Solving the k,v-Partition Team Formation Problem with Participant Agency

Leona Teunissen - 6294499
Supervisor: Ioanna Lykourentzou
Second Supervisor: Mihaela Mitici
Master: Computing Science

October 12, 2023

Abstract

Giving candidate teammates agency is becoming an increasingly important aspect of the Team Formation Problem(TFP); the problem of forming a team or multiple teams out of a pool of participants. This thesis gives a taxonomy of the several extensions of TFP in the literature. This overview divides the TFP into seven subproblems and shows the gaps in the work field. In addition, one of the extensions is examined, namely the k,v-Partition TFP. A group of candidates is divided into optimal teams based on the skill levels and preferences of the candidates. We present four ILP models to solve this problem with different skill requirements. The models produced an optimal team formation for a group of 25 candidates. Within two hours, the models managed to divide a group of 60 candidates close to optimal. Therefore, we recommend using a heuristic method for partitioning with larger groups.

Contents

1 Introduction

Big problems need a lot of effort to solve and often this cannot be done by an individual. Therefore, teams are created to bundle the strength. A team has the potential to be more than just the sum of its parts. The composition of the team plays a major role in this. The topic of this thesis is team formation.

In order to find the best teams, it is wise to reflect on what makes a team good. Borrego et al. summarize the literature on team effectiveness in five constructs that should be pursued: inclusion of all workers, interdependence among the workers to complete the task, avoidance of incompatibilities and different views, confidence in the ability of others and shared knowledge structures, which can be extra important in interdisciplinary teams (Borrego et al., 2013).

Aranzabal formulated some factors that can withhold good teamwork: different motivation, different expectations, different commitments, personality clashes, dominant and passive members, lack of guidance, task ambiguity, role ambiguity, academic disparity, resistance to work together, lack of interpersonal skills, lack of group form, interpersonal conflict, and poor time management (Aranzabal et al., 2022).

Michaelsen concludes that in order to find a great team we have to search for a group that possesses sufficient intellectual resources to complete the task and in addition for members that interact with each other in a productive way (Michaelsen et al., 2014).

1.1 Problem Description

In the Team Formation problem(TFP) a group or multiple groups have to be found for a specific task out of a pool of experts. More formally we can define TFP as finding a team x or a set of teams X to complete a task t, where x should be a subset of a group of candidates C. The candidates C have a set of skills. A requirement of the team is that the members possess the skills needed to complete task t. (Lappas et al., 2009). We call this the **General TFP**. There are many variants on and within the TFP. We will discuss those variants below. An overview of the variants is shown in Figure 1.

A constraint that is often added as a requirement for the team is that members get along, or that the communication is smooth, such that team members can interact in an effective manner. If the candidates are modeled on a social network G which captures the compatibility among the candidates the problem is often called TFP-SN (Team Formation Problem on Social Networks) in the literature (Samie & Rajabzadeh, 2023). The links in graph G represent the tie strength or communications costs between the candidates. The goal in the TFP-SN is not only finding competent teams but also a team that works well together. In this research, we will deal with the TFP-SN. This idea of focusing on collaboration and presenting the candidates on a social network could be added to every variant in Figure 1.

Originally, Lappas et al. defined TFP as finding one suitable team for one task, we call this the **single TFP**. A lot of researchers extended this TFP to the Multiple Team Formation

Problem (**MTFP**) where multiple teams are selected. In the literature, the MTFP is often denoted as TFP. Therefore, in line with the literature, we will use the term TFP for specific variants. We will use MTFP as an umbrella term to distinguish this problem from the single TFP. If the problem is to find multiple teams for one task we will add the words 'single-task' before the name of the variant. If the problem is to find multiple teams for multiple tasks, we will add the words 'multiple-task' before the variant. If we do not want to specify, but talk about both problems, we will omit the words. In the case of multiple tasks, the selected teams should be assigned or allocated to the task that fits the team the most (Gutiérrez et al., 2016).

A part of the MTFP can be described as the **Top-*k* Team Formation Problem** where k teams should be created out of a pool of candidates instead of just one team (Kargar & An, 2011).

Another variant in the MTFP is where multiple teams are created and every candidate should take place in a team. We call this variant the **Partition Team Formation Problem**. This Partition TFP can be divided into three subproblems.

The first way to make a partition is to set the number of groups and divide the candidates into these groups. We call this the ***k*-Partition Team Formation Problem**.

The second subproblem is the Partition TFP where the group size v is set and the group of candidates should be divided into groups of size v. We call this the ***v*-Partition Team Formation Problem**.

The ***k,v*-Partition TFP** is a combination of the k-partition TFP and the v-partition TFP, where the pool of candidates should be divided into k groups of maximum group size v. There can also be a minimum of the group size.

A slightly different problem than the Partition TFP is the **Overlapping TFP**. In this problem, the pool of candidates is divided into teams too. However, the candidates can work in multiple teams (Anagnostopoulos et al., 2010).

In these problems (see Figure 1) a static environment can be assumed. In a static environment, the number of candidates, the tie strength, and other information stays the same over time. All these problems can also assume a dynamic environment. In a dynamic environment, candidates can be absent for an amount of time, the tie strength between candidates can change or candidates can make new connections for example.

The focus in the literature is mainly on the original TFP and the Top-k TFP. Fewer papers focus on the Partition Team Formation Problem. In most of the research, a static environment is assumed.

1.2 Self-organized vs. Top-down Made Teams

There are several methods to form teams out of a pool of candidates. The teams can be randomly selected, made by the workers (self-organized), made by an algorithm based on some information, or a more hybrid approach where the algorithm selects teams based on some information about the candidates and based on the candidates' preferences. Earlier in

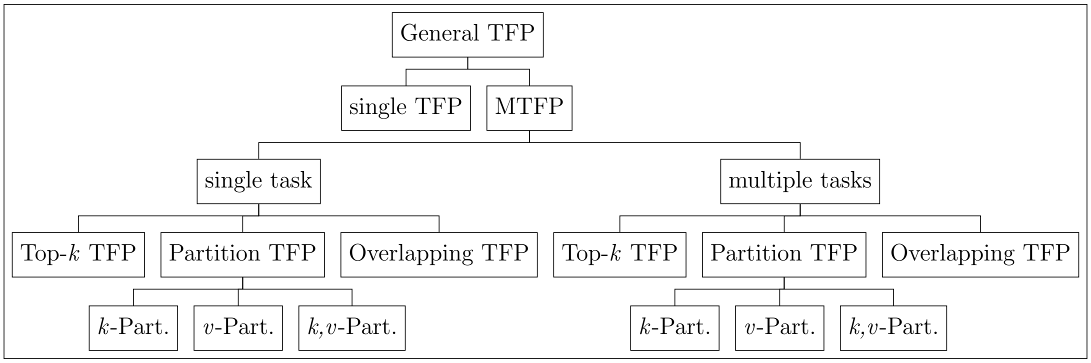

Figure 1: The different variants of the TFP are presented in this figure. All the problems are manifestations of the General Team Formation Problem, with on the left the original problem, which looks for one team. On the right, all the problems looking for multiple teams (MTFP) are shown. The MTFP can be divided into problems that have a single task and problems that work with multiple tasks. To the MTFP belong the Top-k TFP, the Partition TFP and Overlapping TFP, where the Top-k TFP looks for k teams, the Partition TFP divides the whole pool of candidates in teams and the Overlapping TFP assigns every worker to one or multiple teams. The Partition TFP can be divided into the k-Partition, v-Partition, and k,v-Partition TFP.

this chapter we concluded which characteristics a good team should have: enough skills and a productive interaction between the members. In this section, we discuss which methods are capable of making a team with these characteristics.

The first method is assignment by random selection. In this method, every team is selected by a computer, and every candidate in the pool has the same chance to get picked. Assigning candidates randomly into groups is fast, easy, and has some other advantages. For example, there is no chance for discrimination and it gives more diversity in a group in contrast with self-organizing groups. However, research shows random groups perform worse than self-organized teams (Chapman et al., 2006). Also, random group generation can create a suboptimal group, where some important skills could be completely absent.(Bacon et al., 2001)

To avoid this problem most studies incorporate a top-down approach where an algorithm selects the best team based on the information the Decision Maker (DM) has. Even though the TFP is NP-Hard (Lappas et al., 2009) good results are booked with this approach. The advantage is the control of the DM, the requirements of the teams can easily be varied, such as required skills per team or diversity constraints.

However, a disadvantage of this method is that the DM has a limited amount of information about the candidates, and collecting and analyzing this information costs time and effort. Especially to estimate which teams will interact in a productive way. Another disadvantage of this approach is that candidates have little say in what group they want to work with and what a good team should look like.

A solution to these problems is to use a bottom-up approach instead of a top-down approach. In a bottom-up approach, teams are not selected by the DM, but by the candidates. In the literature, these teams are denoted as self-organized teams. Candidates often have more information than a Decision Maker about which candidates they interact with in a productive manner. If candidates form teams themselves, it is likely to end up with similar members in one team. Students, for example, tend to select friends. (Chapman et al., 2006). This can lead to less conflict in a team and Oakley et al. observed less loafing in self-organized teams (Oakley et al., 2007). In some cases, the self-organized teams even perform better than teams selected with a top-down approach or hybrid approach (Vinella et al., 2022). The study of Lykourentzou shows that the members of the self-organized teams felt more productive, creative, and responsible for their work product (Lykourentzou et al., 2020).

