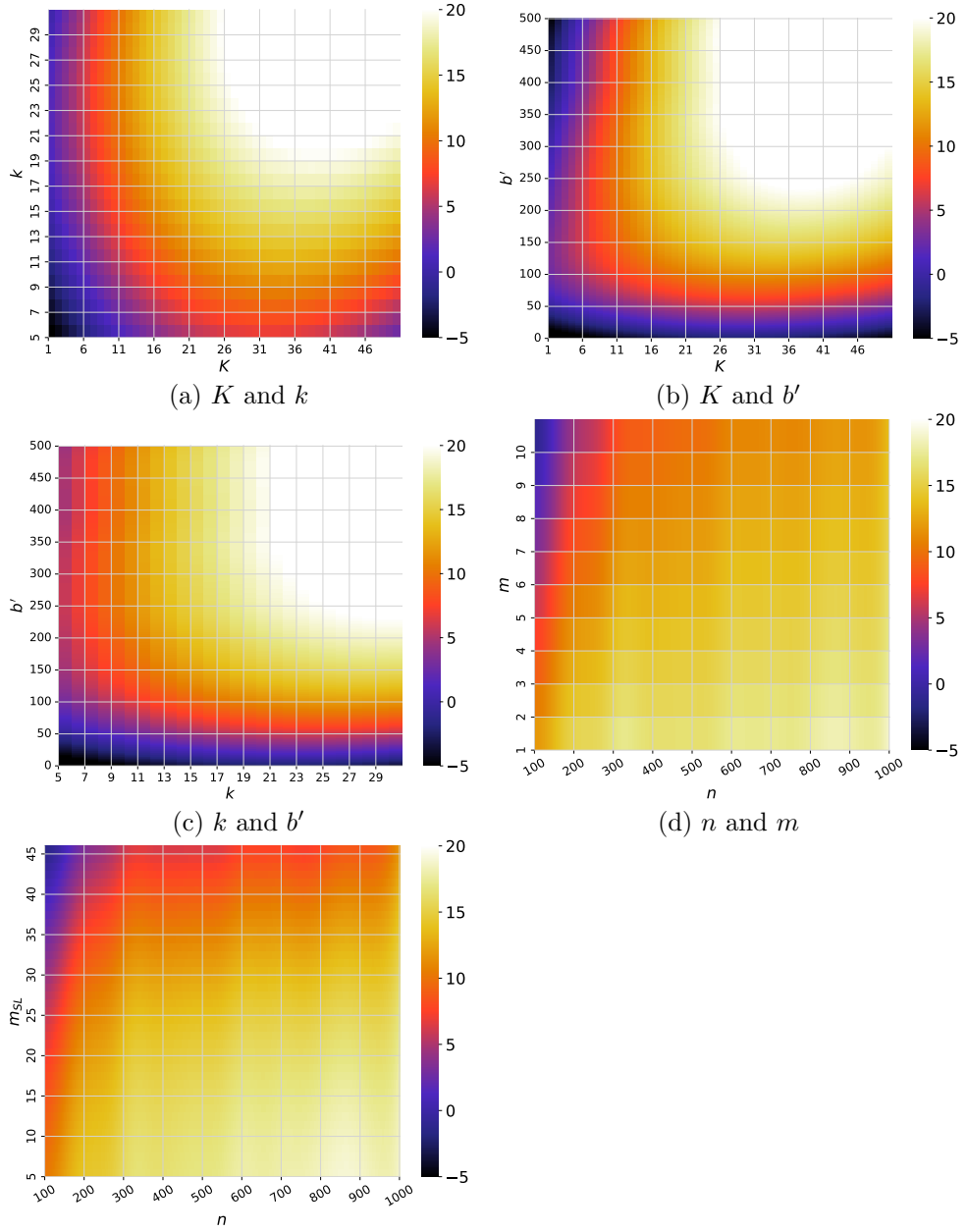
5.2.2. Fixed β cases

The same analysis was performed for β, and the results are shown in Fig. 5. For comparison, we computed the evaluation values for all α values with $\beta = 0$. Note that setting $\beta = 0$ is equivalent to applying no smoothing in the optimization. Fig. 5 indicates that the evaluation values are most favorable when $\beta = 0$, and the difference between these values and those obtained for small β is trivial. Therefore, the smoothing process appears to be ineffective.

