Recombination Operators for the Multi-objective Team Formation Problem in Social Networks

Julio Juárez*, Carlos A. Brizuela†, Hugo Terashima-Marín* and Carlos A. Coello-Coello‡
*School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, Mexico
Email: {juarez.julio | terashima}@tec.mx
†Department of Computer Science, CICESE Research Center, Ensenada, Mexico
Email: cbrizuel@cicese.mx
‡Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico
Email: ccoello@cs.cinvestav.mx

***Abstract*—The Team Formation Problem in Social Networks (TFP-SN) describes the process of finding an effective group of people, drawn from a network of experts, to perform a particular task. For a team to be considered as effective, it requires to comply with a task-specific skills set while also showing a high degree of cohesiveness. Although team effectiveness is subject to multiple criteria, the study of the problem from a multi-objective (MO) perspective is still scarce. In this paper, we focus on an MO TFP-SN whose objective is to maximize the team's level of expertise and the team's density, simultaneously. To solve this problem, we introduce two novel recombination operators to be used within the framework of the well-known NSGA-II. Our proposed crossover operators act as heuristics that compute the parents' unique and shared information, which is then combined for generating potentially improved offspring. Our experiments show that each of the two proposed crossover operators lead to significantly better results when compared to a naive crossover operator taken from the specialized literature. Particularly, the results consistently show higher hypervolume values when compared to the use of a simple recombination operator. The good performance of our proposed operators may be attributed to the incorporation of knowledge that exploits the structure of the problem.**

***Index Terms*—Team formation problem, Recombination operators, Multi-objective optimization, Combinatorial optimization**

I. Introduction

Teams are pervasive, establishing themselves as the building blocks of most organizations [18]. Featuring agile and flexible responses to complex dynamic tasks [15], [22], teams enable the potential to accomplish things beyond the reach of any individual acting alone [12]. From businesses, governments, communities, military, health, science, and everyday services, teams of people are the brains—and sometimes also the motor—behind many modern world activities [16].

Whenever a new team of people is required, a team formation problem arises. Team formation is the process of identifying suitable candidates to constitute a team, and the problem is to find the most effective team to undertake a particular task [14]. Team formation in the context of social networks deals with a problem variant where the candidates and their relationships are captured within a network of experts [17]. This network is typically modeled with a social graph. In such graphs, a node represents a candidate, and an edge represents some type of social relation. Also, each skill that a candidate possesses is preserved as an attribute of its corresponding node. Then, in its basic version, the Team Formation Problem in Social Networks (TFP-SN) consists of finding a set of candidates, with tightly knitted relationships, whose combined skills set is a superset of the task's demanded skills set [14], [17].

It has been over a decade since Lappas et al. [17] introduced the TFP-SN. Initially, research on this problem solely focused on single-objective formulations, mostly aimed at minimizing communication distance [17] or maximizing collaboration density [10]. However, as time progressed, other specific objectives (e.g., expertise level, personnel cost, geographical location distance) started gaining attention, leading to the exploration of multi-objective studies. In spite of this increase in multi-objective formulations, they still remain relatively scarce compared to the single-objective ones.

Multi-objective TFP-SN formulations have often been addressed using multi-objective evolutionary algorithms (MOEAs). However, most MOEA implementations for this problem have been simply regarded as black-box optimization tools, overlooking prior knowledge about the problem. For instance, commonly used recombination operators for the TFP-SN include adaptations of SBX [1], [4], n-point crossover [2], [9], [19] and uniform crossover [5], [11], [13], [20], which are relatively generic and somewhat naive. However, there is ample literature available about the problem (whether single-objective or multi-objective), allowing us to heuristically design more effective recombination operators and move beyond generic black-box approaches [14]. Other problems have benefited from what is called grey-box optimization, designed to actively exploit some structure or property about them [23].

In this paper, we direct our attention at a MO TFP-SN formulation that simultaneously maximizes the team's level of expertise and the team's density. In practice, these two objectives often conflict, as the most experienced individuals are not necessarily the most likely to work together and vice

C. Coello-Coello is also with the Faculty of Excellence of the School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey.

versa. To solve this problem, we present two novel grey-box recombination operators to be used within the framework of the well-known NSGA-II. We aim to study the effects of our proposed recombination operators in the performance of the NSGA-II. Our preliminary results show improved non-dominated fronts compared to those obtained from a naive black-box recombination operator, over a battery of test instances.

The remainder of this paper is organized as follows. Section II formally states the multi-objective problem to be solved, while Section III briefly reviews the previous related works. Section IV describes our proposed recombination operators. The experimental setup, including the data-set and the test instances adopted, along with the analysis of the results are described in Section V. Finally, in Section VI we present our conclusions and some possible paths for future research.

II. Problem definition

As a foreword to this section, Lappas et al.'s [17] TFP-SN notation has become common ground on this problem, we will utilize most of it, accordingly. Additionally, we will adopt the MO TFP-SN formulation from [13] as a model of study. Details of the problem definition will be described throughout this section.

A. Notation

Let $G = (\mathcal{X}, E)$ be an undirected weighted graph encoding a social network of experts. The set of nodes $\mathcal{X} = \{1, 2, \cdots, n\}$ represents the indices for each of the n candidates. The set of edges $E = \{(i, i')\}$, represents the pairwise collaborations of candidates $i, i' \in \mathcal{X}$. Also, let $w : E \to \mathbb{N}$ be a function that assigns a weight of collaboration to each edge. Then, a *team* $\mathcal{X}' \subseteq \mathcal{X}$ is represented by a subset of the candidates, and their relations are captured within the induced subgraph $G[\mathcal{X}']$.