However, self-organized teams also tend to be unbalanced in skills, abilities, specialism, gender, or ethnic background and, thus, limit learning opportunities (Aranzabal et al., 2022). In addition, (Bacon et al., 2001) shows that a 'remainder problem' can arise if some candidates have no or little tie strength with other candidates. Candidates who are more embedded in the social network perform better on group projects. So this self-organization approach could especially work well or fair if every candidate gets the chance to know the others.

To obtain the good properties of the self-organized teams and top-down selected teams, the approaches can be combined in a hybrid approach where teams are selected by an algorithm based on costs, skills, but also on preferences of the candidates.

1.3 Our Focus

Our goal in this thesis is to develop a hybrid algorithm-assisted team formation approach, which forms competent teams taking into account both the candidates' skills and their wish to be in a team with teammates they prefer. In other words, we do not only want a method that identifies teams that can deliver good results but also teams the members of which are comfortable working with one another. In order to do that, we want to give participants agency in the team formation process.

We hypothesize that this approach can help ensure that the team will both have the necessary skill requirements to complete the task and at the same time its members will interact in a productive manner.

Since the Single TFP and Top-k TFP have been examined a lot, we want to examine the Partition TFP in this thesis. Specifically the k,v-partition.

In addition, we discuss how our solutions would hold up in a dynamic environment. In a dynamic environment the situation can change, so the optimal solution can change too. Running a TFP algorithm every time there is a change costs a lot of time. In this research we investigate the possibility of re-using previous computations to produce satisfying results in situations that differ only slightly from the original situation, using minimal computational resources. That is, we investigate how we can use previously formed teams to generate new teams based on a reformulation of the original task, without performing the team-formation computation from scratch.

A lot of studies on TFP not only examine the forming of teams but also the allocation of tasks, where different teams have different tasks. Since our focus is on collaboration and candidate agency we will not examine task allocation in our problem. We assume that every team has to perform the same task, like in an educational setting, where every team of students has the same project.

2 Related Work

In this chapter, the literature on a lot of variants of TFP is discussed, following the structure of Figure 1 in a depth-first order. After this overview, some other additions to the TFP that are relevant to this thesis are discussed. For example, participant agency and dynamic environment. These aspects can be added to each TFP variant or to a subset of the variants.

2.1 Overview TFP Variants

2.1.1 TFP - single task

In Chapter 1, TFP is defined as finding a team or multiple teams out of a group of candidates to complete one or multiple tasks. Originally, the focus was on the single TFP, where only one team is created out of a pool of candidates to complete one task. Similar problems already existed before the single TFP was introduced, like the Knapsack Problem and the Minimum Spanning Tree Problem.

In the Knapsack Problem, the problem is to find a set of items that fit in a knapsack while maximizing the sum of the values of the items. The knapsack has a capacity and every item has a size and a value. In order to fit, the sum of the items in the knapsack has to be less or equal to the capacity of the knapsack (Martello & Toth, 1987). This problem is equivalent to the single TFP if we consider workers to have a value (in the form of skills) and a size (in the form of a salary). The objective is to maximize the skills in the knapsack. The capacity of the knapsack could be translated into the budget to create a team.

Minimum Spanning Tree Problem is finding a Minimum Spanning Tree on a weighted graph. The Minimum Spanning Tree is a tree on the graph, which means that it is a set of edges that are connected to each other, without any cycles. For a tree to be a spanning tree, it should 'span' over the whole graph, meaning it should cover all the nodes. The spanning tree should be minimum, in the sense that the weights of the edges should be minimum in relation to all other possible spanning trees (Graham & Hell, 1985).

Lappas et al. use the Minimum Spanning Tree Problem to solve the single TFP on a social network (Lappas et al., 2009). In the paper, the single TFP is defined on a weighted graph, where every node in the graph is a candidate and the weight of the edge between the nodes is the communication cost between the candidates. Every candidate (so every node) has a set of skills. The goal is to find a team for a specific task while minimizing the communication costs. To complete the task some skills are needed, so the team should cover these skills. Lappas et al. give an algorithm where he finds a minimum spanning tree that covers every required skill. Now the spanning is not over every node in the graph but over all the required skills.

More studies focus both on interpersonal relationships and competence to create an excellent team. For example, the study of Zhang and Zhang et al (Zhang & Zhang, 2013), where candidates do not only have a suitable score for the task but also an interpersonal relationship score. The authors propose a multi-objective team optimization model and an algorithm for Pareto Solutions as an alternative.

Other studies that address the single TFP are (Bhowmik et al., 2014), (Majumder et al., 2012)

2.1.2 MTFP - single task

Later extensions to TFP, like the Multiple Team Formation Problem (MTFP), made the definition of the Team Formation Problem wider. Instead of one team, multiple teams now have to be created to complete a specific task. As we already have seen in Figure 1, three TFP problems fall into this category: the single task Top-k TFP, the single task Partition TFP, and the single task Overlapping TFP. Now a short overview of papers that address these problems follows.

The single task Top-k TFP is discussed in the study of Rahman et al (Rahman et al., 2015). In this study, a group of candidates is selected out of a pool of individuals based on the skills and costs of the candidates. Afterward, this group is divided into subgroups where the tie strength within the subgroups is maximized. This results in k top teams that perform the same task.

The single task Top-k TFP can also be solved using Graph Pattern Matching(GPM). GPM is an assignment-based approach, where all candidates are presented in a (weighted) graph. Every candidate has a role or a set of skills. This is called the data graph. The requirements of the team are presented in a pattern graph. For example, the requirement of some roles that should have a strong connection can be represented with a (weighted) edge in the pattern graph. Teams can be formed by searching for the pattern in the data graph. In the study of Kou et al. Graph Pattern Matching is used to solve the single task Top-k TFP. The paper presents an efficient team formation method based on Constraint Pattern Matching. In addition to skill coverage also structure constraints and communications constraints are taken into account. (Kou et al., 2020)

In the study of Salehi et al., an example of the single task Partition TFP is presented (Salehi & Bernstein, 2018). The proposed system that splits the large collective into small teams is called Hive. The size of the teams is set, so Hive is an example of a solution for single task v-partition TFP. Once the candidates are divided into teams, Hive rotates the members of the teams during the project. Hive's network rotation algorithm decides which member of the team should move and to what group based on collaboration in previous rounds. The goal of Hive is to maximize network efficiency and tie strength. Another study that addresses single-task Partition TFP is the study of Vinella et al., in which a top-down, bottom-up, and a hybrid approach are compared to solve the k,v-partition TFP (Vinella et al., 2022).

To our best knowledge, the single-task Overlapping TFP is not yet examined. This could be due to the limited practical applications of this problem.

2.1.3 MTFP - multiple tasks

Another extension of the Team Formation Problem is the multiple-task MTFP, where teams are not just created for one specific task, but for multiple tasks.

Based on set partitioning Das et al. propose a model to solve the multiple-task MTFP where they look for existing teams on a social network. They argue that collaboration is better for existing teams or people that already work together. The requirements for the teams are sufficient skills teams and effective communication. (Daş et al., 2022) Another study takes the same approach of matching tasks to naturally existing groups. Since this problem is NP-hard, Jiang et al. propose a heuristic approach. (Jiang et al., 2019). Another studies that addresses the multiple-task MTFP is (Wu et al., 2021).

To our knowledge, the multiple-task Partition TFP has not yet been examined.

The multiple-task Overlapping TFP is studied by (Anagnostopoulos et al., 2010). A group of candidates is divided among several tasks, such that nobody is left out and no workload is too high. Every skill required by the task is covered by the team members. In contrast with most other (recent) studies on the TFP, there is no focus on interaction or relationships between the members of the team.

2.2 Additions

To make the Team Formation Problem more applicable to different situations aspects like workload, worker agency, and dynamic environment have been added to the Team Formation Problem. These aspects can be added to all the TFP variants or a subset of the variants.

2.2.1 Workload

Some studies that solve the TFP add some constraints on the workload of the team members. In that case, candidates can take place in multiple teams or tasks, as long as their workload does not transcend the threshold. For example, Majumder et al. solve the single TFP, such that the members of the team are socially close, and investigate how such teams can be formed on a social network. In addition, there is also a focus on workload and the members of the team can not be overloaded by their part of the task (Majumder et al., 2012). (Anagnostopoulos et al., 2010) adds workload to the TFP likewise.

2.2.2 Participant Agency

A recent aspect that has been added to the Team Formation Problem is participant agency, sometimes called worker agency. Participant agency can be added by letting the candidates take control in choosing their teammates, and have a say in how a team should look or how the tasks should be assigned to the candidates. A good example of a system with worker agency is Foundry in combination with flash organizations proposed by (Valentine et al., 2017). The authors structure crowds like organizations and the system divides labor into roles with a hierarchy. Workers propose changes by adding new tasks and roles, which are reviewed higher in the hierarchy. These new roles become available on the platform and potential workers can apply for these new roles.