6. Conclusion

In this study, we defined a crowdsourcing team formation problem, termed the multi-team formation problem, and proposed a heuristic team formation algorithm. In addition to the skill level and budget constraints considered in previous studies, our team formation problem introduced a team size requirement. The objective function is based on density – a plausible measure of team compatibility derived from social networks. Furthermore, the problem assumed concurrent team formation for multiple tasks while limiting the worker load.

The simulation experiment verified both the characteristics and effectiveness of the proposed algorithm. GPR was used to elucidate the relationship between the eight experimental parameters and the evaluation val-



(a) K and k

(b) K and b'

(c) k and b'

(d) n and m

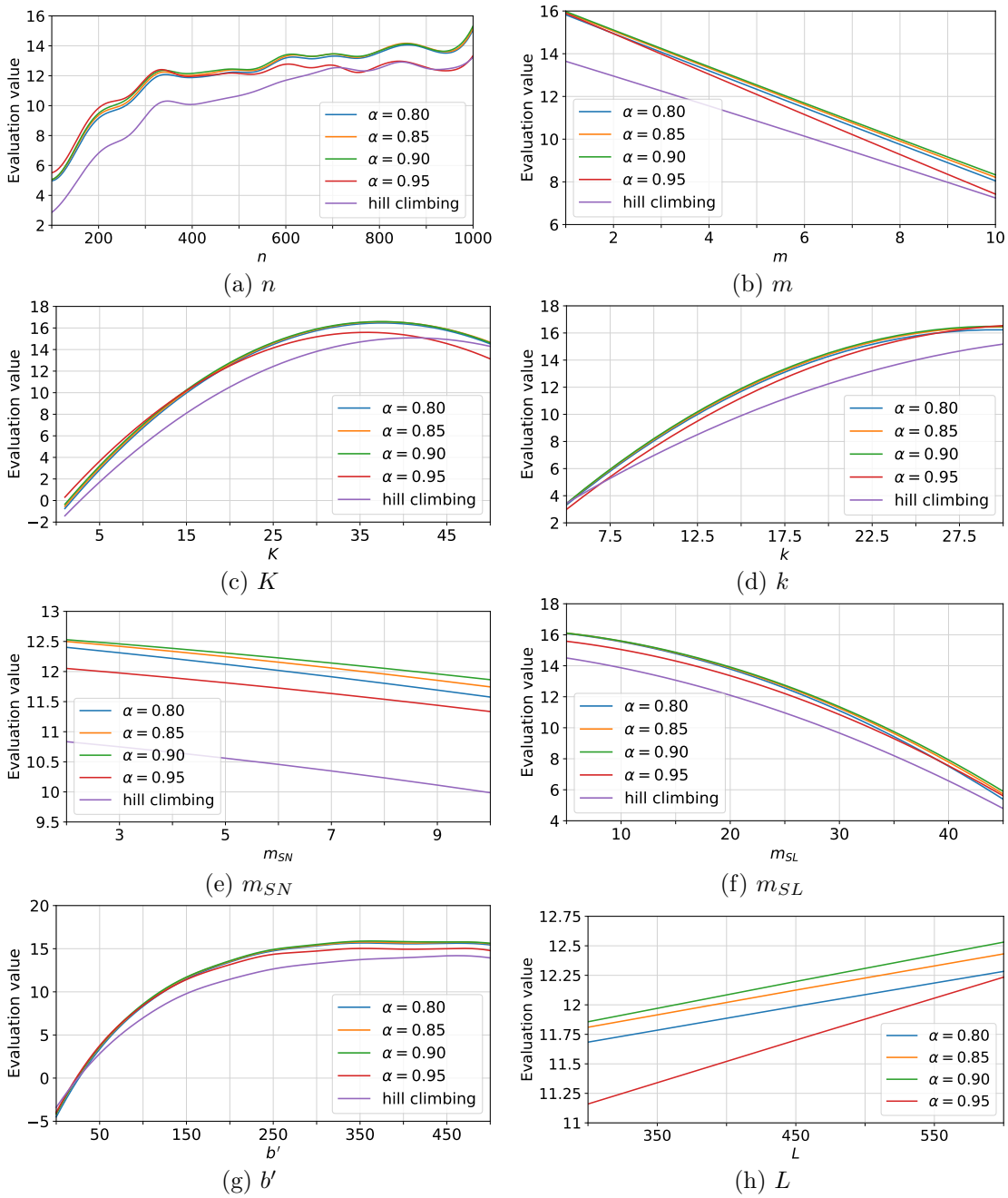
(e) n and m_{SL}

Fig. 3. Estimated evaluation values for each parameter pairs

ues. The results suggest that low values for the team size limit, degree mean, and additional budget adversely affect the evaluation values. We determined that a large number of workers is necessary when both the number of tasks and the distribution mean of required skill levels are high. The proposed algorithm has two