Let $\mathcal{A}$ denote the universe of skills that any candidate could possess. Then, the particular skills set of candidate $i \in \mathcal{X}$ is denoted by $X_i \subseteq \mathcal{A}$. Additionally, let us define the support set of a skill $a \in \mathcal{A}$ as $S(a) = \{i \in \mathcal{X} | a \in X_i\}$, i.e., the subset of candidates that possess skill a. Finally, let us define a task $T = \{(a_1, k_{a_1}), (a_2, k_{a_2}), \cdots, (a_m, k_{a_m})\}$ as the set of m pairs (a_j, k_{a_j}) where $a_j \in \mathcal{A}$ and $k_{a_j} \in \mathbb{N}$, which indicates the required number of candidates (k_{a_j}) concerning any given skill (a_j). Thus, $\mathcal{X}' \subseteq \mathcal{X}$ is a valid team if $\forall (a_j, k_{a_j}) \in T$, $|\mathcal{X}' \cap S(a_j)| \geq k_{a_j}$.

B. Objective 1: Density

TFP-SN formulations are mostly concerned with either distance-based or density-based objectives [14]. In this paper, we focus on teams that show their effectiveness potential through a highly collaborative density.

Given $G = (\mathcal{X}, E)$ and a team $\mathcal{X}' \subseteq \mathcal{X}$, the team's collaborative density is modeled as the weighted graph density of $G[\mathcal{X}']$. Formally, the collaborative density is defined as:

$$D(\mathcal{X}') = \sum_{e \in E_{G[\mathcal{X}']}} \frac{2 \cdot w(e)}{|\mathcal{X}'| \cdot (|\mathcal{X}'| - 1)} \quad , \tag{1}$$

where $E_{G[\mathcal{X}']}$ represents the set of edges of the graph $G[\mathcal{X}']$. Note that this objective is defined as the weighted graph density, as opposed to the subgraph density studied in [10]. A previous study [13] revealed that graph density as an objective may perform better than subgraph density in terms of resulting team size and disconnected team components.

C. Objective 2: Expertise

Usually, for the TFP-SN, it is not only relevant to make sure that the demanded skill set is covered, but also to quantify the expertise level of a team with respect to the task.

Let $z_i(a)$ be an non-negative integer that represents the *level of expertise* of a candidate $i \in \mathcal{X}$ on a skill $a \in \mathcal{A}$. If $a \notin X_i$, then $z_i(a) = 0$. Thus, the total level of expertise of a team $\mathcal{X}'$ over a given task T is defined as:

$$Z(\mathcal{X}') = \sum_{i \in \mathcal{X}'} \sum_{(a_j, k_{a_j}) \in T} \frac{z_i(a_j)}{|\mathcal{X}'|} \quad . \tag{2}$$

Note that the total level of expertise exclusively quantifies the skills demanded by the task. Also note that this objective is a ratio of expertise to team size; otherwise, teams would grow excessively.

D. Multi-objective TFP-SN

In the team formation context, two conflicting objectives have been described: the density objective and the expertise objective. On the one hand, we may have a tightly knit team but very inexperienced, while on the other hand, we may have a very disjoint group of people yet very skillful. Both objectives deserve careful thought when constituting a new team [13].

The multi-objective team formation problem in social networks is formally defined as: Given a social graph G and a task T,

$$\begin{aligned} \underset{\mathcal{X}' \subseteq \mathcal{X}}{\text{maximize}} \quad & (D(\mathcal{X}'), Z(\mathcal{X}')), \\ \text{subject to} \quad & |\mathcal{X}' \cap S(a_j)| \geq k_{a_j}, \\ & \forall (a_j, k_{a_j}) \in T. \end{aligned} \tag{3}$$

Let us recall that in multi-objective optimization, we aim to optimize all the objectives simultaneously. Since there are often conflicting objectives, the goal is not to find a single solution but rather to search for a set of trade-off solutions. Specifically, we seek for the "best" set of trade-off solutions that cannot be improved in any individual objective without worsening any of the others. For additional information on multi-objective optimization, we refer the reader to [6].

III. Previous Related Work

In recent years, the number of studies on Team Formation Problems addressing multiple objectives has grown, yet they still remain somewhat scarce. Early multi-objective formulations, which were closer to Operations Research, were less focused on the social networks version of the problem. Feng et al. [9] introduced a formulation for team member selection based on three objectives: individual performance, collaborative performance within the team, and collaborative performance outside the team, each comprising multiple criteria. Zhang and Zhang [24] proposed a MO TFP considering team member capabilities and relationships based on personality composition. Awal and Bharadwaj [2] simultaneously optimized team expertise score and team trust score. Ahmed et al. [1] addressed cricket team selection with three objectives: batting performance, bowling performance, and fielding performance. Pérez-Toledano et al. [20] focused on basketball teams, maximizing the Performance Index Rating while minimizing overall team cost.

Transitioning to MO TFP-SN formulations, Niveditha et al. [19] proposed a model minimizing communication cost, team member cost, and team size. Chen et al. [5] introduced a TFP-SN minimizing communication costs and geographical location distance for a software development team. Juárez and Brizuela [13] incorporated team expertise level and collaborative density into the model. Selvarajah et al. [21] presented a TFP-SN formulation with four objectives: communication cost, expertise, geographical proximity, and collective trust. Gómez-Zará et al. [11] introduced a multi-team bi-objective TFP-SN for maximizing diversity and familiarity. Similarly, Casotti and Krohling [4] proposed a multi-team bi-objective TFP-SN for maximizing cohesion and disagreement (as a proxy for diversity).

All the aforementioned MO TFP(-SN) works solve their version of the problem using some form of evolutionary computation, with the majority (i.e., [1], [5], [9], [11], [13], [19]–[21]) resorting to implementations of the well-known NSGA-II algorithm [7]. Although the performance of NSGA-II and other (MO)EAs rely on their recombination operators [6], the literature tackling MO TFPs normally use generic crossover operators, such as SBX in [1], [4], n-point crossover in [2], [9], [19], or adaptations of the uniform crossover in [5], [11], [13], [20].