The study of (Vinella et al., 2022) compared three different approaches with three different amounts of worker agency; A top-down, a bottom-up, and a hybrid approach. The hybrid and the bottom-up approach give the workers control in choosing their teammates. In the hybrid approach, this is done by letting the algorithm propose a rotation in the teams after which the workers can accept or decline this offer. In the paper, a measure of teamwork quality is proposed as a weighted sum of skill coverage, interpersonal compatibility, and group size. (Vinella et al., 2022).

Even though the addition of participant agency in the TFP is recent, the problem of matching pairs together based on a preference list is a classic problem called the marriage problem. In the marriage problem, the candidates are partitioned into two groups (in the original problem presented as the men and the women), where every man gives a ranking of the women and every woman gives a ranking of the men. The goal is to find stable marriages, which means that no woman prefers another man over her current husband, while the other man also prefers her over his current wife. For every list of preferences a stable marriage exists. If the pool of workers is not divided into two, so every worker can 'marry' every other worker the problem is called the roommate problem. In the roommate problem, there is no guarantee a stable match can be found.

2.2.3 Dynamic Environment

A study that uses a preferred set of candidates to match them in groups is the study of li et al (Li et al., 2022). In this study, GPM is used to solve the single TFP. Besides the preferred set, a dispreferred set of candidates is added for the candidates who are not appreciated to appear in the final team. The preferred and dispreferred sets of experts can change, without the need to process the GPM again. The previous matching can be used and revised for small changes. So this algorithm is practical in a dynamic environment.

Some studies make their solution more practical for a dynamic environment by giving alternatives for their optimal solution. This is practical in case some candidates are absent for a reason. For example, in the paper of Fathian et al. not only skill coverage and collaboration are considered, but also the reliability of a team. In the paper, a mathematical model is proposed for the reliable team formation problem. The members of a team are divided into two groups: reliable and unreliable. For the unreliable members, a backup is proposed (Fathian et al., 2017). Zhang and Zhang et al. also give alternatives to their optimal solution. Not by proposing an alternative for every unreliable member, but by proposing another algorithm that gives multiple alternative teams. (Zhang & Zhang, 2013)

3 Method

The problem that we examine in this thesis is the single-task *k,v*-Partition TFP. The Partition TFP is not examined a lot, as shown in Chapter 2. The single-task *k,v*-Partition TFP is a variant of the MTFP. We call this problem *k,v*-Partition TFP, instead of *k,v*-Partition MTFP because the MTFP is often denoted as TFP in the literature. We add some participant agency to the problem by asking the candidates to rate their peers. Based on this rating, teams will be formed.

The models we present can be used to solve the single-task *v*-Partition TFP and can easily be extended to solve the multiple-task *k,v*-Partition TFP.

We propose four optimization models to solve this problem. Our first model has a bottom-up approach; it is only based on the reciprocal ratings of the candidates. The second en third models are hybrid models where in addition to the ratings also the skills of the candidates are taken into consideration. The last model adds strong preferences.

We use an optimization model programming to solve this problem since it is an ideal method for problems that need a large amount of computation effort. Other methods like simulated annealing give a heuristic solution and some methods that are suitable for the single or top-k TFP are not suitable for the Partition TFP, such as GPM.

3.1 Formal Problem Description

Let $i \in C$, where $i, (i = 1, \ldots, |C|)$ is a candidate and C is the set of candidates. The set C is partitioned into teams of minimum size A and maximum size B. From this, we can calculate the minimum number of teams $Q(Q = \lceil \frac{|C|}{B} \rceil)$ and the maximum number of teams $N(N = \lfloor \frac{|C|}{A} \rfloor)$.

Let $t \in T$, where $t, (t = 1, \ldots, |T|)$ is a team, T is the set of teams and $|T|$ is an integer in the range of $[Q, R]$.

Between every two candidates, the tie strength is denoted by b_{ij}, where $i, j \in C$ and $i = 1, \ldots, |C|$ and $j = 1, \ldots, |C|$. b_{ij} is an integer in range $[1, 5]$.

Let $s \in S$, where $s, (s = 1, \ldots, |S|)$ is a skill and S is the set of skills. Every candidate i has a skill level l_{is} for every skill s. l_{is} is an integer in the range $[1, 5]$.

We say a candidate i has a skill if the skill level l_{is} is higher or equal to threshold v_s. We say that a team has skill s if and only if at least one of the candidates in the team has skill s.

In addition to the tie strength, there is a strong preference between every two candidates denoted by m_{ij}, where $i, j \in C$ and $i = 1, \ldots, |C|$ and $j = 1, \ldots, |C|$. The preferences can be the input of the candidates or the decision-maker.

Every candidate i should be a member of a team and the team size should be an integer in the range of $[A, B]$. The objective and additional constraints on the problem differ per model.

3.2 Model 1 (Tie Strength): Optimizing Tie Strength

The first model that we propose is only based on the tie strength of the candidates, and thus a bottom-up approach. The only requirements of this model are that the team size should be an integer in the range of $[A, B]$, and every candidate i should be a member of exactly one team.

We solve the Tie Strength Model for all $|T| \in [Q, R]$, where Q is the minimum number of teams based on the input and R is the maximum number of teams based on the input. After solving the models we pick the $|T|$ for which the objective value was the highest.

3.2.1 Inputs

- number of candidates $|C| \in \mathbb{N}$
- minimum team size $A \in \mathbb{N}$
- maximum team size $B \in \mathbb{N}$
- tie strength matrix $D \in \mathbb{N}^{|C| \times |C|}$, with values $b_{ij} \in \{1, \ldots, 10\}$ and $i, j \in \{1, \ldots |C|\}$

3.2.2 Variables

$$x_{ijt} = \begin{cases} 1 & \text{if candidate } i \text{ and candidate } j \text{ are part of team } t \\ 0 & \text{otherwise} \end{cases}$$

3.2.3 Objective

The objective is to maximize the tie strength between the members of every team.

$$\text{maximize} \sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij} \cdot x_{ijt}$$

3.2.4 Constraints

The constraints of this model hold for every other model too and will be called the basic constraints.

$$\sum_{t=1}^{|T|} x_{iit} = 1 \qquad \forall i \in C \tag{1}$$

$$x_{ijt} = x_{jit} \qquad \forall i \in C, \forall j \in C, \forall t \in T \tag{2}$$

$$x_{ijt} \leq x_{iit} \qquad \forall i \in C, \forall j \in C, \forall t \in T \tag{3}$$

$$(1 - x_{iit}) + (1 - x_{jjt}) \geq 1 - x_{ijt} \qquad \forall i \in C, \forall j \in C, \forall t \in T \tag{4}$$

$$\sum_{i=1}^{|C|} x_{iit} \leq B \qquad \forall t \in T \tag{5}$$

$$\sum_{i=1}^{|C|} x_{iit} \geq A \qquad \forall t \in T \tag{6}$$

$$x_{ijt} \in \{0, 1\} \tag{7}$$

1. Every candidate is member of exactly one team.
2. If candidate i and candidate j are together in team t, then candidate j and candidate i are together in team t.
3. If candidate i and candidate j are together in team t, then candidate i is in team t.
4. If candidate i is in team t and candidate j is in team t, then candidate i and j are together in team t.
5. The maximum team size is B.
6. The minimum team size is A.
7. Variable x_{ijt} is binary.

3.3 Model 2 (Skilled Teams): Optimizing Tie Strength with Minimum Skills per Team

The model description for this model with skill constraints is similar to the model description of the Tie Strength model, where we have candidates, teams, and tie strength between the candidates. However, in this model, the candidates have skills and a corresponding skill level. In addition, we introduce a minimum required number of skills r, which is the same for each team t. So, each team needs r skills to be present, this could be any subset of skill set S. To create the Skilled Teams Model, an extra variable is needed.

The number of teams $|T| \in [Q, R]$ that is optimal for the Skilled Team Model is determined in the same way as in the previous model.

3.3.1 inputs

Next to the inputs of the previous model, the following inputs are added:

- number of skills $|S| \in \mathbb{N}$
- threshold skill level vector $V \in \mathbb{N}^{|S| \times 1}$, with values $v_s \in \{1, \ldots, 5\}$, with $s \in \{1, \ldots, |S|\}$
- skill matrix $F \in \mathbb{N}^{|C| \times |S|}$, with values $l_{is} \in \{1, \ldots, 5\}$, $i \in \{1, \ldots |C|\}$ and $s \in \{1, \ldots, |S|\}$
- minimum required skills $r \in \{1, \ldots, |S|\}$

3.3.2 Preprocessing

For this model, the skill level of the candidates should be binary. If the skill level of candidate i is higher or equal to the threshold v_s we set h_{is} to 1, otherwise, h_{is} is 0.

