Team Formation with Mutual Respect

A Thesis

submitted to the designated
by the Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Iro Spyrou

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER SYSTEMS ENGINEERING

WITH SPECIALIZATION
IN DATA SCIENCE AND ENGINEERING

University of Ioannina
School of Engineering
Ioannina 2022

Examining Committee:

- **Panayiotis Tsaparas**, Assoc. Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Evaggelia Pitoura**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Spyridon Kontogiannis**, Assoc. Professor, Department of Computer Science and Engineering, University of Ioannina

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Iro Spyrou, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2022.
Team Formation with Mutual Respect.
Advisor: Panayiotis Tsaparas, Associate Professor.

The Team Formation problem in Social Networks [1], asks for a team of experts that covers the skill requirements of a collaborative task, while having low communication cost, as this is computed over the social network that connects the experts. The communication cost captures the quality of the team, that is, the ability of the experts to work together. Several extensions of this work have been considered, with different team quality measures, or different team design criteria.

In this work, we consider an extension of the Team Formation problem, where team quality is measured as the respect between the team members. Given a directed graph for each skill, which captures the respect relationships between experts, we want to create a team where each skill is assigned an expert and the overall respect that the assigned experts receive from the team members is maximized. The respect maximization problem is NP-hard, and a variety of Greedy heuristics have been proposed for solving it [2]. In our work, we propose an Integer Quadratic Programming (IQP) formulation, and we provide an alternative heuristic algorithm for the respect maximization problem.

We then consider a variation of the aforementioned problem, where respect is *antisymmetric*. This means that if there is positive respect from expert u to expert v for some skill, then there is equal but negative *disrespect* from expert v to expert u. If expert u is assigned to this skill, adding expert v to the team will impact negatively the quality of the team.

We first consider a special case of this problem, where the antisymmetric respect values are derived by a scored ranking of the experts. In this case, we show that the respect maximization problem can be reduced to the maximum weight matching problem, which can be solved optimally (using the Hungarian algorithm), or approximately (using a Greedy algorithm) in polynomial time. Building on this observation, we propose a landmark-based algorithm for the general case that reduces to the ranking case.

We implemented and evaluated our algorithms on real datasets against existing baselines. For the general respect maximization problem, our IQP heuristic produces teams with higher respect, albeit with higher computational cost. For the antisymmetric case, for the ranking case, the Greedy algorithm produces solutions very close to the optimal Hungarian algorithm. For the general case, the landmark heuristics perform comparably with the IQP solution and other Greedy approaches, while having lower computational cost.

Εκτεταμενη Περιληψη

Ηρώ Σπύρου, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2022.
Δημιουργία Ομάδων με Αμοιβαίο Σεβασμό.
Επιβλέπων: Παναγιώτης Τσαπάρας, Αναπληρωτής Καθηγητής.

Η συγκρότηση ομάδων είναι ένα πρόβλημα που αντιμετωπίζεται σε διάφορα περιβάλλοντα (π.χ. εκπαίδευση, εργασία, άθληση, παιχνίδια) για την επίτευξη ενός κοινού στόχου. Είναι όμως σημαντικό τα μέλη μιας ομάδας να μπορούν να συνεργαστούν το καλύτερο δυνατό. Συνεπώς, τίθεται το πρόβλημα της Δημιουργίας Ομάδων σε Κοινωνικά Δίκτυα [1], το οποίο λαμβάνει υπόψη τις κοινωνικές σχέσεις των υποψήφιων μελών κατά την δημιουργία τους. Πιο συγκεκριμένα, δοθέντος ενός κοινωνικού δικτύου εργαζόμενων, το οποίο απεικονίζει τις κοινωνικές σχέσεις τους, των δεξιοτήτων τους και ενός συλλογικού έργου, το οποίο απαιτεί ένα σύνολο δεξιοτήτων για την διεκπεραίωσή του, στόχος είναι η δημιουργία μίας ομάδας εργαζόμενων, τα μέλη της οποίας θα καλύπτουν τις απαιτήσεις δεξιοτήτων του έργου και θα έχουν χαμηλό κόστος επικοινωνίας. Το κόστος επικοινωνίας υπολογίζεται βάσει του κοινωνικού δικτύου και δηλώνει την ποιότητα της ομάδας, δηλαδή την ικανότητα των μελών να συνεργαστούν. Έκτοτε έχουν εξεταστεί πολλές επεκτάσεις του προκειμένου προβλήματος, με διαφορετικές μετρικές της ποιότητας των ομάδων ή διαφορετικά κριτήρια σχεδιασμού των ομάδων.