IV. Recombination operators

Within (MO)EAs, recombination operators play a crucial role in generating new feasible individual solutions by combining the information from two (or more) parent solutions [8]. However, in the context of the (MO) TFP-SN, the employed operators have been usually variants of generic recombinations. Furthermore, the problem constraints yield simple recombinations to often produce infeasible offspring. This encourages further investigation into more suitable operators, leveraging on the structure and characteristics of the problem. It is also desired that both the solution codification and the recombination operators work in a way that the individuals are regularly feasible.

In the subsequent subsections, we will introduce the individual representation along with its exploitable structure, followed by our proposed recombination operators explicitly designed to leverage on the parents' unique and shared information. Moreover, these recombinations not only ensure that the offspring are always feasible but also promote that the quality of the offspring is no worse than that of its parents.

A. *Individual representation*

The individual solution representation is simply the team definition in the problem formulation (Section II-A). That is, a feasible individual $\mathcal{I} \subseteq \mathcal{X}$ is a subset of the input nodes such that $\forall (a_j, k_{a_j}) \in T$, $|\mathcal{I} \cap S(a_j)| \geq k_{a_j}$. Additionally, let us refer to the support set of a particular candidate $\mathcal{I}$, for some skill a_j, as $S_{\mathcal{I}}(a_j) = \{i \in \mathcal{I} | a_j \in X_i\}$. Figure 1 shows an illustration of two feasible individuals, each representing a set of candidates possessing specific skill sets, which comply with the skill requirements of a given task.

B. *An exploitable structure*

For any two feasible individuals $\mathcal{I}_1$ and $\mathcal{I}_2$, the following partitions of candidates could be obtained: the intersection subset $H = \mathcal{I}_1 \cap \mathcal{I}_2$, and the symmetric difference subset $D = U_1 \cup U_2$, with $U_1 = \mathcal{I}_1 \setminus \mathcal{I}_2$ and $U_2 = \mathcal{I}_2 \setminus \mathcal{I}_1$.

Given partitions H and D, all candidates from one partition can be combined with a subset of candidates from the other, to form new (and potentially better) individuals. That is, a new team can be generated by using candidates from partition H as a base and selecting some additional candidates from D to complement it. Similarly, we may use partition D as a base of candidates and complement it with a subset of candidates from H. Ultimately, this structure defines a suitable search space for producing feasible offspring. Figure 2 shows an example of this concept featuring the individuals from the previous example (from Figure 1). In the following subsections, we will show how this structure could be used to efficiently produce feasible offspring. In algorithm 1, lines 2, 3 and 4,

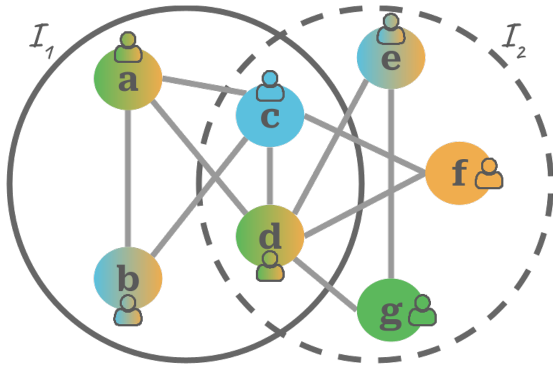


Fig. 1: Example of two feasible individuals $\mathcal{I}_1 = \{a, b, c, d\}$ (continuous-line circle) and $\mathcal{I}_2 = \{c, d, e, f, g\}$ (dashed-line circle), with skills s_1 (green), s_2 (orange) and s_3 (blue), considering task $T = \{(s_1, 2), (s_2, 3), (s_3, 2)\}$, where $X_a = X_d = \{s_1, s_2\}$, $X_b = X_e = \{s_2, s_3\}$, $X_c = \{s_3\}$, $X_f = \{s_2\}$, $X_g = \{s_1\}$.

C. *Recombination 1 ($R_\times$)*

This recombination operator functions similarly to an exhaustive local search, computing all feasible offspring combinations based on the information from two individuals. In Algorithm 1, lines 2 and 3, two core partitions U_1 and U_2 are computed. Then, for each pair (u_1, u_2) in $U_1 \times U_2$, two

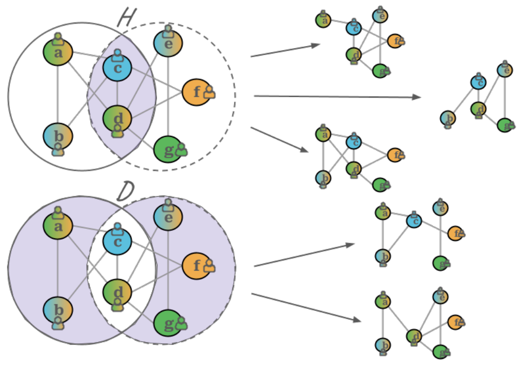


Fig. 2: Example of partitions H and D, and their possible offspring, given the parents from Figure 1.

new offspring are generated by swapping candidates u_1 and u_2 within individuals $\mathcal{I}_1$ and $\mathcal{I}_2$ (lines 5 and 6). Subsequently, any feasible offspring produced this way is stored in a set Q (lines 7–10). Finally, the non-dominated individuals in Q are returned (lines 12 and 13). Due to the exhaustive exploration involved, Algorithm 1 ensures that it will find a feasible offspring better than its parents, if any exist.

As an example, given parents $\mathcal{I}_1$ and $\mathcal{I}_2$, and a task T from Figure 1, the following offspring would been generated $O_1 = \{e, b, c, d\}$, $O_2 = \{c, d, a, f, g\}$, $O_3 = \{f, b, c, d\}$, $O_4 = \{c, d, e, a, g\}$, $O_5 = \{g, b, c, d\}$, $O_6 = \{c, d, e, f, a\}$, $O_7 = \{a, e, c, d\}$, $O_8 = \{c, d, b, f, g\}$, $O_9 = \{a, f, c, d\}$, $O_{10} = \{c, d, e, b, g\}$, $O_{11} = \{a, g, c, d\}$, $O_{12} = \{c, d, e, f, b\}$. However, not all of them represent a feasible solution, only O_4, O_6, O_7, O_8, and O_{10} are feasible concerning T. Finally, only the non-dominated ones are kept.