3.3.3 Variables

The first variable is equal to the variable in the Tie Strength Model:

$$x_{ijt} = \begin{cases} 1 & \text{if candidate } i \text{ and candidate } j \text{ are part of team } t \\ 0 & \text{otherwise} \end{cases}$$

Another variable is added for ensure every team has enough skills:

$$y_{ts} = \begin{cases} 1 & \text{if skill } s \text{ is present in team } t \\ 0 & \text{otherwise} \end{cases}$$

3.3.4 Objective

The objective of this model is equal to the objective of the Tie Strength Model, which is to maximize the tie strength between the members of every team.

$$\text{maximize} \sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij} \cdot x_{ijt}$$

3.3.5 Constraints

See Chapter 3.2.4 for the basic constraints that also hold for this model. The added constraints are listed in this section. The first two constraints that are added are formulated using an indicator function, which is available in a solver like Gurobi.

$$\text{if } y_{ts} = 0, \text{ then} \sum_{i=1}^{|C|} h_{is} \cdot x_{iit} = 0 \qquad \forall t \in T, \forall s \in S \tag{8}$$

$$\text{if } y_{ts} = 1, \text{ then} \sum_{i=1}^{|C|} h_{is} \cdot x_{iit} \geq 1 \qquad \forall t \in T, \forall s \in S \tag{9}$$

$$\sum_{s=1}^{|S|} y_{ts} \geq r \qquad \forall t \in T \tag{10}$$

$$y_{ts} \in \{0, 1\} \tag{11}$$

8. This constraint makes sure that if $y_{ts} = 0$, then there is no member of team t that has skill s.
9. This constraint makes sure that if $y_{ts} = 1$, there is at least one member in the team that possesses skill s.
10. For each team, the number of present skills is higher or equal to the required skills.
11. Variable y_{ts} is binary.

For a solver that does not have indicator functions, the following constraints can be used instead of constraints 8 and 9. The U and L stand for an upperbound and lowerbound respectively. In this thesis, we will write the constraints using the indicator functions, since it is more intelligible.

$$(\sum_{i=1}^{|C|} h_{is} \cdot x_{iit}) - 1 \leq U \cdot y_{ts} \qquad \forall t \in T, \forall s \in S$$

$$L \cdot (1 - y_{ts}) \leq (\sum_{i=1}^{|C|} h_{is} \cdot x_{iit}) - 1 \qquad \forall t \in T, \forall s \in S$$

3.4 Model 3 (Max Skilled Teams - MST): Optimizing Tie Strength and Skill

In the previous model, every team should have the required number of skills (or more). If one of the teams fails to collect the required skills, the model will not give a solution. In that case, the number of required skills could be decreased. However, in practice, this is not always desirable or possible. Therefore, we present another model which maximizes the number of competent teams. This model is more flexible because it will also give a solution if not all skill-related constraints can be met. A competent team is defined as a team that has the required number of skills r or more.

An extra variable z_t is added, such that the number of competent teams can be maximized.

The number of teams $|T| \in [Q, R]$ that is optimal for the Max Skilled Team Model is determined in the same way as in the previous model.

3.4.1 inputs

The inputs are equal to the inputs in previous models. One addition is made:

- weight $\alpha \in \mathbb{R}$

3.4.2 Preprocessing

Like in the Skilled Teams Model, the skill levels of the candidates are binarized. If the skill level of candidate i is higher or equal to the threshold v_s we set h_{is} to 1, otherwise, h_{is} is 0.

3.4.3 Variables

The first two variables are the same as in the previous model.

$$x_{ijt} = \begin{cases} 1 & \text{if candidate } i \text{ and candidate } j \text{ are part of team } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ts} = \begin{cases} 1 & \text{if skill } s \text{ is present in team } t \\ 0 & \text{otherwise} \end{cases}$$

For this model, we introduce an extra variable z_t, which denotes if team t is competent.

$$z_t = \begin{cases} 1 & \text{if team } t \text{ has the required number of skills or more} \\ 0 & \text{otherwise} \end{cases}$$

3.4.4 Objective

The objective is the weighted sum of the tie strength between the members of the teams and the number of teams with (at least) the required number of skills.

$$\text{maximize } \alpha(\sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij} \cdot x_{ijt}) + (1 - \alpha) \sum_{t=1}^{|T|} z_t$$

3.4.5 Constraints

See Chapter 3.2.4 for the basic constraints that also hold for this model. The added constraints are listed in this section.

$$\text{if } y_{ts} = 0, \text{ then} \sum_{i=1}^{|C|} h_{is} \cdot x_{iit} = 0 \qquad \forall t \in T, \forall s \in S \tag{12}$$

$$\text{if } y_{ts} = 1, \text{ then} \sum_{i=1}^{|C|} h_{is} \cdot x_{iit} \geq 1 \qquad \forall t \in T, \forall s \in S \tag{13}$$

$$y_{ts} \in \{0, 1\} \tag{14}$$

$$\text{if } z_t = 0, \text{ then} \sum_{s=1}^{|S|} y_{ts} \leq r - 1 \qquad \forall t \in T \tag{15}$$

$$\text{if } z_t = 1, \text{ then} \sum_{s=1}^{|S|} y_{ts} \geq r \qquad \forall t \in T \tag{16}$$

$$z_t \in \{0, 1\} \tag{17}$$

12. This constraint makes sure that if $y_{ts} = 0$, then there is no member of team t that has skill s.

13. This constraint makes sure that if $y_{ts} = 1$, there is at least one member in the team that possesses skill s.

14. Variable y_{ts} is binary.

15. This constraint makes sure that if $z_t = 0$, team t does not have the required number of skills.

16. This constraint makes sure that if $z_t = 1$, team t has the required number of skill or more.

17. Variable z_t is binary.

3.5 Model 4 (Pref): Optimizing Tie Strength, Skill, and Preferences

Our last model is almost equal to the previous model. However, in this model, strong preferences are added. The strong preferences can be the input of the candidates or the decision-maker in the form of a preference matrix. This input can be given before any proposal of teams is given or after a proposal. The good matches that are made can be given as a positive strong preference for the next run and the not desired matches can be given as a negative strong preference.

If the strong preference m_{ij} is 1, the candidates should be together in a team. If the strong preference between i and j is -1, the candidates should not be together in a team. In some cases the problem because infeasible if too many preferences are given. That is why the realized preferences are maximized, in contrast with adding hard constraints.

The number of teams $|T| \in [Q, R]$ that is optimal for the Pref Model is determined in the same way as in the previous model.

3.5.1 inputs

The inputs are similar to the inputs of the previous model. The strong preference matrix is added and an extra weight, since the objective is extended with another term.

- strong preference matrix $M \in \mathbb{N}^{|C| \times |C|}$, with values $m_{ij} \in \{-1, 0, 1\}$ and $i, j \in \{1, \dots |C|\}$
- weight $\alpha, \beta \in \mathbb{R}$

3.5.2 Preproccessing

Like in the other models the skill levels of the candidates are binarized.

3.5.3 Variables

The variables of this model are equal to the variables of the Max Skilled Teams model.

$$x_{ijt} = \begin{cases} 1 & \text{if candidate } i \text{ and candidate } j \text{ are part of team } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ts} = \begin{cases} 1 & \text{if skill } s \text{ is present in team } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_t = \begin{cases} 1 & \text{if team } t \text{ has the required number of skills or more} \\ 0 & \text{otherwise} \end{cases}$$

3.5.4 Objective

The objective is the weighted sum of the tie strength between the members of the teams, the number of teams with (at least) the required number of skills, the number of positive preferences that are realized, minus the number of negative preferences that are realized(, while they should not).

$$\text{maximize } \alpha \sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij} \cdot x_{ijt} + \beta \sum_{t=1}^{|T|} z_t + (1 - \alpha - \beta) \sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} x_{ijt} \cdot m_{ij}$$

3.5.5 Constraints

The constraints of this model are equal to the constraints of the Max Skilled Teams Model.

3.6 Evaluation Metrics

In this section, we give a description of the metrics that are used to measure the solutions. The measures are divided into four categories, namely: Global, the weak preferences (WP), the skills, and the strong preferences (SP) metrics.

3.6.1 Global Metrics

In this section, the global metrics are listed and defined. These metrics give an overall view of how the models perform.

- Team Quality Measure (TQM)

We propose the TQM as the weighted sum of the number of teams that can perform the task, tie strength between the candidates in the teams, and the number of strong preferences that have been included in the solution:

$$TQM = \alpha \frac{\sum_{t=1}^{|T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij} \cdot x_{ijt}}{\sum_{i=1}^{|C|} \sum_{j=1}^{|C|} b_{ij}} + \beta \frac{\sum_{t=1}^{|T|} z_t}{|T|} + (1 - \beta - \alpha) \frac{\sum_{i=1}^{|C|} \sum_{j=1}^{|C|} m_{ij} \cdot x_{ijt}}{\sum_{i=1}^{|C|} \sum_{j=1}^{|C|} m_{ij}}$$

For the Tie Strength and Skilled Teams Model variable z_t is not defined, so we calculated the number of competent teams based on the solution.

The idea of the TQM is to give a value that is the summary of how well the models perform considering tie strength, skills, and strong preferences.

- Gap

The Gap is the difference between the lower bound and the upper bound of the objective value. The higher the gap, the higher the risk that the solution is not close to the optimal solution.

- Runtime

The runtime is presented in a format containing hours, minutes, and seconds (hh:mm:ss).

3.6.2 Weak Preference Metrics

The second category considers the weak preferences (WP) metrics.

- Sum WP

Sum WP gives the sum of all tie strengths that are realized in the solution times the values of tie strengths. This is the same as the objective function for the Tie Strength and the Skilled Teams models.

- Realized Tie Strength

There are five metrics of the realized tie strengths. These metrics show how many weak preferences are realized per tie strength. So the metric Realized 1, shows the number of weak preferences of value 1 that are realized.

- Average WP

The average WP is the average of the realized tie strength between each pair of members of the team. So every team has a WP average. The average WP of all the teams is an average of all the averages.