Στην παρούσα εργασία μελετάμε μία επέκταση του προβλήματος Δημιουργίας Ομάδων, όπου η ποιότητα των ομάδων μετράται ως προς τον σεβασμό μεταξύ των μελών μιας ομάδας. Δοθέντος ενός συλλογικού έργου, για την ολοκλήρωση του οποίου απαιτούνται συγκεκριμένες δεξιότητες, και ενός κατευθυνόμενου γράφου για κάθε απαιτούμενη δεξιότητα, ο οποίος απεικονίζει τις σχέσεις σεβασμού μεταξύ των εργαζόμενων, στόχος μας είναι η δημιουργία μίας ομάδας, όπου σε κάθε

δεξιότητα ανατίθεται ένας εργαζόμενος, μεγιστοποιώντας τον συνολικό σεβασμό που λαμβάνουν οι εργαζόμουν που έχουν ανατεθεί σε κάθε δεξιότητα από τα υπόλοιπα μέλη της ομάδας. Οι διαφορές του προβλήματος μεγιστοποίησης σεβασμού με το γενικό πρόβλημα Δημιουργίας Ομάδων είναι πως το πρόβλημα εξετάζεται ως πρόβλημα ανάθεσης και όχι ως πρόβλημα κάλυψης, δηλαδή απαιτείται ακριβως ένας εργαζόμενος για μία δεξιότητα και ένας εργαζόμενος μπορεί να αναλάβει μόνο μία δεξιότητα του έργου. Εκτός αυτού, το κοινωνικό δίκτυο που χρησιμοποιείται για την εξαγωγή των σχέσεων είναι κατευθυνόμενο, το οποίο σημαίνει ότι οι σχέσεις δεν είναι απαραίτητα αμοιβαιες, και είναι διαφορετικό για κάθε δεξιότητα, δηλώνοντας πως οι σχέσεις σεβασμού εξαρτώνται και από την δεξιότητα την οποία αφορά. Το πρόβλημα της μεγιστοποίησης σεβασμού έχει οριστεί στο [2], όπου αποδεικνύεται ότι η πολυπλοκότητα του είναι NP-Hard και προτείνεται μια ποικιλία ευριστικών αλγορίθμων για την επίλυσή του. Επιπλέον, ορίζουν και επιλύουν μία υποπερίπτωση του προβλήματος μεγιστοποίησης σεβασμού, δοθέντος μιας κατάταξης των εργαζόμενων για κάθε δεξιότητα, αντί ενός κοινωνικού δικτύου. Στην δική μας εργασία προτείνουμε μία διατύπωση Integer Quadratic Programming (IQP) και παρέχουμε έναν εναλλακτικό ευριστικό αλγόριθμο για το πρόβλημα της μεγιστοποίησης σεβασμού.

Έπειτα θεωρούμε μια παραλλαγή το προαναφερθέντος προβλήματος, όπου ο σεβασμός είναι *αντισυμμετρικός*. Αυτό σημαίνει πως εάν υπάρχει θετικός σεβασμός από τον εργαζόμενο u προς τον εργαζόμενο v για κάποια δεξιότητα, τότε υπάρχει ίση αλλά αρνητική *ασέβεια ή έλλειψη σεβασμού* από τον εργαζόμενο v προς τον εργαζόμενο u. Αν ο εργαζόμενος u ανατεθεί σε αυτήν την δεξιότητα, τότε η προσθήκη του v στην ομάδα, σε κάποια άλλη δεξιότητα, θα επηρεάσει αρνητικά την ποιότητα της ομάδας.

Αρχικά θεωρούμε μία ειδική περίπτωση του προβλήματος, όπου οι αντισυμμετρικές τιμές σεβασμού εξάγονται από μία βαθμολογημένη κατάταξη των εργαζόμενων. Σε αυτήν την περίπτωση, δείχνουμε ότι το πρόβλημα της μεγιστοποίησης σεβασμού μπορεί να αναχθεί σε πρόβλημα maximum weight matching, το οποίο μπορεί να λυθεί βέλτιστα (με την χρήση του Hungarian αλγορίθμου), ή προσεγγιστικά (με την χρήση ενός Greedy αλγορίθμου) σε πολυωνυμικό χρόνο. Βάσει αυτής της παρατήρησης προτείνουμε αλγόριθμο με ορόσημα, ύστερα από μελέτη διάφορων τρόπων για την βέλτιστη επιλογή ορόσημων, για την γενική περίπτωση του προβλήματος, το οποίο ανάγεται στην περίπτωση της κατάταξης. Επιπροσθέτως, προτείνουμε παραλλαγές κάποιων ευριστικών αλγορίθμων οι οποίοι έχουν προταθεί στο [2] για την επίλυση του νέου προβλήματος με *αντισυμμετρικό* σεβασμό. Τέλος, εξετάσαμε την εφαρμογή του ευριστικού αλγορίθμου βάσει της διατύπωσης IQP, τον οποίο προτείνουμε για το γενικό πρόβλημα μεγιστοποίησης σεβασμού.