The worst-case scenario is when the intersection of the parents $\mathcal{I}_1 \cap \mathcal{I}_2$ is empty, which means that $U_1 = \mathcal{I}_1$ (from line 2) and $U_2 = \mathcal{I}_2$ (from line 3), which also means a total of $2 \cdot |\mathcal{I}_1 \times \mathcal{I}_2|$ possible offspring. Note that, for a given task T, no individual $\mathcal{I}$ in the pool is larger than $|\mathcal{I}| \leq K = \sum_{(a_j,k_{a_j})\in T} k_{a_j}$ but no smaller than $|\mathcal{I}| \geq k = \max_{(a_j,k_{a_j})\in T}\left(k_{a_j}\right)$. The number of basic operations is $O(K^3)$ in the **for** loop (lines 5 – 10), plus $O(2K^2)$ in the `Fast-non-dominated-sort(·)` procedure (line 12). Thus, the overall complexity of Algorithm 1 is $O(K^3)$, with $\Omega(k^3)$ when $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$. However, the worst case scenario is only expected during early generations, where individuals are generally less likely to share candidates.

D. Recombination 2 (R_+)

The recombination operator described here, unlike the previous one, utilizes a stochastic process to generate offspring. Algorithm 2 generates two offspring, each centered around partition sets H and D (lines 2–4). Then for each skill deficit in each offspring, the missing skills are covered by randomly adding candidates from the complementary partition's support set (lines 7–10). That is, offspring whose core team is given by H is complemented with suitable candidates from D, and vice versa. Finally, the algorithm discards any offspring that

Algorithm 1: Recombination $R_\times$

```
input : two individuals, $\mathcal{I}_1$ and $\mathcal{I}_2$, and a task $T$
output: a set of individuals $Q$
1  $Q \leftarrow \emptyset$;
2  $U_1 \leftarrow \mathcal{I}_1 \setminus \mathcal{I}_2$;
3  $U_2 \leftarrow \mathcal{I}_2 \setminus \mathcal{I}_1$;
4  for $(u_1, u_2) \in U_1 \times U_2$ do
5      $O \leftarrow (\mathcal{I}_1 \setminus \{u_1\}) \cup \{u_2\}$;
6      $O' \leftarrow (\mathcal{I}_2 \setminus \{u_2\}) \cup \{u_1\}$;
7      if isValid($O, T$) then
8          $Q \leftarrow Q \cup O$;
9      if isValid($O', T$) then
10         $Q \leftarrow Q \cup O'$;
11 end
12 $F \leftarrow$ Fast-non-dominated-sort($Q$);
13 return $Q \leftarrow F_1$;
```

fails to dominate at least one of the two parents (lines 12–15). Algorithm 2 ensures that the generated offspring is feasible (by default), but also that it is better than at least one of the parents.

As an example, given parents $\mathcal{I}_1$ and $\mathcal{I}_2$, and a task T from Figure 1, the offspring would have been initialized as $O_1 = \{c, d\}$ and $O_2 = \{a, b, e, f, g\}$. While O_2 is already valid, i.e., has no deficit concerning T. On the other hand, O_1 requires additional candidates to meet the demanded skill set. The missing candidates are randomly drawn from D according to the support set of the skills deficit.

Similar to Recombination 1, the worst case scenario for Recombination 2 occurs when H is empty (line 2), which implies that offspring O_1 is initialized as an empty set (line 4). It would also imply that the deficit of O_1 concerning task T is total. There is a mirroring scenario when D is empty (although this would imply that $\mathcal{I}_1 = \mathcal{I}_2$). In that case, O_2 is initialized as an empty set (line 5). The `Sampling`(S, k) routine computes a randomly sampled subset of size k from an input set S, which would take $O(|S|)$ u.t., with $|S| \leq |\mathcal{I}_1| + |\mathcal{I}_2|$, but since $|\mathcal{I}| \leq K$, then the complexity corresponding to the lines 8–11 can be simply rewritten as $O\big(K\big)$ units of time. Thus, the overall time complexity for this algorithm is $O\big(K \cdot |T|\big)$.

V. Experiments

In this section, we evaluate the performance of our proposed recombination operators for the Multi-objective Team Formation Problem in Social Networks. We describe the experimental setup, including the dataset, test instances, and running parameters. For comparison purposes, we also implemented a recombination operator introduced in [13] for this problem and a random choice hyper-heuristic recombination. We refer to each of them as $R_\circ$ and R_*, respectively. For simplicity, throughout this section, we will also refer to our proposed recombination operators as $R_\times$ for Algorithm 1 and R_+ for Algorithm 2. All the operators are implemented within the

Algorithm 2: Recombination R_+

```
input  : two individuals, $\mathcal{I}_1$ and $\mathcal{I}_2$, and a task $T$
output: a set of individuals $Q$
1  $Q \leftarrow \emptyset$;
2  $H \leftarrow \mathcal{I}_1 \cap \mathcal{I}_2$;
3  $D \leftarrow (\mathcal{I}_1 \setminus \mathcal{I}_2) \cup (\mathcal{I}_2 \setminus \mathcal{I}_1)$;
4  $O_1 \leftarrow H$;
5  $O_2 \leftarrow D$;
6  for $(a_j, k_{a_j}) \in T$ do
7      if $|S_{O_1}(a_j)| < k_{a_j}$ then
8          $O_1 \leftarrow$
           $O_1 \cup \texttt{Sampling}(S_D(a_j), k_{a_j} - |S_{O_1}(a_j)|)$;
9      if $|S_{O_2}(a_j)| < k_{a_j}$ then
10         $O_2 \leftarrow$
           $O_2 \cup \texttt{Sampling}(S_H(a_j), k_{a_j} - |S_{O_2}(a_j)|)$;
11 end
12 if $O_1$ dominates $\mathcal{I}_1 \vee O_1$ dominates $\mathcal{I}_2$ then
13     $Q \leftarrow Q \cup O_1$;
14 if $O_2$ dominates $\mathcal{I}_1 \vee O_2$ dominates $\mathcal{I}_2$ then
15     $Q \leftarrow Q \cup O_2$;
16 return $Q$;
```

NSGA-II because it is one of the most widely used MOEAs for this problem [14] (for particular details about this method refer to [7]). Our experimental results, are presented next. We also show some of the obtained non-dominated fronts.