- Best team

The team with the highest Average WP is the best team in the WP category.

- Worst team

The worst team in the WP category is the team with the lowest average WP.

- Std

The Standard Deviation shows how much the averages of the teams differ from the average WP of all teams.

3.6.3 Skills Metrics

The first row contains the number of teams that possess the required number of skills. In the next three rows the average of the number of skills in a team, the number of skills in the best team and the worst team.

- # skilled teams

The # skilled teams metric is the number of competent teams found in the solution. A team is competent if it has at least the required number of skills. The required number of skills differs per experiment.

- Average of teams (# skills)

The average of teams is the number of skills the teams have on average.

- Best team (#skills)

The best team considering the number of skills is the team with the highest number of skills.

- Worst team (#skills)

The worst team considering the number of skills is the team with the lowest number of skills.

- Average Coverage

Every candidate with a skill has a skill level of 1 to 5. The skill coverage of a team is the sum of the levels of the skills in the team. The average coverage is the skill coverage the teams have on average.

- Best team

The best team considering skill coverage is the team with the highest skill coverage.

- Worst team

The worst team considering skill coverage is the team with the lowest skill coverage.

3.6.4 Strong Preference Metrics

The last category consists of the realized strong preferences (SP) metrics.

- Realized positive

The realized positive metric gives the number of realized positive preferences in the solution.

- Realized negative

This metric shows the number of realized negative strong preferences in the teams. Ideally, this number is close to zero.

- Realized pos - neg

To give a summary of how well the model performs on the realized strong preferences, the realized negative strong preferences are subtracted from the realized positive strong preferences.

3.7 Experiments and Parameters

To test and compare our models, we did six experiments using a dataset from a real-world, university-level, and project-based course. The course consisted of 60 students, split into groups of 4 to 6 people. We are provided with an anonymized *skill matrix*, denoting the skill level on a 0-5 scale of each student across 6 distinct skills, and a *tie strength matrix* denoting the students' preferences for teammates, which were obtained through the process of team dating Lykourentzou et al., 2017. The tie strength between two candidates can be an integer in the range of [1, 5]. However, some of the tie strength values are equal to 10. We interpret this value as a positive strong preference and a tie strength of 5. If the tie strength between two candidates is equal to 1, we interpret this as a negative strong preference (and a tie strength of 1). During the process of team dating a candidate does not get the chance to meet every other candidate, which results in missing values in the tie strength matrix. Missing values are replaced with a value of 3 since this is a value we consider as neutral.

In experiments 1, 3, and 5, we used the first 25 candidates in the dataset as the candidate set C. In experiments 2, 4, and 6, we used the first 60 candidates in the dataset as the candidate set C. For all the experiments, we used a team size minimum of 4 and a team size maximum of 5, since this team size often occurs in practice. The thresholds of the skill levels were set to 4. In the first two experiments, the required number of skills is set to 4. In the next two experiments, 5 skills were required to make a team competent and in the last two experiments, 6 skills were required. We chose the weights α and β in such a way no objective was prioritized over the other. A teacher's heuristic solution serves as a baseline for comparison in experiment 6.

For running the models we used Gurobi with the Python extension gurobipy. We worked in Jupyter Notebook. The laptop used has an Intel Core i7-1165G7 (11th Gen) processor and a 2.80 GHz 8 Cores CPU.

4 Results

4.1 Results Experiment 1 - 25 Candidates - 4 required skills

In the first experiment, the models solve the k,v-Partition TFP with a candidate set C of 25 candidates. This set is a sample out of the whole dataset. The minimum and maximum team size is set to 4 and 5 retrospectively. The skill level thresholds and the required number of skills for a team are set to 4. Every part of the objective has the same weight, so no objective is prioritized over the other. The results of this experiment can be found in table 1. A description of the used metrics is given in Chapter 3.6.

	Model	Tie Strength	Skilled Teams	MST	Pref
Global	TQM	1.19	**1.27**	1.23	**1.27**
	Gap	0.0	0.0	0.0	0.0
	Runtime (hh:mm:ss)	**00:06:15**	00:09:42	00:31:03	00:09:50
Weak Preferences	Sum WP	372	372	372	372
	Realized 1	0	0	0	0
	Realized 2	1	1	2	1
	Realized 3	73	73	71	73
	Realized 4	4	4	5	4
	Realized 5	27	27	27	27
	Average WP	3.53	3.53	3.53	3.53
	Best team	4.06	3.88	4.06	3.88
	Worst team	**3.0**	**3.0**	2.94	**3.0**
	Std	0.39	**0.34**	0.41	**0.34**
Skills	#skilled teams	6	6	6	6
	Average of teams	4.0	**4.33**	4.17	**4.33**
	Best team	4.0	5.0	5.0	5.0
	Worst team	4.0	4.0	4.0	4.0
	Average coverage	28.17	28.17	28.17	28.17
	Best team	36.0	**37.0**	36.0	**37.0**
	Worst team	20.0	**22.0**	20.0	**22.0**
Strong Preferences	Realized positive	19	**21**	20	**21**
	Realized negative	0	0	0	0
	Realized pos - neg	19	**21**	20	**21**

Table 1: The last four columns of this table contain the results of the four models presented in Chapter 3, where MST stands for the Max Skilled Teams model. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), average weak preference (WP av.), skills, skill coverage, and strong preferences (SP). If a value is bold, this means the model outperforms the models without a bold value on this characteristic. See Chapter 3.6 for a more detailed description of the table and see below for a description of the results.

In table 1, the first category contains the global metrics TQM, gap, and runtime. The TQM is defined (in Chapter 3.6.1) as the weighted sum of the number of competent teams, the tie strength between the candidates in the teams, and the number of realized strong preferences. This gives an idea of how the models perform overall. The gap is zero for all the models, which means all the models solved the k,v-Partition TFP optimally for 25 candidates. The Skilled Teams and Pref models score similarly high on the global characteristics. The Max Skilled Teams does not score far below these models regarding the TQM but runs substantially slower. The Tie Strength model runs slightly faster than the better-scoring models, however, it scores the worst on the TQM.

The realized weak preferences are very similar for every model. However, there is some difference notable if we compare the tie strength distribution in the teams. If the goal is to make a set of teams that perform evenly well, the goal may be to make the teams resemble each other. In the Max Skilled Teams model the teams' tie strength averages differ the most from each other. In the Skilled Teams and Pref model, the teams resemble each other the most, according to the standard deviation.

In the third category, the characteristics concerning the skills are listed. In the solutions, all teams have four or five skills, so all teams are competent. The Skilled Teams and the Pref model score the best on the average number of skills per team, so they give the most teams with five skills and the least teams with only four skills. The Tie Strength model only gives teams with four skills.

The average skill coverage of a team is equal for every model. The Skilled Teams and the Pref model produce the team with the highest skill coverage and also the worst team has a higher score than the worst teams of the other models. So in this experiment on Skill coverage, the Skilled Teams and the Pref models score the highest.

In the last category, once more, the Skilled Teams and the Pref models outperform the other models in this experiment. The models realize the most positive strong preferences and no negative strong preferences.

4.2 Results Experiment 2 - 60 Candidates - 4 required skills

In the second experiment, the whole dataset is used with a candidate set C of 60 candidates. The models do not give an optimal solution within two hours. Therefore the experiment is executed three times and the average of the results is presented in Table 2.

The parameters that are used for this experiment are equal to the parameters of the first experiment, except for the time-limit of 7200 seconds (2 hours).

A more detailed description of the metrics in Table 2 can be found in Chapter 3.6. The table is divided into categories. The first category contains some global metrics like TQM (defined in Chapter 3.6.1). The runtime is the same for all models since a time limit of two hours was set. In these two hours, the Pref model managed to achieve the highest score on TQM, with 1.39. The Tie Strength model has a higher gap.

In the second category, the Tie Strength and Pref model scored similarly on the Sum WP. Compared to the other models, the distribution of the weak preferences is slightly better for

	Modelname	Tie Strength	Skilled Teams	MST	Pref
Global	TQM	1.3	1.29	1.35	**1.39**
	Gap	0.68	**0.42**	**0.42**	**0.42**
	Runtime	02:00:00	02:00:00	02:00:00	02:00:00
Weak Preferences	Sum WP	944.67	936.0	939.0	**945.0**
	Realized 1	1.0	**0.0**	0.33	**0.0**
	Realized 2	2.0	3.0	**1.67**	3.0
	Realized 3	115.33	121.0	120.33	116.0
	Realized 4	**14.67**	13.0	14.0	14.0
	Realized 5	**107.0**	103.0	103.67	**107.0**
	Average of teams	**3.94**	3.9	3.91	**3.94**
	Best team	4.5	4.5	4.5	4.5
	Worst team	3.04	**3.31**	**3.31**	**3.31**
	Std	0.47	**0.4**	0.41	0.43
Skills	#skilled teams	13.0	**15.0**	**15.0**	**15.0**
	Average # skills	4.33	**4.6**	**4.6**	4.33
	Best team	6.0	5.0	5.0	5.0
	Worst team	2.0	**4.0**	**4.0**	**4.0**
	Average coverage	30.2	30.2	30.2	30.2
	Best team	42.0	46.0	44.67	46.0
	Worst team	13.0	**18.0**	**18.0**	**18.0**
Strong preferences	Realized positive	68.67	64.0	68.67	**71.0**
	Realized negative	1.0	**0.0**	0.33	**0.0**
	Realized pos - neg	67.67	64.0	68.33	**71.0**

Table 2: The last four columns of this table contain the results of experiment 2 using the four models presented in Chapter 3. The results are an average of three runs. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), the average WP of the teams, the number of skills, the skill coverage, and Strong Preferences (SP). The bold values indicate which models scored best on that characteristic. See below for a more detailed description.

the Pref model since it has no realized weak preferences of 1 and the most realized weak preferences of 5.