Υλοποιήσαμε και αξιολογήσαμε τους αλγορίθμους μας σε πραγματικά σύνολα δεδομένων έναντι υπαρχόντων μελετών ή αλγορίθμων. Για το γενικό πρόβλημα μεγιστοποίησης σεβασμού, ο ευριστικός αλγόριθμος IQP παράγει ομάδες με υψηλότερο σεβασμό, έχοντας όμως μεγαλύτερο υπολογιστικό κόστος. Για την αντισυμμετρική περίπτωση, για την περίπτωση με την κατάταξη, παρατηρούμε πως ο Greedy αλγόριθμος παράγει λύσεις πολύ κοντά σε αυτές του Hungarian αλγορίθμου. Για την γενική περίπτωση με γράφο, οι ευριστικοί αλγόριθμοι με τα ορόσημα αποδίδουν παρόμοια με την λύση του IQP και των άλλων ευριστικών προσεγγίσεων, έχοντας χαμηλότερο υπολογιστικό κόστος.

Chapter 1

Introduction

Team formation is a problem faced in varying settings for the accomplishment of a common goal. Typically, a good team is one that employs the best experts for the skills required for the task at hand [3, 4]. However teams should not be created based solely on the expertise of the people involved, but also take into account their personal relations (often referred to as "team chemistry"), as both are important to ensure that the team will work efficiently.

The problem of Team Formation in Social Networks was defined in [1] in order to combine the importance of expertise and personal relations of the team members during the creation of a team. More specifically, given a collaborative task, requiring a certain skill-set to be completed, an undirected weighted social network of workers, that captures their social relations, and their skills, the goal is the creation of a team of workers covering the skills required for the task while minimizing the communication cost among team members. The communication cost is calculated on the induced subgraph of the chosen workers and measures the ability of the workers to cooperate effectively. Several extensions of this work have been considered, with different team quality measures [5, 6, 7, 8, 9, 10], or different team design criteria [11, 12, 13, 14].

The Team Formation problem, as defined above, extracts all personal relations over a single undirected graph and assumes tasks that don't require a specific structure for the team. But the reality is, that personal relations aren't always reciprocal and depend on different criteria, while teams cooperate better if each member has specific responsibilities. The work in [2] considered an alternative setting where skills are assigned to workers and there are respect relationships between the workers rather than compatibility relations. They defined the *MaxMutualRespect* problem, where given a task with specific skill requirements and a directed social network capturing the respect relations between the workers for each of the required skills, we want to create a team by assigning each skill to a single worker, such that we maximize the respect the workers receive from the remaining team members with respect to the skill they have been assigned to. Having a graph for every skill shows that while one worker may be highly respected in a certain field, doesn't mean that he enjoys the same amount of respect in a different field. The fact that the edges in these graphs are directed depicts that relationships are not necessarily mutual. They also define the *MaxRankingRespect* problem, a special case of the *MaxMutualRespect* problem, where the respect relations are derived over rankings. The *MaxMutualRespect* problem has been proven to be NP-hard, while the complexity of the *MaxRankingRespect* problem is unresolved.

This thesis extends the work of [2] in two ways. First, we propose an Integer Quadratic Programming (IQP) formulation for the *MaxMutualRespect* problem, providing an alternative heuristic algorithm. Our formulation is general enough to be used for all variants of the problem.

Subsequently, we consider a variation of the *MaxMutualRespect* problem, where respect is *antisymmetric*. This means that if a worker v has respect for worker u in a skill, then worker u will have equal but negative *disrespect* for worker v. If worker u is assigned to the skill, then adding worker v to the team has a negative effect on the team. For example, this could be the case when a v is senior, or more experienced on the skill than worker u. We define *MaxMutualAntisymmetricRespect* to denote this variant of the problem.

We first consider the *MaxRankingAntisymmetricRespect* problem, a special case, where the antisymmetric respect values are derived over a ranking, and show that it can be reduced to the maximum weight matching problem. This problem can now be solved optimally with the use of the Hungarian algorithm, or approximately with the use of a Greedy algorithm in polynomial time. Note that the complexity of the corresponding problem in [2] was left unresolved.