A. Other recombination operators

For comparison purposes, we present Algorithm 3 originally introduced in [13] for this problem. Algorithm 3, or $R_\circ$ for short, is a recombination operator inspired by uniform crossover [8], particularly tailored for this problem. $R_\circ$ performs uniform crossover, not for every candidate, but for each skills support set $(S(a_j))$, as dividing sections. Since each parent's support set meets the required number of skilled candidates by the task, then the resulting offspring is always valid.

Algorithm 3: Recombination $R_\circ$

```
input  : two individuals, $\mathcal{I}_1$ and $\mathcal{I}_2$, and a task $T$
output: two individuals, $O_1$ and $O_2$
1  $O_1 \leftarrow \emptyset$;
2  $O_2 \leftarrow \emptyset$;
3  for $(a_j, k_{a_j}) \in T$ do
4      if CoinFlip() = Heads then
5          $O_1 \leftarrow O_1 \cup S_{\mathcal{I}_1}(a_j)$;
6          $O_2 \leftarrow O_2 \cup S_{\mathcal{I}_2}(a_j)$;
7      else
8          $O_1 \leftarrow O_1 \cup S_{\mathcal{I}_2}(a_j)$;
9          $O_2 \leftarrow O_2 \cup S_{\mathcal{I}_1}(a_j)$;
10     end
11 end
12 return $O_1, O_2$;
```

As a supplementary comparison, we adopted a no-learning random selection hyper-heuristic, denoted as R_*. The procedure for this recombination is as follows: Every ten generations, a new recombination is randomly chosen with uniform probability from $R_\times$, R_+, or $R_\circ$. While this method utilizes the proposed recombinations, its primary aim is to show an intermediate trade-off between computing time and the quality of Pareto front approximations. Also, it is worth mentioning that recent success has been observed in hyper-heuristics selecting recombinations for addressing other community detection-related problems [3].

B. The DBLP dataset

The DBLP has become one of the primary sources of data for benchmarking and testing in the TFP-SN literature. It is an online bibliographic information system on computer science publications and authors. Its data—openly available through a webpage[1] or an XML file[2]—is used to produce the input social graph, where each author is represented by a node, and mutually co-authored publications are represented by an edge. The general methodology to generate the social graph, including candidates, skills-set, an collaborations, is described in [10], [13], [17]. The papers published in 16 selected conferences are categorized in the domains of Artificial Intelligence (AI), Databases (DB), Data Mining (DM), and Theory (T) as follows: AI = {ICML, ECML, COLT, UAI}, DB = {SIGMOD, VLDB, ICDE, ICDT}, DM = {WWW, KDD, SDM, PKDD}, and T = {SODA, FOCS, STOC, STACS}. The set of candidates $\mathcal{X}$ contains authors skilled in at least one of these four domains. The skillset of any author i is determined as follows: if i has three or more publications in, say AI, then $i \in S$(AI) and AI $\in X_i$. The same goes for every other domain. An edge $e = (i, j)$ in E, $i, j \in \mathcal{X}$, exists if candidates i and j are coauthors in at least two publications. Also, the total number of collaborations between i and j determines the weight $w(e)$ of the edge $e = (i, j)$. Then, the level of expertise $z_i(a_j)$ of a candidate $i \in \mathcal{X}$ over a particular skill $a_j \in \mathcal{A}$ is given by the total number of publications within that skill domain. We used the DBLP XML data from a snapshot taken on March 02, 2015, to produce the input social network graph.

C. Test instances

We have adopted the 15 test instances from [13], which consist of five tasks for each size of $K = \{4, 8, 12\}$. Due to the small sizes of the first set of instances, we have randomly generated an additional set of tests, and five tasks for each size of $K = \{14, 16, 18\}$. We use 30 tasks in total, as a benchmark for our experiments. Let us recall that $K = \sum_j k_{a_j}$, where k_{a_j} specifies the minimum required number of experts in a team to cover the skill a_j, from a task T. Each new task T is generated by randomly choosing one skill from $\{AI, DM, DB, T\}$, repeated K times. All the test instances, former (left side) and new (right side), are presented in Table I.

[1]http://dblp.uni-trier.de/
[2]http://dblp.uni-trier.de/xml/

TABLE I: Randomly generated benchmark set of test instances.

Tasks	K	AI	DM	DB	T	Tasks	K	AI	DM	DB	T
Test 1	4	1	1	0	2	Test 16	14	0	7	4	3
Test 2	4	1	1	2	0	Test 17	14	4	3	3	4
Test 3	4	2	0	0	2	Test 18	14	2	1	5	6
Test 4	4	0	3	1	0	Test 19	14	3	3	6	2
Test 5	4	2	1	1	0	Test 20	14	4	4	6	0
Test 6	8	3	2	3	0	Test 21	16	2	4	8	2
Test 7	8	1	2	4	1	Test 22	16	4	4	2	6
Test 8	8	2	2	2	2	Test 23	16	5	3	2	6
Test 9	8	2	3	2	1	Test 24	16	4	4	3	5
Test 10	8	2	1	5	0	Test 25	16	5	2	2	7
Test 11	12	4	2	3	3	Test 26	18	3	4	5	6
Test 12	12	3	3	3	3	Test 27	18	4	9	4	1
Test 13	12	2	3	5	2	Test 28	18	4	10	4	0
Test 14	12	4	0	6	2	Test 29	18	7	7	0	4
Test 15	12	1	5	4	2	Test 30	18	4	3	2	9

D. Experimental setup and results

All the algorithms[3], including the NSGA-II, were implemented in Java (1.8) and the corresponding tests were performed on an ASUS TUF FX505DY, AMD Ryzen 5 CPU (64 bits), Windows 10 machine with 8 GB of memory.