hyperparameters: α, which controls the rate of temperature decrease in the simulated annealing method, and β, which controls the smoothing degree. According to this experiment, the highest evaluation values were observed when $\alpha = 0.9$, while variations in β had minimal impact. This suggests that smoothing process implemented in the proposed algorithm is ineffective. A comparison between the proposed algorithm and the hill-climbing method demonstrated the effectiveness of the annealing method.

One limitation of this study is that we compared the proposed algorithm only with a hill-climbing method as a baseline, a simple heuristic. While this sufficed to demonstrate the superiority of the proposed algorithm, more comprehensive comparisons with other metaheuristics such as genetic algorithms or reinforcement learning methods remain for future work.

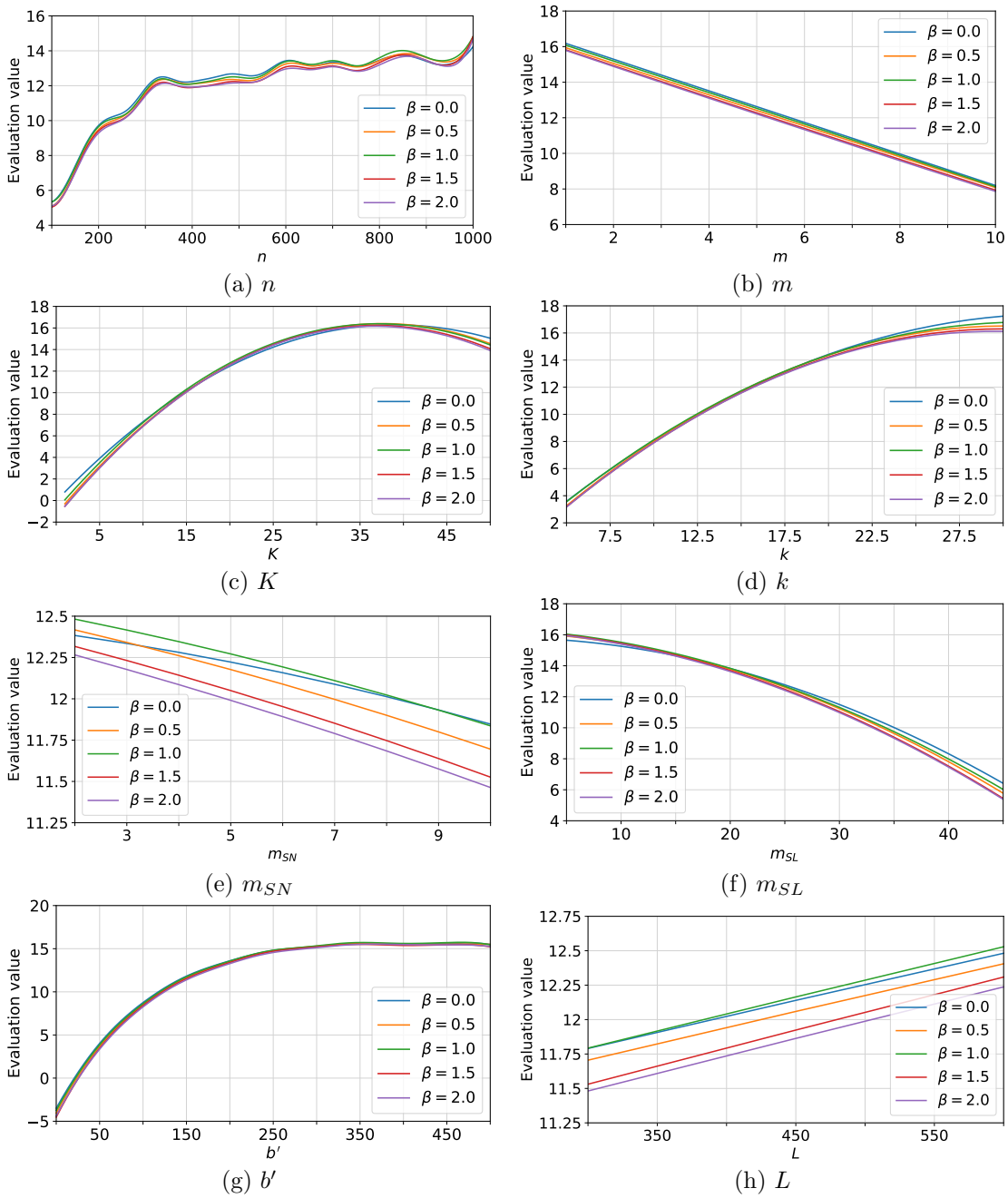
There are two primary areas for future research in this field. The first concerns the handling of new workers in the system. In the proposed algorithm, workers who are closely connected on a social network are more likely to be selected as team members. Consequently, well-connected workers tend to be chosen more often, whereas those with fewer neighboring nodes are