In the third category, the WP average is considered. The solutions of the models are very similar in this category, except that the Tie Strength Model's worst team is slightly worse. If we compare the rest of the models, the Pref model has the highest average weak preference, but also the highest standard deviation, so there is no obvious model that outperformed the others in this category.

In the fourth category, the number of skills are considered. The Tie Strength model has a lower score on the number of competent teams and also has the worst team. The other three models perform better on these characteristics. The teams produced by the Skilled Teams and Max Skilled Teams models possess more skills on average. so these models outperformed the others in this category.

Again, the Tie Strength model produced the worst team with the lowest score. The average skill coverage is the same for the other three models as well as the skill coverage for the worst team. The skill coverage of the best team is slightly lower. This means the difference between the worst and best teams is less high. So, if the goal is to create similar teams, the Max Skilled Teams model is the winner in this category.

In the last category, the Pref model outperforms the others. It realizes the most positive preferences and no negative preferences. The Skilled Teams model scored the lowest in this category.

In conclusion, the Pref model scored best in most categories. However, the teams in the Skilled Teams and the Max Skilled Teams models possess a higher number of skills on average, the standard deviation in the weak preferences is lower for the Skilled Teams model and the difference in skill coverage between the best and worst team is lower for the Max Skilled Teams model. So if similar performance of the teams is prioritized, the Skilled Teams or Max Skilled Teams model may be a better choice.

4.3 Experiment 3 - 25 candidates - 5 required skills

In the third experiment, the models solved the k,v-Partition TFP with a candidate set C of 25 candidates. This set is a sample out of the whole dataset. The minimum and maximum team size is set to 4 and 5 retrospectively. The skill level thresholds are set to 4, and the required number of skills for a team is set to 5. The results of this experiment can be found in table 3.

In this section, a detailed description of the results in table 3 is given. The table is structured in the same way as in the first experiment. For an explanation of the structure, categories, and characteristics in the table visit Paragraph 3.6.1.

In the global category, the Pref model scored highest on the TQM (defined in chapter 3.6.1) and has the lowest runtime, which indicates that the Pref model scored best on average in this experiment. The gap is for each model 0, so all models produced an optimal solution.

On the weak preferences, the Tie Strength model scores best overall. It scores the highest on the sum of the weak preferences and the realized weak preferences of 4 and 5. The produced teams also have a higher tie strength on average. The Pref model realizes less low tie strengths (tie strengths of 1 or 2) and the standard deviation is less high, which means there are fewer differences between the teams.

On skills, the last three models outperform the Tie Strength model. Between the models is little difference. The best team in the Pref model has a lower skill coverage than the best teams produced by the Skilled Teams and the Max Skilled Teams model. However, the average skill coverage is the same. So if the goal is to produce similar teams, the Pref model is preferred in this category.

In the strong preferences category, the Tie Strength scored the best. The model realized two more positive strong preferences compared to the other models and realized no negative

	Model	Tie Strength	Skilled Teams	MST	Pref
Global	TQM	0.86	1.06	1.06	**1.1**
	Gap	0.0	0.0	0.0	0.0
	Runtime	00:09:15	00:16:40	00:22:38	**00:09:05**
Weak preferences	Sum WP	**372**	357	357	356
	Realized 1	**0.0**	1.0	1.0	**0.0**
	Realized 2	1.0	**0.0**	**0.0**	**0.0**
	Realized 3	73.0	81.0	81.0	84.0
	Realized 4	**4.0**	2.0	2.0	1.0
	Realized 5	**27.0**	21.0	21.0	20.0
	Average WP	**3.53**	3.37	3.37	3.36
	Best team	4.06	3.75	3.75	3.75
	Worst team	3.0	3.0	3.0	3.0
	Std	0.39	0.32	0.32	**0.29**
Skills	#skilled teams	0	**6**	**6**	**6**
	Average #skills	4.0	**5.0**	**5.0**	**5.0**
	Best team	4.0	5.0	5.0	5.0
	Worst team	**4.0**	5.0	5.0	5.0
	Average coverage	28.17	28.17	28.17	28.17
	Best team	36.0	37.0	37.0	33.0
	Worst team	**20.0**	24.0	24.0	24.0
Strong preferences	Realized positive	**19**	17	17	17
	Realized negative	**0**	1	1	**0**
	Realized pos - neg	**19**	16	16	17

Table 3: The last four columns of this table contain the results of the four models presented in Chapter 3. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), Skills, and Strong Preferences (SP). See Paragraph 4.1 for a more detailed description of the table.

strong preferences. The Pref model did not realize a negative strong preference either.

The Tie Strength model outperformed the other models in the category of weak preferences. However, the model did not succeed in the category of skills. A reason for this is that the Tie Strength model only focuses on maximizing the tie strength within the teams and not on the skills of the teams, like the other models. The other models perform almost evenly well on the skills. However, the teams produced by the Pref model have a slightly lower tie strength on average and the best team has a lower skill coverage than the best teams of the Skilled Teams and the Max Skilled Teams models. To compensate, the Pref model performs slightly better on the strong preferences, which makes the three models equally good if we consider the solutions. However, the Pref model runs a lot faster, so this is the best-performing model for this experiment.

4.4 Experiment 4 - 60 candidates - 5 required skills

In the fourth experiment, the same parameters are used as in the previous experiment. So in every team, five skills are required to classify the team as competent, and the skill thresholds are set to 4. In contrast with the third experiment, the models try to solve the Partition TFP for 60 candidates within two hours. In this section, the results of the experiment are described in detail. A description of the categories and the characteristics in the table can be found in section 3.6.

	Model	Tie Strength	Skilled Teams	MST	Pref
Global	TQM	1.16	-	1.35	**1.36**
	Gap	0.1	-	**0.04**	0.15
	Runtime	02:00:00	-	02:00:00	02:00:00
Weak preferences	Sum WP	**943.0**	-	936.0	926.0
	Realized 1	1.0	-	1.0	**0.33**
	Realized 2	1.0	-	1.0	1.0
	Realized 3	116.0	-	121.0	127.33
	Realized 4	**18.0**	-	15.0	15.0
	Realized 5	**104.0**	-	102.0	96.33
	Average WP	**3.93**	-	3.9	3.86
	Best team	4.5	-	4.5	4.5
	Worst team	3.12	-	3.12	**3.21**
	Std	0.47	-	**0.45**	0.47
Skills	#skilled teams	8.0	-	**13.0**	**13.0**
	Average #skills	4.4	-	4.73	**4.82**
	Best team	6.0	-	6.0	6.0
	Worst team	2.0	-	2.0	**3.0**
	Average coverage	30.2	-	30.2	30.2
	Best team	53.0	-	42.0	42.0
	Worst team	13.0	-	13.0	**21.33**
Strong preferences	Realized positive	67.0	-	**72.0**	**72.0**
	Realized negative	1.0	-	1.0	**0.33**
	Realized pos - neg	66.0	-	71.0	**71.67**

Table 4: The last four columns of this table should contain the results of the four models presented in Chapter 3. The Skilled Teams model did not give a solution. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), Skills, and Strong Preferences (SP). If a value is bold, the model outperforms the other models without a bold value on this characteristic. See below for a more detailed description of the results.

The Skilled Teams model could not find a solution, since there is no solution where every team is competent with the parameters of this experiment.

On TQM, the Max Skilled Teams and the Pref model score similarly well, while the Tie Strength model is lacking. This gives a summary of how well the Tie Strength performs on the total sum of tie strengths, the number of competent teams, and the number of strong preferences that are realized. The gap is lowest for the Max Skilled Teams model.

The best-scoring model in the weak preferences category is the Tie Strength model. The sum of the tie strength between the members of the teams is highest for the Tie Strength model, just as the realized tie strength of 4 and 5. The model that realized the least number of lower tie strengths is the Pref model. Since the difference in the realized lower tie strengths is not that high, while the difference in the realized higher tie strengths is definitely present, the Tie Strength is still the best scoring model in the weak preferences.

This is also visible while looking at the average weak preferences. The Tie Strength model scores highest on the average WP. The best teams produced by the model have the same average tie strength. The worst team produced by the Pref model has the highest tie strength average and the Max Skilled Teams shows the lowest standard deviation. So choosing a preferred model in this category depends on what features are prioritized. If creating similar teams is the priority, the Max Skilled Teams model is preferred. If creating no teams with a low weak preference average is a priority, the Pref model is preferred. If a high average of realized tie strengths is the priority, the Tie Strength model is preferred over the other models in this category.