The running parameters for NSGA-II were set as follows: population size = 50, tournament size = 2, generations = 500, recombination probability = 0.95, and mutation probability = 0.05 (utilizing the same mutation operator as in [13]). We carried out 30 independent runs for each task (from Table I), for each recombination. For a fair comparison, each recombination was tested using a consistent initialized population. For performance assessment of the recombination operators, we selected the hypervolume (HV) indicator [25]. Furthermore we visualized some of the empirical attainment functions from the resulting populations of the runs [25].

Table II displays the average HV indicator values (with reference point at the origin) and their standard deviations obtained from the experiment. The cells in dark- and light-gray shade represent the best and second-best results, respectively, for each test. The superscripts $(\times, +, \circ, *)$ next to each HV value indicate statistical significance (p-value < 0.05), based on a one-tailed Wilcoxon rank-sum test, over the recombination indicated by the superscript. The overall results in Table II show that recombination operators $R_\times$, R_+ and R_* outperformed $R_\circ$ in terms of HV. Most remarkably, $R_\circ$ failed to secure a first or second place in any test. In contrast, $R_\times$ achieved the best results in 11 tests and the second-best results in 10 tests (21 top results in total), R_+ scored 15 first places and seven second places (22 in total), while R_* attained only four best results and 13 runner-up positions (17 in total).

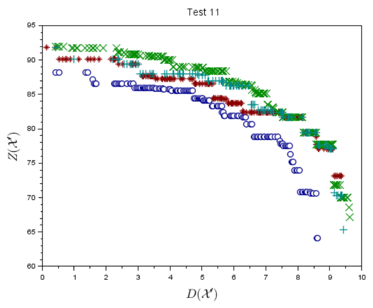
At first glance, it appears that operators $R_\times$ and R_+ exhibit slightly better overall performance than R_*, although statistical significance supporting this observation is only achieved on a few tests. Furthermore, $R_\times$ and R_+ seem to produce comparable results. However, when focusing exclusively on the larger tests (tests 15–30), it becomes evident that R_+ outperformed $R_\times$ in 10 out of the 15 tests, showing a trend of better performance for R_+ on larger tests. Nevertheless, drawing definitive conclusions from the available experiments and executions is challenging, as statistical significance was not consistently observed in the results.

[3]All the codes and algorithms will be available upon request.

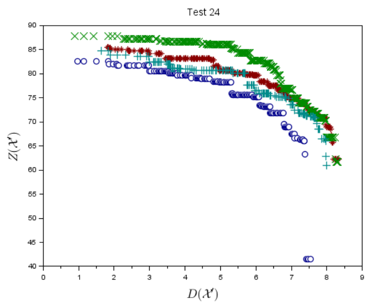
TABLE II: Comparative analysis of average hypervolume indicator scores (and standard deviations) from each recombination operator on the test instances.

Test No.	$R_\times$	R_+	$R_\circ$	R_*
1	2185.26° (481.03)	2272.02° (403.61)	1895.31 (588.09)	2171.27° (488.74)
2	14157.54 (4275.00)	15619.57 (1015.71)	11886.43 (5547.91)	15080.62$^{\circ\times+}$ (3109.39)
3	1613.03$^{\circ*}$ (187.51)	1540.84$^{\circ*}$ (212.61)	1203.65 (402.16)	1387.64° (302.43)
4	4890.60° (3005.05)	4256.17° (2561.77)	2964.05 (2631.52)	4925.34° (2774.40)
5	4729.06° (1287.05)	4633.30 (1333.69)	3774.91 (1885.26)	4342.05 (1666.17)
6	2725.15 (426.35)	2929.93$^{\circ\times}$ (264.49)	2594.93 (469.86)	2811.01° (372.62)
7	4188.48° (363.02)	4162.71° (433.82)	3359.19 (1019.16)	4089.35° (303.53)
8	1745.88° (300.66)	1657.22 (289.74)	1556.01 (340.47)	1782.96° (330.43)
9	2194.31° (788.79)	2603.59$^{\circ*}$ (942.51)	1741.07 (841.74)	2028.06 (841.20)
10	3900.27° (371.91)	3927.06° (377.52)	3135.67 (815.24)	3950.83° (340.63)
11	847.27° (95.63)	839.21° (108.79)	712.30 (128.44)	840.19° (135.81)
12	1122.64° (165.79)	1088.55° (148.82)	904.93 (180.50)	1118.44° (187.06)
13	2083.38° (202.45)	2080.72° (231.71)	1738.46 (399.54)	2101.93° (202.80)
14	1174.35° (121.31)	1198.26$^{\circ*}$ (101.37)	943.77 (121.61)	1121.51° (138.60)
15	2540.83° (476.57)	2504.88° (534.66)	2031.26 (678.25)	2400.48° (514.51)
16	1370.27° (334.61)	1556.81$^{\circ\times}$ (312.73)	1158.59 (330.76)	1513.07$^{\circ\times}$ (325.34)
17	784.89$^{\circ+}$ (84.64)	759.06° (101.97)	650.90 (105.00)	777.02° (108.70)
18	916.83° (85.13)	942.56° (88.12)	792.19 (114.41)	925.71° (118.51)
19	1513.24° (150.32)	1535.97° (76.39)	1274.31 (260.06)	1479.97° (158.09)
20	1802.46° (162.86)	1867.13° (130.31)	1531.62 (271.08)	1846.46° (137.23)
21	1686.87° (152.55)	1702.23° (164.48)	1306.45 (263.20)	1649.33° (144.25)
22	607.90$^{\circ+*}$ (62.91)	574.01° (57.03)	500.31 (79.94)	575.37° (59.54)
23	542.15$^{\circ+*}$ (38.47)	506.11° (33.49)	403.00 (63.60)	505.84° (54.01)
24	697.19$^{\circ+}$ (60.98)	649.83° (49.19)	568.82 (110.18)	692.42$^{\circ+}$ (61.73)
25	462.23° (47.61)	468.86$^{\circ*}$ (32.76)	380.82 (45.82)	446.75° (46.50)
26	778.44° (73.50)	773.56° (57.30)	627.40 (93.53)	763.50° (64.44)
27	916.88° (224.75)	927.81° (188.44)	708.85 (249.31)	903.82° (214.06)
28	883.56° (234.51)	986.10$^{\circ\times}$ (225.34)	608.14 (135.37)	892.69° (214.56)
29	434.73° (49.68)	461.07° (77.85)	338.86 (48.06)	458.53° (62.58)
30	399.31° (31.35)	423.14$^{\circ\times*}$ (29.16)	343.71 (41.39)	399.28° (40.28)