We then consider the general case where the antisymmetric respect values are derived over a general respect graph. Inspired by the work on landmark-based distance estimation (e.g., see [15]), we propose a landmark-based algorithm for this case, and we show that the algorithm reduces to solving the *MaxRankingAntisymmetricRespect* problem. We consider different strategies for selecting landmarks, and we evaluate them experimentally. We also propose variations of some of the heuristic algorithms defined in [2] and examine the application of the IQP heuristic algorithm for this case.

Our proposed algorithms have been implemented and evaluated using real datasets against existing baselines. For the *MaxMutualRespect* problem, the heuristic IQP algorithm assigns teams with higher respect score, while for the *MaxRankingRespect* problem solutions with maximum respect are found, though in both cases with higher computational cost. We thus confirm that our formulation can compute a higher respect score than the current heuristics.

For the asymmetric respect case, for the *MaxRankingAntisymmetricRespect* problem we observe that the Greedy algorithm creates teams very close to those of the Hungarian algorithm, while having smaller cost. For the general *MaxMutualAntisymmetricRespect* problem we observe that our landmark-based algorithm's performance is close to that of the IQP heuristic algorithm's and that of the other heuristics, while having lower computational cost.

1.1 Thesis Contributions

In summary in this thesis we make the following contributions:

- We present a novel Integer Quadratic Programming formulation for the *MaxMutualRespect* problem. We evaluate it experimentally, and we demonstrate that it achieves higher score than existing heuristics, albeit at a higher cost.
- We propose a novel variant of the *MaxMutualRespect* problem where respect is antisymmetric. We show that for the ranking case of our problem we can find the optimal solution in polynomial time.
- We propose a landmark-based algorithm for the general antisymmetric case, which utilizes the algorithms for the ranking case to find a solution. We evaluate

different approaches for selecting the landmarks.

- We evaluate our algorithms on real datasets, and we compare against existing baselines.

1.2 Thesis Roadmap

The outline of this thesis is as follows:

- In Chapter 1 we introduced the problem we study in the thesis.
- In Chapter 2 we present previous work related to the problem we examine
- In Chapter 3 we define our problem, and we propose an IQP heuristic algorithm. We evaluate our algorithm against existing heuristics.
- In Chapter 4 we define the antisymmetric respect problem, and we propose algorithms for the different cases of the problem. We evaluate our algorithms experimentally.
- Chapter 5 contains our conclusions on this work.

Chapter 2

Related Work

The Team Formation problem in Social Networks was first defined in [1], where given a set of workers, a task and an undirected graph depicting the compatibility between the workers, the goal is to find a subset of workers that covers the skills required for the task, while inducing a subgraph with low communication cost. They examine two variations for the communication cost function, one being the diameter of the induced subgraph, the other being the minimum spanning tree on the induced subgraph.

Since then, the Team Formation problem has been studied, examining more variations of the communication cost formulation and introducing new requirements. In [5] and [6] density-based measures are proposed as communication cost functions, while in [7] the computational complexity of different measures is evaluated.

The existence of personnel cost besides communication cost is considered in various works. In [8] and [9] combined cost functions are proposed, while the authors of [10] apply a budget to the personnel cost and strive to create teams that can cover multiple tasks.

Even distribution of the task among the team members, meaning no one is overloaded or singled-out, is studied in [11], [12] and [13]. At the same time [12] and [13] examine online Team Formation, as does [14]. Online Team Formation means that the tasks arrive successively, instead of them all being available from the beginning, and upon each task arrival a team fulfilling the requirements is created.

More variations of the requirements are studied, such as the inclusion of a designated team leader in [16] and [17], the diversity of the team members in [18], and a combination of several design criteria in [19] where a submodular function is proposed.

In [20] the Team Formation problem is studied on signed social networks, where workers can also have negative relationships, making them non-compatible. The goal here is to create compatible teams, covering the skill requirements and minimizing communication cost.

The Team Formation problem is also examined in different setting, such as online games in [21] and [22]. Both works propose different evaluation criteria for the teams, with respect to the setting.

In the aforementioned publications the Team Formation problem is a set-cover problem, though our work focuses on the Team Formation problem as an assignment problem. Previous works studying the Team Formation problem as an assignment problem include [23] and [2]. In [23] the teams created must have a certain structure based on a given template in the form of a graph. This structure ensures hierarchies among the team members. The objective is to assign workers to the roles of the template while minimizing the communication cost along the template edges.