(a) n (b) m (c) K (d) k (e) m_{SN} (f) m_{SL} (g) b' (h) L

Fig. 4. Estimated evaluation values for each experimental parameter at different α

less likely to be selected. As teamwork is implemented, the edges of the social network are updated; however, a newly registered crowd worker typically has little experience with teamwork and few connections. This raises the challenge of fairly integrating new workers into the system. One possible approach is to adjust communication costs for edges connected to new workers, either by slightly increasing them to reflect uncertainty in their compatibility, or by lowering them to give such workers additional opportunities to be selected. Alternatively, predictive models could be employed to predict and augment their network connections based on worker profiles and the overall social network structure. Addressing how to balance fairness for new workers with the reliability of team performance remains an important direction for future work. The second area involves developing a program to generate initial solutions for the proposed algorithm. Before applying an optimization algorithm based on simulated annealing, it is essential to determine whether feasible solutions that satisfy all constraints exist. The current approach addresses this issue using an external SAT solver; however, this solver requires significant computation time when the search space is large and the region of feasible solutions is small. In addition, a large solution space combined with excessive constraints can lead to file encoding errors. Therefore, we are currently developing a new algorithm to solve this constraint satisfaction problem, as



(a) n (b) m (c) K (d) k (e) m_{SN} (f) m_{SL} (g) b' (h) L

Fig. 5. Estimated evaluation values for each experimental parameter at different β

described in [38].

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP21H03553, JP23K26329.

References:

[1] K. Kashima, S. Oyama, and B. Yukino, Human Computation and Crowdsourcing, 1st ed. Kodansha, Tokyo, 2016.

[2] G. Hertel, "Synergetic effects in working teams," Journal of Managerial Psychology, Vol.26, No.3, pp. 176-184, 2011. https://doi.org/10.1108/02683941111112622

[3] J. Hüffmeier and G. Hertel, "When the whole is more than the sum of its parts: Group motivation gains in the wild," Journal of Experimental Social Psychology, Vol.47, No.2, pp. 455-459, 2011. https://doi.org/10.1016/j.jesp.2010.12.004

[4] I. Lykourentzou, A. Antoniou, Y. Naudet, and S. P. Dow, "Personality matters: Balancing for personality types leads to better outcomes for crowd teams," Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, pp. 260-273, 2016. https://doi.org/10.1145/2818048.2819979

[5] K. L. Smart and C. Barnum, "Communication in cross-functional teams: an introducton to this special issue," IEEE Transactions on Professional Communication, Vol.43, No.1, pp. 19-21, 2000. https://doi.org/10.1109/TPC.2000.826413

[6] A. Majumder, S. Datta, and K. V. M. Naidu, "Capacitated team formation problem on social networks," Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1005-1013, 2012. https://doi.org/10.1145/2339530.2339690

[7] R. Kenna and B.Berche, "Managing research quality: critical mass and optimal academic research group

size," IMA Journal of Management Mathematics, Vol.23, No.2, pp. 195-207, 2011. https://doi.org/10.1093/imaman/dpr021

[8] F. Khatib, F. DiMaio, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, H. Zabranska, I. Pichova, J. Thompson, Z. Popović, M. Jaskolski, D. Baker, F. C. Group, and F. V. C. Group, "Crystal structure of a monomeric retroviral protease solved by protein folding game players," Nature Structural & Molecular Biology, Vol.18, No.10, pp. 1175-1177, 2011. https://doi.org/10.1038/nsmb.2119