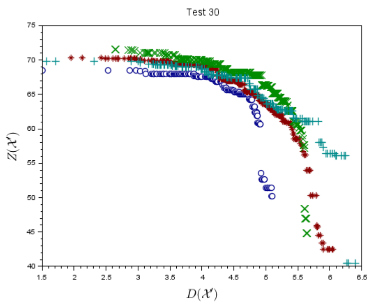
Figure 3 shows some non-dominated fronts of 50% empirical attainment functions for selected tests. These tests and their corresponding figures effectively showcase the trade-offs between the two objectives, while also illustrating certain typical scenarios. First, in all six figures, we observe that operator $R_\circ$ is visibly outperformed by $R_\times$, R_+, and R_*. In Figures 3a, 3b, and 3c, there is a better performance from $R_\times$ compared to the other recombination operators, while



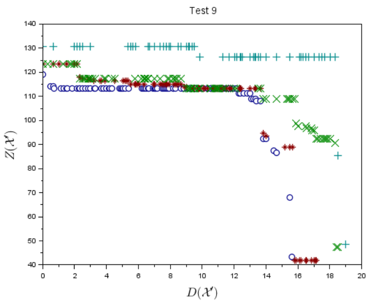
(a) Test 11



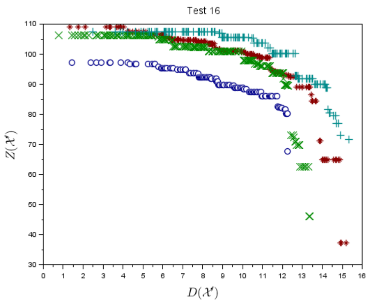
(b) Test 23



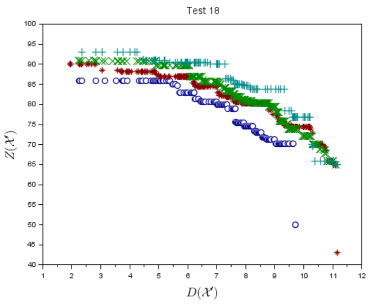
(c) Test 30



(d) Test 9



(e) Test 16



(f) Test 18

Fig. 3: 50% empirical attainment non-dominated fronts using recombination operators $R_\times$ (green $\times$), R_+ (cyan $+$), $R_\circ$ (blue $\bigcirc$) and R_* (red $*$), from selected test instances.

in Figures 3d, 3e, and 3f, it is R_+ that outperforms the other operators. Once again, distinguishing a clear superiority between $R_\times$ and R_+ is challenging, whereas R_* exhibits regular performance (sometimes better sometimes worse), and $R_\circ$ shows a consistently poor overall performance.

Figure 4 displays a box plot summarizing the distribution of the average running times of NSGA-II implementing different recombination operators across all tests. Each rectangular box represents the interquartile range, encapsulating the middle 50% of the corresponding data, with a line inside denoting the median. From this figure, it is evident that the overall lowest running time is achieved by $R_\circ$, while $R_\times$ and R_* have the highest median and worst running times, respectively. On the other hand, the overall running time of R_+ falls somewhat between these two groups. In general, the running times of $R_\times$, R_+, and $R_\circ$ align with expectations based on their corresponding time complexities. However, achieving better performance with one recombination operator often comes with a trade-off. Particularly, the trade-off of quality is associated with an increased number of comparisons, embedded in the non-domination sorting (line 13) in Algorithm 1 and the dominance comparisons (lines 13–16) in Algorithm 2. This, of course, creates an unfair comparison for $R_\circ$ since the experiments are fixed to the number of generations rather than to the number of evaluations. For future studies, it would be intriguing not only to set a fixed number of evaluations but also to assess the impact on quality by removing these additional comparisons.

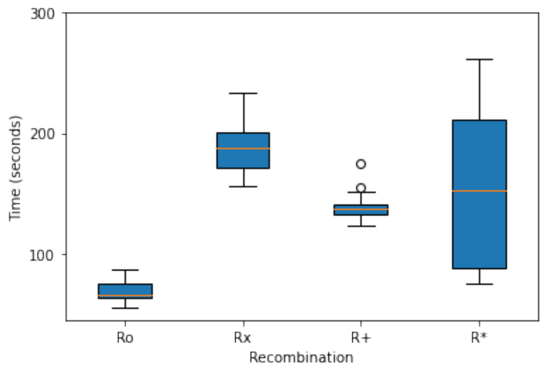


Fig. 4: Box plot comparison of the execution times of the NSGA-II featuring different recombination operators.

E. Discussion

The overall results favor the proposed recombination operators $R_\times$ and R_+. However, certain issues require a careful discussion. Firstly, the parameter values were consistent across all operators. In other words, no specific effort was made to determine optimal parameter values for each recombination. This aspect needs attention in future tests, as the overall performance of these operators may vary depending on the parameter values.

Furthermore, it is important to note that $R_\circ$ is not the only recombination operator that should be tested against the proposed recombination operators. While $R_\circ$ is a more suitable and straightforward recombination as it does not require any modification to the individual solution representation, comparisons with respect to other recombination operators are necessary. However, adaptations to other recombination operators might influence their performance.