Our work is an extension of the work presented in [2]. In this case a social network in the form of a directed graph is provided for each skill of the task, denoting the respect relationships of the workers for the specified skill. Note that the graphs in this case are directed, meaning that respect relations are not necessarily mutual. Each skill of the given task gets assigned one worker, forming a team, where instead of minimizing communication cost, the objective is to maximize respect score among the workers across the different skills. To solve the problem various heuristic algorithms are proposed. The authors also define a special case of the problem, based on an ordered ranking of the workers instead of a graph, and propose a polynomial algorithm which finds teams of maximum respect if such a team exists, as well as approximation algorithms. In our work we propose an Integer Quadratic Programming formulation to solve this problem and define a variation of it.

Chapter 3

Integer Quadratic Programming

In this Chapter we formally define the Respect Maximization problem that was fist considered in [2]. We then show how the optimization problem can be formulated as an Integer Quadratic Program. We provide experiments comparing our algorithm with those in [2].

3.1 Problem Definition

We now define two variants of the respect maximization problem defined in [2].

We are given directed graph $G^s = (X, E^s)$, for each skill s, where X denotes the set of workers and the graph denotes the respect relationships between the workers. Every directed edge $(x_i, x_j) \in E^s$ denotes that x_j respects x_i for skill s. Our goal is to create teams of workers $F \subseteq X$ where each skill is assigned to one worker and a worker can occupy only one skill. The team F produced should have maximum respect possible.

Specifically, a skill assignment is defined as an injective function $f : S \to X$, where $f(i)$ is the worker assigned to skill $i \in S$. $F = f(S)$ denotes the selected team of experts covering set of skills S. The respect $R_i(f)$ that worker $f(i)$ receives is computed by the number of outgoing edges in graph G^i to the other workers in the assignment and is defined as:

$$R_i(f) = |\{(f(i), u) \in E^i : u \in F, u \neq f(i)\}| \tag{3.1}$$

The total respect for an assignment is given by the sum of the respect values of each of the workers assigned and defined as:

$$R(f) = \sum_{i \in S} R_i(f) \tag{3.2}$$

The *RespectMaximization* problem can now be defined.

Problem 1 (*RespectMaximization*). *Given a set of workers X, a set of skills S and respect graphs $G^i = (X, E^i), \forall i \in S$, find an assignment $f : S \rightarrow X$, that maximizes $R(f)$.*

A natural way to derive respect relationships between the workers for a skill is via a ranking of the workers. The ranking defines a pecking order where those lower in the ranking respect those higher in the ranking.

The ranking case can be easily captured by our general definition. Given a ranking, we can create a respect graph as follows. Let P^i denote the ranking for skill $i \in S$. For every worker, the value $P^i[x]$ is the position of worker x in the ranking of skill i. The lower the value of $P^i[x]$ the higher the worker is in the ranking. A worker in a ranking P^i respects all workers above him in the ranking. The graph G^i produced by P^i places an edge (v, u) for all pairs of nodes where $P^i[u] > P^i[v]$.

We use *MaxRankingRespect* to refer to this special case of *RespectMaximization*. We will consider this problem separately in our algorithms and experiments.

3.2 An Integer Quadratic Programming Formulation

The goal of this section is to formulate algorithms to solve the *RespectMaximization* and *MaxRankingRespect* problems using Quadratic Programming (*QP*) [24]. A *QP* optimizes a quadratic function using equality, inequality and bound constraints. An Integer Quadratic Program (*IQP*) only has discrete variables in the model.

We have n workers X and k skills S. For each skill s we have a graph G^s that denotes the respect relationships between the workers. We assume that a directed edge (x_i, x_j) between two workers denotes that x_j respects x_i (or, x_i commands the respect of x_j).

Our goal is to find an assignment $f : S \to X$ that maximizes the respect of the team. The respect is defined as the sum over all skills, of the outgoing edges from the worker assigned to the skill to the remaining members of the team. Let F be the team of assigned workers. Without loss of generality we assume that $f(j) = x_j$. Let A^j denote the adjacency matrix of graph G^j. The respect for skill j is now computed as:

$$R_j(f) = \sum_{x_i \in F} A^j[x_j, x_i], \tag{3.3}$$

where $A^j[x_j, x_i] = 1$ means that an edge (x_j, x_i) exist in G^j, thus x_i respects x_j for skill j.

The total respect is defined as:

$$R(f) = \sum_{j=1}^{k} R_j(f) \tag{3.4}$$

Let $\mathbf{f}$ denote an n-dimensional binary vector that defines the set F, where $\mathbf{f}_i = 1$ if $x_i \in F$ and zero otherwise. Then we can write:

$$R_j(f) = A^j[x_j, :]\mathbf{f}, \tag{3.5}$$

where $A^j[x_j, :]$ denotes the x_j-row of the matrix A^j. Since $\mathbf{f}$ denotes the position $\forall x_i \in F$, the inner product of Equation 3.5 gives the sum of the existing edges $(x_j, x_i), \forall x_i \in F$ in G^j, as does Equation 3.3.