[9] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 467-476, 2009. https://doi.org/10.1145/1557019.1557074

[10] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," Proceedings of the IEEE 2nd International Conference on Social Computing, pp. 9-16, 2010. https://doi.org/10.1109/SocialCom.2010.12

[11] A. Gajewar and A. D. Sarma, "Multi-skill collaborative teams based on densest subgraphs," Proceedings of the 2012 SIAM International Conference on Data Mining (SDM), pp. 165-176, 2012. https://doi.org/10.1137/1.9781611972825.15

[12] G. K. Awal and K. K. Bharadwaj, "Team formation in social networks based on collective intelligence –an evolutionary approach," Applied Intelligence, Vol.41, No.2, pp. 627-648, 2014. https://doi.org/10.1007/s10489-014-0528-y

[13] M.-C. Juang, C.-C. Huang, and J.-L. Huang, "Efficient algorithms for team formation with a leader in social networks," The Journal of Supercomputing, Vol.66, No.2, pp. 721-737, 2013. https://doi.org/10.1007/s11227-013-0907-x

[14] K. Selvarajah, P. M. Zadeh, Z. Kobti, Y. Palanichamy, and M. Kargar, "A unified framework for effective team formation in social networks," Expert Systems with Applications, Vol. 177, p. 114886, 2021. https://doi.org/10.1016/j.eswa.2021.114886

[15] C.-T. Li, M.-K. Shan, and S.-D. Lin, "On team formation with expertise query in collaborative social networks," Knowledge and Information Systems, Vol.42, No.2, pp. 441-463, 2015. https://doi.org/10.1007/s10115-013-0695-x

[16] C. Dorn and S. Dustdar, "Composing near-optimal expert teams: A trade-off between skills and connectivity," On the Move to Meaningful Internet Systems: OTM 2010, pp. 472-489, 2010. https://doi.org/10.1007/978-3-642-16934-2_35

[17] F. Farhadi, M. Sorkhi, S. Hashemi, and A. Hamzeh, "An effective expert team formation in social networks based on skill grading," Proceedings of the IEEE 11th International Conference on Data Mining Workshops, pp. 366-372, 2011. https://doi.org/10.1109/ICDMW.2011.28

[18] C. Dorn, F. Skopik, D. Schall, and S. Dustdar, "Interaction mining and skill-dependent recommendations for multi-objective team composition," Data & Knowledge Engineering, Vol.70, No.10, pp. 866-891, 2011. https://doi.org/10.1016/j.datak.2011.06.004

[19] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," Machine Learning and Knowledge Discovery in Databases, pp. 483-498, 2012.

[20] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," Proceedings of the 2013 SIAM International Conference on Data Mining, pp. 587-595, 2013. https://doi.org/10.1137/1.9781611972832.65

[21] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," Proceedings of the 21st International Conference on World Wide Web, pp. 839-848, 2012. https://doi.org/10.1145/2187836.2187950

[22] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," Proceedings of the 22nd International Conference on World Wide Web, pp. 1077-1088, 2013. https://doi.org/10.1145/2488388.2488482

[23] Y. Yang and H. Hu, "Team formation with time limit in social networks," Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, pp. 1590-1594, 2013. https://doi.org/10.1109/MEC.2013.6885315

[24] L. Chen, Y. Ye, A. Zheng, F. Xie, Z. Zheng, and M. R. Lyu, "Incorporating geographical location for team formation in social coding sites," World Wide Web, Vol.23, No.1, pp. 153-174, 2020. https://doi.org/10.1007/s11280-019-00712-x

[25] K. Selvarajah, A. Bhullar, Z. Kobti, and M. Kargar, "Wscan-tfp: Weighted scan clustering algorithm for team formation problem in social network," Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference, pp. 209-212, 2018.