A more obvious difference between the models is visible when skills are considered. Of the 15 teams, the Tie Strength models produced 8 competent teams. Whereas the Max Skilled Teams and the Pref model produce 13 competent teams. The teams produced by the Pref model have a higher number of skills on average. In addition, the worst team has more skills and a higher skill coverage compared to the worst teams of the other models.

On the strong preferences, once more, the Tie Strength model performs less well in comparison. The other models perform similarly. However, the Pref realizes less negative strong preferences on average. This makes the Pref model the preferred model in the category.

In summary, the Tie Strength model performs better on the weak preferences, and worse on the skills and strong preferences. The Pref model performs better on the skills and strong preferences and worse on the weak preferences. On weak preferences, the Max Skilled Teams model performs better than the Pref model and worse than the Tie Strength model. On the skills and strong preferences, the Max Skilled Teams model performs better than the Tie Strength model but slightly worse than the Pref model. Depending on the priorities, another model should be chosen.

4.5 Experiment 5 - 25 candidates - 6 required skills

In this experiment, a sample of 25 candidates is used as a candidate set, like in the first experiments. The minimum and maximum team size is set to 4 and 5 retrospectively. The skill level thresholds are set to 4, and the required number of skills for a team is set to 6. The results of this experiment can be found in table 4. In chapter 3.6 an explanation of the characteristics and categories of the table is given. In this section, a detailed description of the results is given.

For all models that could give a solution in this experiment, the gap is 0. This means the solutions are optimal (for the given parameters). The Pref model has the highest TQM and

	Model	Tie Strength	Skilled Teams	MST	Pref
Global	TQM	0.86	-	0.94	**0.98**
	Gap	0.0	-	0.0	0.0
	Runtime	00:10:14	-	00:12:03	**00:04:59**
Weak preferences	Sum WP	**372**	-	361	361
	Realized 1	**0.0**	-	1.0	**0.0**
	Realized 2	**1.0**	-	**1.0**	2.0
	Realized 3	73.0	-	77.0	78.0
	Realized 4	**4.0**	-	3.0	2.0
	Realized 5	**27.0**	-	23.0	23.0
	Average WP	**3.53**	-	3.41	3.43
	Best team	4.06	-	3.75	4.06
	Worst team	3.0	-	3.0	3.0
	Std	0.39	-	**0.29**	0.37
Skills	#skilled teams	0	-	**3**	**3**
	Average #skills	4.0	-	**4.83**	**4.83**
	Best team	4.0	-	6.0	6.0
	Worst team =	**4.0**	-	3.0	3.0
	Average coverage	28.17	-	28.17	28.17
	Best team	36.0	-	37.0	45.0
	Worst team	**20.0**	-	**20.0**	16.0
Strong preferences	Realized positive	**19**	-	18	18
	Realized negative	**0**	-	1	**0**
	Realized pos - neg	**19**	-	17	18

Table 5: The last four columns of this table contain the results of the four models for experiment 5. The Skilled Teams Model did not give a solution. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), Skills, and Strong Preferences (SP). See Paragraph 4.1 for a more detailed description of the table.

the shortest runtime.

On all characteristics in the weak preferences category, scores the Tie Strength model the best. In that solution, the least low tie strengths and the most high tie strength are realized compared to the other models. Consequently, the sum of all realized tie strength is also higher.

The average tie strength in the teams is also the highest for the Tie Strength model. The score of the best team and worst team of this model is similar to the best and worst teams of the other models. The standard deviation of the tie strength in the teams is relatively high for the Tie Strength and the Pref model. So if similar teams are required, the Max Skilled Teams model is preferred. Otherwise, the Tie Strength model is a better option.

The Tie Strength model did not produce one competent team. The other models produced three competent teams and three not-competent teams. The teams of these have a higher number of skills on average, but there is also more difference between the teams. The worst teams Max Skilled Teams and the Pref models are worse than the worst team of the Tie Strength model. So, neither of the models scored high in this category. In the skill coverage category, there is not much difference between the models. However, in the Pref model, there is a larger difference between the best team and the worst team.

Surprisingly, the Tie Strength model scored best on the strong preferences, although the focus of that model is on weak preferences. This is because every positive strong preference is also a weak preference of 5 and every strong negative preference is also a weak preference of 1.

In conclusion, there is no model that performs perfectly in this experiment. The choice is between no competent teams with a higher tie strength or the half of the teams are competent, but the difference between the teams is bigger and the tie strength is a bit lower compared to the Tie Strength model.

4.6 Experiment 6 - 60 candidates - team size 5

In this experiment, we compare one of our models to a theoretical optimum and a solution made by a human. For the Pref model the parameters are set as follows: The minimum teamsize is set to 5 and the maximum teamsize is set to 6. The threshold vector is [5,4,4,4,4,4] and at least 5 skills are required to classify a team as competent. To analyze the human solution results the same criteria are set. Since the model did not solve the problem to the optimal solution within two hours, we ran the model three times. So the results are the average of three solutions. The results are visible in table 6. In this section, also a detailed description of the results of experiment 6 is given. A description of the metrics used in Table 6 can be found in Chapter 3.6.

The Pref model has a much higher TQM (defined in chapter 3.6.1) than the human solution. The gap is close to zero, which means the difference between the lower and upperbound is not big, so the solution is close to the optimal solution.

In the weak preference category, the Pref model realized the least low weak preferences (the tie strengths of 1 and 2) and the most high weak preferences (the tie strengths of 4 and 5). As a result, the sum of the weak preferences is also the highest for the Pref model. This is also visible in the average of the weak preferences in the teams. The best teams produced by the model and the human have the same average tie strength between the members of the team. However, the worst team of the model has a higher average tie strength. The standard deviation for the model is higher.

In the skill category, the Pref model also scores higher. Almost all teams produced by the Pref model are competent (11 of the 12 teams). In the human solution 7 of the teams are competent. The average number of skills in the teams is higher for the Pref model on average, just as the number of teams in the worst team.

A bigger difference is visible in the skill coverage category. The skill coverage (of the teams that are competent) is higher for the human solution. The best team in the human solution scores a lot higher than the best team in the models' solution, while the worst-produced teams score similarly. This means there is a higher difference between the teams in the

	Model	Pref	Human Solution	Total
Global	TQM	**1.67**	1.25	-
	Gap	**0.06**	-	-
	Runtime	02:00:00	-	-
Weak preferences	Sum WP	**1146.0**	1039.0	11015
	Realized 1	**2.0**	4.0	111
	Realized 2	**2.0**	5.0	94
	Realized 3	166.0	190.0	3090
	Realized 4	**8.0**	5.0	79
	Realized 5	**122.0**	87.0	226
	Average WP	**3.82**	3.57	-
	Best team	4.6	4.6	-
	Worst team	**3.2**	2.92	-
	Std	0.54	**0.51**	-
Skills	#skilled teams	**11.0**	7.0	12
	Average (#skills)	**4.92**	4.75	-
	Best team (#skills)	5.0	6.0	-
	Worst team (#skills)	**4.0**	3.0	-
	Average coverage	30.42	**37.42**	-
	Best team	38.0	63.0	-
	Worst team	**21.0**	20.0	-
Strong preferences	Realized positive	**94.0**	75.0	134
	Realized negative	**2.0**	4.0	111
	Realized pos - neg	**92.0**	71.0	-

Table 6: In this table the results of the Pref model and the human solution are compared. The last column contains the total number of occurrences in the data. The characteristics are listed in the second column and are divided into the categories global, weak preferences (WP), skills, and strong preferences (SP). If a value is bold, this means the model outperforms the models without a bold value on this characteristic. See below for a more detailed description of the results.

human solution. If the skills were more evenly divided over the teams, more teams would be competent.

In the strong preferences, the model scores higher on the realized positive preferences and lower on the realized negative preferences. So, the Pref model scored best on the strong preferences.

In conclusion, the Pref model scored best on the weak preferences, the number of skills, and the strong preferences. The human solution has a higher skill coverage in the competent teams on average. However, this is at the expense of the number of competent teams. For more balanced skilled teams with a higher tie strength, the Pref model is preferred.

5 Discussion

In this section, the different experiments are compared and an interpretation of the results is given. We discuss if our findings can be generalized and compare our results with our expectations and other studies on this topic. Lastly, the limitation of this research is discussed as well as possibilities for future research.

In the first experiment, the Skilled Teams and the Pref model performed similarly. Surprisingly, the Tie Strength Model performs only slightly less than the Skilled Teams and Pref Model. It has the same number of competent teams, even though the Tie Strength Model does not opt for competent teams. This could be because the criteria in this experiment with only four required skills were too lenient, so almost every team that is randomly selected would be a competent team. However, in later results teams with three skills appeared, so not every team is competent if four skills are required. A second reason could be that skilled candidates get a higher rating in tie strength. Because the tie strength is relatively evenly spread in the solution of the Tie Strength model, the skills are also evenly spread.

Another remarkable result is the similarity of the Skilled Teams and Pref model in experiment 1. The Skilled Teams model scores as high as the Pref model even though the model does not focus on strong preferences. A reason for this could be that every positive strong preference is a weak preference of 5 and every negative strong preference is also a weak preference of 1. In this way, when maximizing the weak preferences, the Skilled Teams model also maximizes the strong preferences. The same effect is also visible in experiments 3, 4, and 5, where the Max Skilled Teams model and the Pref model score similar on the strong preferences. However, in the second experiment, where the requirements for skills are still limited, the Pref model scores higher on the strong preferences, as we would expect.