Also let $\mathbf{f}^j$ be an n-dimensional one-hot vector that defines the assignment of f for skill j, where $\mathbf{f}^j(i) = 1$ when $f(j) = x_i$ and zero everywhere else. Note that:

$$R_j(f) = {\mathbf{f}^j}^T A^j \mathbf{f} = \sum_{i=1}^{k} {\mathbf{f}^j}^T A^j \mathbf{f}^i, \tag{3.6}$$

where the inner product ${\mathbf{f}^j}^T A^j$ gives $A^j[x_j, :]$, and Equation 3.6 follows directly from Equation 3.5.

Therefore:

$$R(f) = \sum_{j=1}^{k} \sum_{i=1}^{k} {\mathbf{f}^j}^T A^j \mathbf{f}^i \tag{3.7}$$

We can write this in a standard quadratic form. We use $\mathbf{x} = [\mathbf{f}^1; \mathbf{f}^2; \cdots; \mathbf{f}^k]$ to denote the $(n \times k)$-dimensional vector that is defined as the stacking of the $\mathbf{f}^j$ vectors. Also we define the $(n \times k) \times (n \times k)$ matrix M as follows:

$$M = \begin{bmatrix} A^1 & A^1 & \cdots & A^1 \\ A^2 & A^2 & \cdots & A^2 \\ \vdots & \vdots & & \vdots \\ A^k & A^k & \cdots & A^k \end{bmatrix} \tag{3.8}$$

We can see that:

$$R(f) = \mathbf{x}^T M \mathbf{x}, \tag{3.9}$$

Note that $\mathbf{x}^T M = [\mathbf{f}^{1^T} A^1 + \mathbf{f}^{2^T} A^2 + \cdots + \mathbf{f}^{k^T} A^k; \cdots; \mathbf{f}^{1^T} A^1 + \mathbf{f}^{2^T} A^2 + \cdots + \mathbf{f}^{k^T} A^k]$, and each section of the vector is equal to the sum of $\mathbf{f}^{j^T} A^j \forall j \in S$. Accordingly, $\mathbf{x}^T M \mathbf{x}$ gives the sum of respect over the complete assignment as defined in 3.7.

We observe that $\mathbf{x}^T M \mathbf{x} = \mathbf{x}^T M^T \mathbf{x}$, as shown below:

$$\begin{aligned} \mathbf{x}^T M \mathbf{x} = & [\mathbf{f}^{1^T} A^1 + \mathbf{f}^{2^T} A^2 + \cdots + \mathbf{f}^{k^T} A^k; \mathbf{f}^{1^T} A^1 + \mathbf{f}^{2^T} A^2 + \cdots + \mathbf{f}^{k^T} A^k; \\ & \cdots; \mathbf{f}^{1^T} A^1 + \mathbf{f}^{2^T} A^2 + \cdots + \mathbf{f}^{k^T} A^k]\mathbf{x} \\ = & [A^1[x_1, :] + A^2[x_2, :] + \cdots + A^k[x_k, :]; A^1[x_1, :] + A^2[x_2, :] + \\ & \cdots + A^k[x_k, :]; \cdots; A^1[x_1, :] + A^2[x_2, :] + \cdots + A^k[x_k, :]]\mathbf{x} \\ = & [A^1[x_1, x_1] + A^2[x_2, x_1] + \cdots + A^k[x_k, x_1] + A^1[x_1, x_2] + A^2[x_2, x_2] + \\ & \cdots + A^k[x_k, x_2] + \cdots + A^1[x_1, x_k] + A^2[x_2, x_k] + \cdots + A^k[x_k, x_k]], \end{aligned} \tag{3.10}$$

and:

$$\begin{aligned} \mathbf{x}^T M^T \mathbf{x} = & [\mathbf{f}^{1^T} A^{1^T} + \mathbf{f}^{2^T} A^{1^T} + \cdots + \mathbf{f}^{k^T} A^{1^T}; \mathbf{f}^{1^T} A^{2^T} + \mathbf{f}^{2^T} A^{2^T} + \cdots + \mathbf{f}^{k^T} A^{2^T}; \\ & \cdots; \mathbf{f}^{1^T} A^{k^T} + \mathbf{f}^{2^T} A^{k^T} + \cdots + \mathbf{f}^{k^T} A^{k^T}]\mathbf{x} \\ = & [A^{1^T}[x_1, :] + A^{1^T}[x_2, :] + \cdots + A^{1^T}[x_k, :]; A^{2^T}[x_1, :] + A^2 T[x_2, :] + \cdots + A^2 T[x_k, :]; \\ & \cdots; A^{k^T}[x_1, :] + A^{k^T}[x_2, :] + \cdots + A^{k^T}[x_k, :]]\mathbf{x} \\ = & [A^{1^T}[x_1, x_1] + A^{1^T}[x_2, x_1] + \cdots + A^{1^T}[x_k, x_1] + A^{2^T}[x_1, x_2] + A^{2^T}[x_2, x_2] + \\ & \cdots + A^{2^T}[x_k, x_2] + \cdots + A^{k^T}[x_1, x_k] + A^{k^T}[x_2, x_k] + \cdots + A^{k^T}[x_k, x_k]], \end{aligned} \tag{3.11}$$