[26] K. Selvarajah, P. M. Zadeh, M. Kargar, and Z. Kobti, "Identifying a team of experts in social networks using a cultural algorithm," Procedia Computer Science, Vol. 151, pp. 477-484, 2019. https://doi.org/10.1016/j.procs.2019.04.065

[27] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in collaborative crowdsourcing," Proceedings of the 2015 IEEE International Conference on Data Mining, pp. 949-954, 2015. https://doi.org/10.1109/ICDM.2015.119

[28] Y. Sun, W. Tan, and Q. Zhang, "An efficient algorithm for crowdsourcing workflow tasks to social networks," Proceedings of the IEEE 20th International Conference on Computer Supported Cooperative Work in Design, pp. 532-538, 2016. https://doi.org/10.1109/CSCWD.2016.7566046

[29] R. Yamamoto and K. Okamoto, "Worker organization system for collaborative crowdsourcing," Intelligent and Transformative Production in Pandemic Times, pp. 13-22, 2023. https://doi.org/10.1007/978-3-031-18641-7_2

[30] A. Yadav, A. S. Sairam, and A. Kumar, "Concurrent team formation for multiple tasks in crowdsourcing platform," Proceedings of the 2017 IEEE Global Communications Conference, pp. 1-7, 2017. https://doi.org/10.1109/GLOCOM.2017.8255048

[31] Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," Proceedings of the 2015 IEEE International Conference on Communications (ICC), pp. 567-572, 2015. https://doi.org/10.1109/ICC.2015.7248382

[32] W. Wang, J. Jiang, B. An, Y. Jiang, and B. Chen, "Toward efficient team formation for crowdsourcing in noncooperative social networks," IEEE Transactions on Cybernetics, Vol.47, No.12, pp. 4208-4222, 2017. https://doi.org/10.1109/TCYB.2016.2602498

[33] G. Barnabò, A. Fazzone, S. Leonardi, and C. Schwiegelshohn, "Algorithms for fair team formation in online labour marketplaces*," Companion Proceedings of The 2019 World Wide Web Conference, pp. 484-490, 2019. https://doi.org/10.1145/3308560.3317587

[34] W. Chen, J. Yang, and Y. Yu, "Analysis on communication cost and team performance in team formation problem," Collaborative Computing: Networking, Applications and Worksharing, pp. 435-443, 2018. https://doi.org/10.1007/978-3-030-00916-8_41

[35] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 985-994, 2011.

[36] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Physical Review E, Vol.78, No.4, p. 046110, 2008. https://doi.org/10.1103/PhysRevE.78.046110

[37] N. Tamura and M. Banbara, "Sugar: A csp to sat translator based on order encoding," Proceedings of the 2nd International CSP Solver Competition, pp. 65-69, 2008.

[38] R. Yamamoto and K. Okamoto, "An algorithm for solving constraint satisfaction problems in concurrent formation of expert teams," IEICE Transactions on Information and Systems (Japanese Edition), Vol. J107-D, No.3, pp. 87-97, 2024. https://doi.org/10.14923/transinfj.2023JDP7006

Name:
Ryota Yamamoto

Affiliation:
Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications

Address:
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

Brief Biographical History:
2021 Received B.E. degree from The University of Electro-Communications
2023 Received M.E. degree from The University of Electro-Communications

Main Works:
• R. Yamamoto and K. Okamoto, "Worker Organization System for Collaborative Crowdsourcing," Intelligent and Transformative Production in Pandemic Times (Part of Lecture Notes in Production Engineering), pp.13–22, 2023.
• R. Yamamoto and K. Okamoto, "An Algorithm for Solving Constraint Satisfaction Problems in Concurrent Formation of Expert Teams," IEICE Transactions on Information and Systems, Vol.J107-D, No.3, pp.87–97, 2024.

Name:
Kazushi Okamoto

ORCID:
0000-0002-9571-8909

Affiliation:
Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications

Address:
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

Brief Biographical History:
2006 Received B.E. degree from Kochi University of Technology
2008 Received M.E. degree from Kochi University of Technology
2011 Received Dr.Eng. degree from Tokyo Institute of Technology
2011-2015 Assistant Professor, Chiba University
2015-2020 Assistant Professor, The University of Electro-Communications
2020- Associate Professor, The University of Electro-Communications

Main Works:
• K. Sugahara and K. Okamoto, "Hierarchical Co-clustering with Augmented Matrices from External Domains," Pattern Recognition, Vol.142, pp.109657, 2023.
• K. Sugahara and K. Okamoto: "Hierarchical Matrix Factorization for Interpretable Collaborative Filtering," Pattern Recognition Letters, Vol.180, pp.99–106, 2024.

Membership in Academic Societies:
• Institute of Electrical and Electronics Engineers (IEEE)
• Association for Computing Machinery (ACM)
• Japan Society for Fuzzy Theory and Systems (SOFT)
• The Institute of Electronics, Information and Communication Engineers (IEICE)
• The Japanese Society for Artificial Intelligence (JSAI)