Finally, although R_* was intended as a supplementary comparison, the success of $R_\times$ over R_+ in some test instances, and vice versa in others, alongside with the performance of R_*, suggests that a more sophisticated hyper-heuristic scheme might potentially yield better results.

VI. Conclusions and Future Work

In this paper, we addressed the multi-objective Team Formation Problem in Social Networks, optimizing team expertise and density. We introduced two novel grey-box recombination operators ($R_\times$ and R_+) which were incorporated into the NSGA-II. Both recombination operators were designed to make use of the individual representation and problem's structure to produce potentially improved solutions, specifically for the MO TFP-SN. The experiments were performed using the DBLP dataset, tested over 15 benchmark instances from [13], in addition to 15 newly bigger instances. The results were compared against a variation of the uniform crossover from [13] ($R_\circ$) and to a random choice hyper-heuristic (R_*) which randomly selects a new recombination, among the three aforementioned options, to use it every ten generations.

Our experimental results, assessed using both the hypervolume indicator and empirical attainment functions, consistently favored $R_\times$, R_+, and R_* over $R_\circ$. On the one hand, R_* showed competitive results in some of the tests but, overall, it seems to fall slightly behind both $R_\times$ and R_+. On the other hand, $R_\times$ and R_+ showed a similar performance, occasionally outperforming each other. However, R_+ exhibited a more consistent superior performance on larger tests (tests 15–30). An analysis of running times revealed that $R_\circ$ had the lowest overall time, while $R_\times$ and R_* exhibited higher medians. These results, coupled with the ones regarding the hypervolume indicator, illustrate the expected trade-off between computational cost and quality.

Based on these observations, we believe that designing recombination operators that leverage our understanding of the TFP-SN is indeed a promising idea. Future work aims to compare the proposed recombination operators with adaptations of other recombination operators that were excluded due to their unsuitability. Additionally, we plan to explore new and original recombination designs for the TFP-SN using alternative cutting-edge multi-objective evolutionary algorithms. Furthermore, we intend to investigate the effects of combining the strengths of multiple recombination operators through a more sophisticated hyper-heuristic, as has been done in other studies for different problems.

References

[1] F. Ahmed, K. Deb, and A. Jindal. Multi-objective optimization and decision making approaches to cricket team selection. *Applied Soft Computing*, 13(1):402–414, 2013.

[2] G. K. Awal and K. Bharadwaj. Team formation in social networks based on collective intelligence–an evolutionary approach. *Applied intelligence*, 41(2):627–648, 2014.

[3] A. A. Bara'a, A. D. Abbood, A. A. Hasan, C. Pizzuti, M. Al-Ani, S. Özdemir, and R. D. Al-Dabbagh. A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions. *Swarm and Evolutionary Computation*, 63:100885, 2021.

[4] A. L. N. Casotti and R. A. Krohling. A multi-objective formulation for the team formation problem using krippendorff's disagreement and sociometric cohesion with pareto-solutions obtained via evolutionary algorithms. *Computers & Operations Research*, 161:106444, 2024.

[5] 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*, pages 1–22, 2019.

[6] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[8] A. E. Eiben, J. E. Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[9] B. Feng, Z.-Z. Jiang, Z.-P. Fan, and N. Fu. A method for member selection of cross-functional teams using the individual and collaborative performances. *European Journal of Operational Research*, 203(3):652–661, 2010.

[10] A. Gajewar and A. Das Sarma. Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 165–176. SIAM, 2012.

[11] D. Gómez-Zará, A. Das, B. Pawlow, and N. Contractor. In search of diverse and connected teams: A computational approach to assemble diverse teams based on members' social networks. *PloS one*, 17(11):e0276061, 2022.

[12] J. R. Hackman. *Collaborative intelligence: Using teams to solve hard problems*. Berrett-Koehler Publishers, 2011.

[13] J. Juárez and C. A. Brizuela. A multi-objective formulation of the team formation problem in social networks: preliminary results. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 261–268. ACM, 2018.

[14] J. Juárez, C. Santos, and C. A. Brizuela. A comprehensive review and a taxonomy proposal of team formation problems. *ACM Computing Surveys (CSUR)*, 54(7):1–33, 2021.

[15] S. W. Kozlowski and B. S. Bell. Work groups and teams in organizations. *Handbook of psychology*, pages 333–375, 2003.

[16] S. W. Kozlowski and D. R. Ilgen. Enhancing the effectiveness of work groups and teams. *Psychological science in the public interest*, 7(3):77–124, 2006.

[17] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 467–476. ACM, 2009.

[18] E. E. Lawler, S. A. Mohrman, and G. E. Ledford. *Employee involvement and total quality management: Practices and results in Fortune 1000 companies*. Jossey-Bass Inc Pub, 1992.

[19] M. Niveditha, G. Swetha, U. Poornima, and R. Senthilkumar. A genetic approach for tri-objective optimization in team formation. In *2016 Eighth International Conference on Advanced Computing (ICoAC)*, pages 123–130. IEEE, 2017.

[20] M. A. Perez-Toledano, F. J. Rodriguez, J. Garcia-Rubio, and S. J. Ibanez. Players' selection for basketball teams, through performance index rating, using multiobjective evolutionary algorithms. *PloS one*, 14(9), 2019.

[21] 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*, 177:114886, 2021.

[22] R. W. Swezey and E. E. Salas. *Teams: Their training and performance.* Ablex Publishing, 1992.

[23] D. Whitley. Next generation genetic algorithms: a user's guide and tutorial. In *Handbook of Metaheuristics*, pages 245–274. Springer, 2019.

[24] L. Zhang and X. Zhang. Multi-objective team formation optimization for new product development. *Computers & Industrial Engineering*, 64(3):804–811, 2013.

[25] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of pareto set approximations. *Multiobjective optimization: Interactive and evolutionary approaches*, pages 373–404, 2008.