In conclusion, there is a trade-off between the skills and the weak/strong preferences. If the skills requirements are lenient, the Tie Strength model scores similar to the (Max) Skilled Teams on the skills and performs worse on the strong preferences compared to the Pref model. As the skill requirements get more strict, the Tie Strength model can not produce competent teams anymore and the other models get less room to optimize the weak preferences or strong preferences. In this case, the Tie Strength model scores higher on weak preferences and on strong preferences but underperforms on skills.

In some experiments, especially the experiments with 60 candidates (experiments 2 and 4) there is some difference between the best and worst team weak preference average. The model maximizes the tie strength between every two teammates, but there are not any constraints on which team the high tie strengths should be realized. This allows the model to create teams with diverging weak preference averages. A solution could be to add some constraints. For example, the weak preference average of a team should be higher than a parameter set beforehand. However, the problem of setting the right parameter should be solved in this case.

The Skilled Teams model gives solutions with only competent teams. If it is not possible to make all teams competent, the model does not give a solution. The Max Skilled Teams model (and the Pref model) give a solution in this case, where not every team is competent.

A consequence is that the worst team produced by the Max Skilled Teams model, which is a non-competent team, has a relatively low number of skills, compared to the competent teams, as shown in experiment 5, where the best team has all skills and the worst team only has 3. In experiment 4 the worst team of the Max Skilled Teams model has even fewer skills. To solve this the number of required skills should be lowered or the threshold for which level classifies as a skill.

In case the Decision Maker(DM) is only partly satisfied with the solution, the DM could run the model again with changed strong preferences. The DM can alter the strong preferences and set positive strong preferences for the made matches that should stay and negative preferences for the dissatisfying matches. In this way, the approach is still hybrid, but with an extra top-down touch.

In the study of (Vinella et al., 2022), a top-down, hybrid, and two bottom-up approaches are compared. The best team produced by one of the bottom-up algorithms scores highest in the teamwork quality comparison (compared to the best teams of the other algorithms), while the worst team of the bottom-up algorithm scores lowest in the teamwork quality comparison. In this thesis, the most bottom-up model has a similar result. In the experiments where the Tie Strength model produced partly competent and partly non-competent teams (experiments 2 and 4), the best team scored highest on the WP average and number of skills (compared to the best teams of the other models), in contrast to the worst teams which scored lowest on the WP average and lowest on the number of skills (compared to the worst teams of the other models).

In experiment 6, one of the models is compared to a teacher's heuristic solution. The Pref model created a solution with a higher score on the number of skills and the weak and strong preferences. From this, it would be hasty to conclude the model works better than a human in general since we have only one human solution at our disposal, but it can serve as a baseline.

The models presented in this thesis were able to solve the k,v-Partition TFP problem for 25 candidates. For 60 candidates, no optimal solution was reached within two hours. Although the not-optimal solutions scored decent on the team quality measure(TQM) with a low gap, there is no proof that the models are suitable for a larger set of candidates. In that case, other heuristic methods can explored like column generation.

Another research topic could be to divide the higher tie strengths and the skills better between the teams. In this way, the scores of the worst teams would be higher and the teams would be more similar. This is a priority if having no bad teams is more important than having excellent teams.

6 Conclusion

In this thesis, the k,v-Partition TFP is examined with a hybrid approach, so a pool of candidates is divided into teams using partly a top-down and partly bottom-up approach, where the focus is on bottom-up to give the candidates more worker agency.

An overview is given of the literature on TFP, where studies are categorized based on the formation of a single team versus multiple teams, using a single task or matching the teams to multiple tasks, placing a part of the candidates in teams or all candidates in teams or all candidates in multiple teams. This overview shows which problems are examined extensively and which are not.

In this thesis, four models are presented to solve the *k,v*-Partition TFP, a model that forms teams maximizing the tie strength between the candidates, a model that creates competent teams if possible (in addition to maximizing the tie strength), a model which maximizes the number of competent teams and a model that also takes strong preferences between the candidates into consideration.

For a group of 25 candidates, the models managed to give an optimal solution. Depending on the skills requirements, the models could find solutions for the problem with a pool of 60 candidates close to optimal. If the Decision Maker(DM) is partly satisfied with the given solution of a model, the DM can create or alter the strong preferences and run the model again for an alternative solution. When solving the *k,v*-Partition TFP for a big group of candidates, a heuristic method is recommended.

References

Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., & Leonardi, S. (2010). Power in unity: Forming teams in large-scale community systems. *Proceedings of the 19th ACM international conference on Information and knowledge management*, 599–608.

Aranzabal, A., Epelde, E., & Artetxe, M. (2022). Team formation on the basis of belbin's roles to enhance students' performance in project based learning. *Education for Chemical Engineers*, *38*, 22–37.

Bacon, D. R., Stewart, K. A., & Anderson, E. S. (2001). Methods of assigning players to teams: A review and novel approach. *Simulation & Gaming*, *32*(1), 6–17.

Bhowmik, A., Borkar, V., Garg, D., & Pallan, M. (2014). Submodularity in team formation problem. *Proceedings of the 2014 SIAM international conference on data mining*, 893–901.

Borrego, M., Karlin, J., McNair, L. D., & Beddoes, K. (2013). Team effectiveness theory from industrial and organizational psychology applied to engineering student project teams: A research review. *Journal of Engineering Education*, *102*(4), 472–512.

Chapman, K. J., Meuter, M., Toy, D., & Wright, L. (2006). Can't we pick our own groups? the influence of group selection method on group dynamics and outcomes. *Journal of Management Education*, *30*(4), 557–569.

Daş, G. S., Altınkaynak, B., Göçken, T., & Türker, A. K. (2022). A set partitioning based goal programming model for the team formation problem. *International Transactions in Operational Research*, *29*(1), 301–322.

Fathian, M., Saei-Shahi, M., & Makui, A. (2017). A new optimization model for reliable team formation problem considering experts' collaboration network. *IEEE Transactions on Engineering management*, *64*(4), 586–593.

Graham, R. L., & Hell, P. (1985). On the history of the minimum spanning tree problem. *Annals of the History of Computing*, *7*(1), 43–57.

Gutiérrez, J. H., Astudillo, C. A., Ballesteros-Pérez, P., Mora-Melià, D., & Candia-Véjar, A. (2016). The multiple team formation problem using sociometry. *Computers & Operations Research*, *75*, 150–162.

Jiang, J., An, B., Jiang, Y., Zhang, C., Bu, Z., & Cao, J. (2019). Group-oriented task allocation for crowdsourcing in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(7), 4417–4432.

Kargar, M., & An, A. (2011). 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*, 985–994.

Kou, Y., Shen, D., Snell, Q., Li, D., Nie, T., Yu, G., & Ma, S. (2020). Efficient team formation in social networks based on constrained pattern graph. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 889–900.

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

Li, L., Yan, M., Tao, Z., Chen, H., & Wu, X. (2022). Semi-supervised graph pattern matching and rematching for expert community location. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Lykourentzou, I., Kraut, R. E., & Dow, S. P. (2017). Team dating leads to better online ad hoc collaborations. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2330–2343.

Lykourentzou, I., Liapis, A., Papastathis, C., Papangelis, K., & Vassilakis, C. (2020). Exploring self-organisation in crowd teams. *Conference on e-Business, e-Services and e-Society*, 164–175.

Majumder, A., Datta, S., & Naidu, K. (2012). Capacitated team formation problem on social networks. *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, 1005–1013.

Martello, S., & Toth, P. (1987). Algorithms for knapsack problems. *North-Holland Mathematics Studies*, *132*, 213–257.

Michaelsen, L. K., Davidson, N., & Major, C. H. (2014). Team-based learning practices and principles in comparison with cooperative learning and problem-based learning. *Journal on Excellence in College Teaching*, *25*.

Oakley, B. A., Hanna, D. M., Kuzmyn, Z., & Felder, R. M. (2007). Best practices involving teamwork in the classroom: Results from a survey of 6435 engineering student respondents. *IEEE Transactions on Education*, *50*(3), 266–272.

Rahman, H., Roy, S. B., Thirumuruganathan, S., Amer-Yahia, S., & Das, G. (2015). Task assignment optimization in collaborative crowdsourcing. *2015 IEEE International Conference on Data Mining*, 949–954.

Salehi, N., & Bernstein, M. S. (2018). Hive: Collective design through network rotation. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–26.

Samie, M. E., & Rajabzadeh, H. (2023). A realistic criterion for team formation in social network. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *47*(1), 355–367.

Valentine, M. A., Retelny, D., To, A., Rahmati, N., Doshi, T., & Bernstein, M. S. (2017). Flash organizations: Crowdsourcing complex work by structuring crowds as organizations. *Proceedings of the 2017 CHI conference on human factors in computing systems*, 3523–3537.

Vinella, F. L., Hu, J., Lykourentzou, I., & Masthoff, J. (2022). Crowdsourcing team formation with worker-centered modeling. *Frontiers in Artificial Intelligence*, *5*.

Wu, G., Chen, Z., Liu, J., Han, D., & Qiao, B. (2021). Task assignment for social-oriented crowdsourcing. *Frontiers of Computer Science*, *15*(2), 1–11.

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