where we can see that $\mathbf{x}^T M \mathbf{x}$ and $\mathbf{x}^T M^T \mathbf{x}$ are sums over the same values, as $A^j[x_j, x_i] = A^{j^T}[x_i, x_j]$.

We can now define a symmetric matrix P as follows:

$$P = M + M^T = \begin{bmatrix} A^1 + (A^1)^T & A^1 + (A^2)^T & \cdots & A^1 + (A^k)^T \\ A^2 + (A^1)^T & A^2 + (A^2)^T & \cdots & A^2 + (A^k)^T \\ \vdots & \vdots & & \vdots \\ A^k + (A^1)^T & A^k + (A^2)^T & \cdots & A^k + (A^k)^T \end{bmatrix} \tag{3.12}$$

Since $\mathbf{x}^T M \mathbf{x} = \mathbf{x}^T M^T \mathbf{x}$, we know that $\mathbf{x}^T P \mathbf{x} = 2R(f)$, and can now define our integer quadratic program satisfying the symmetry requirements.

We thus have the following integer quadratic program:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^T P \mathbf{x} \\ \text{subject to} \quad & \mathbf{x}_i \in \{0,1\},\ i = 1, \ldots, n \times k \\ & \sum_{i=1}^{n} \mathbf{x}_{i+j} = 1,\ j = 1, \ldots, k \text{ (Every skill is assigned a worker)} \\ & \sum_{j=1}^{k} \mathbf{x}_{i+j} \leq 1,\ i = 1, \ldots, n \text{ (Every worker is assigned to at most one skill)} \end{aligned} \tag{3.13}$$

3.3 Experiments

3.3.1 Datasets

Dataset for RespectMaximization problem

As in [2], we study the *RespectMaximization* problem on real data generated from academic citation networks. In this setting the workers are scientists, and the skills are scientific fields. The respect graph for each scientific field is based on citations. An edge $(x_i, y_i) \in E^i$ means that author x_i has published a paper in field i and author y_i has a publication citing that paper.

Specifically, the following scientific fields on Compute Science are considered: Artificial Intelligence (AI), Neural Networks (NN), Natural Language Processing (NLP), Robotics, Data Mining (DM), Algorithms, Data Bases (DB), Theory, Signal Processing (SP), Computer Networking (CN), Information Retrieval (IR), Wireless Networks and Mobile Computing (Wireless), Software Engineering (SE), High-Performance Computing (HPC), Distributed and Parallel Computing (DPC) and Operating Systems (OS).

With the use of publicly available resources [1] the top-tier conferences for each field are found. Then the DBLP dataset[2] is used to extract the set of publications and authors belonging to these conferences, and the citation networks for the different fields are created. For noise-reduction purposes, all self-loops were removed from the graphs, and authors with less than 5 incoming or outgoing edges were iteratively pruned.

The following six teams are considered:

1. Team 1 is an AI & Applications team requiring scientists from the fields AI, NN, NLP and Robotics
2. Team 2 is a Data & Analysis team requiring scientists from the fields DM, Algorithms, DB and Theory
3. Team 3 requires scientists from all fields in Teams 1 and 2
4. Team 4 is a Systems team requiring scientists from the fields SE, HPC, DPC and OS
5. Team 5 is a Networks team requiring scientists from the fields SP, CN, IR and Wireless
6. Team 6 requires scientists for all fields in Teams 4 and 5

Dataset for MaxRankingRespect problem

The *MaxRankingRespect* problem is studied using the NBA dataset[3], as in [2], which contains individual basketball player statistics for different NBA seasons, for a range of basic statistics such as points, assists rebounds etc., to more advanced performance metrics such as value over replacement. The same data for the seasons 2010 - 2017 is used, as are the same 11 performance metrics that they consider important in assembling a basketball team: STL, AST, FT, BLK, FG, TRB, 2P, 3P, DBPM, OBPM and VORP, whose description can be read in[4]. In our setting these performance metrics correspond to skills, while the players correspond to workers. The set of players is pruned so as to keep the ones that have payed in at least one third of the games of the season, and have played at least 15 minutes per game. A ranking over these

[1] https://dl.acm.org/ccs/ccs_flat.cfm
[2] https://www.aminer.cn/data/?nav=openData#Citation
[3] https://www.kaggle.com/datasets/drgilermo/nba-players-stats
[4] https://www.basketball-reference.com/about/glossary.html

performance metrics is created by sorting the players in decreasing order of the metric value.

In this case we consider every season 2010 - 2017 as a team, each of them having as skills the 11 performance metrics mentioned above.

3.3.2 Algorithms

As our problems have been previously defined and solved in [2], we will evaluate our IQP formulation against the best performing algorithms presented there. For the *RespectMaximization* problem we will compare with the *RandGreedy* algorithm, while for the *MaxRankingRespect* problem will compare to the *AllCandidates* algorithm. We describe these two algorithms in detail below.

RandGreedy

The *RandGreedy* algorithm computes an initial score value for each skill-worker pair as:

$$s(i,x) = deg^{+}_{G^i}(x) + \frac{1}{k-1} \sum_{j \in S : j \neq i} deg^{-}_{G^j}(x), \tag{3.14}$$

where $deg^{+}_{G^i}(x)$ denotes the outgoing edges of worker x in graph G^i and $deg^{-}_{G^i}(x)$ denotes his incoming edges. The intuition is that high out-degree $deg^{+}_{G^i}(x)$ in graph G^i means that worker x is highly respected for skill i, while high average in-degree $deg^{-}_{G^j}(x)$ for the remaining skills means that worker x has on average high respect for the other workers in the other skills.

It then selects a skill uniformly at random and makes the assignment of the skill-worker pair with the highest score value. *RandGreedy* proceeds in an iterative manner, computing an updated score value for each skill-worker pair given the partial assignment F as follows:

$$\begin{aligned} s_F(i,x) = deg^{+}_{G^i[F \cup \{x\}]}(x) + |\{(x, f(j)) \in E_j : f(j) \neq \emptyset\}| \\ + \frac{1}{k-|F|} \sum_{j : f(j) = \emptyset} deg^{-}_{G^j[V \setminus F]}(x), \end{aligned} \tag{3.15}$$

where $f(j) = \emptyset$ denotes an unassigned skill and $G[F]$ denotes the induced subgraph of the set $F \subseteq V$. A skill-worker pair (i,x) receives high score if worker x is highly respected by the assigned workers in F for skill i, worker x has high respect for the workers assigned to other skills, and has high average respect for the unassigned workers in the unassigned skills. The terms in the above values are normalized to be in the same scale. At each iteration, a skill is selected uniformly at random, and the skill-worker pair with the highest score value is assigned. This iterative selection step is repeated until all skills have been assigned a worker. The *RandGreedy* algorithm is repeated $t = 50$ times and the assignment with the highest score is reported.

AllCandidates

The *AllCandidates* algorithm is an algorithm for the *MaxRankingRespect* problem. Given the set of rankings for each skill, it exhaustively considers each possible skill-worker pair $(i, x) \in S \times X$ as a first assignment. For each of the first assignments, it then proceeds by selecting a skill uniformly at random and assigning the highest ranked worker that has not been assigned. The assignment with the highest score is reported.

If a solution with maximum respect score exists, the *AllCandidates* algorithm has been shown in [2] to always find it. Consequently, we examine if our *IQP_MaxRespect* algorithm will also be able to find such a solution.

IQP_MaxRespect

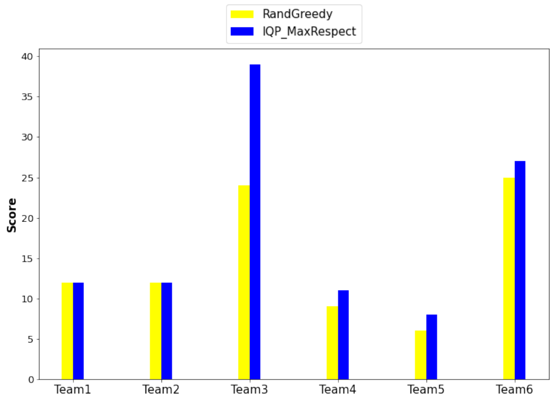
To bring our *IQP* formulation to algorithm form we used the CVXPY open source Python-embedded modeling language, combined with the GUROBI solver. We refer to this algorithm as the *IQP_MaxRespect* algorithm.

3.3.3 Results

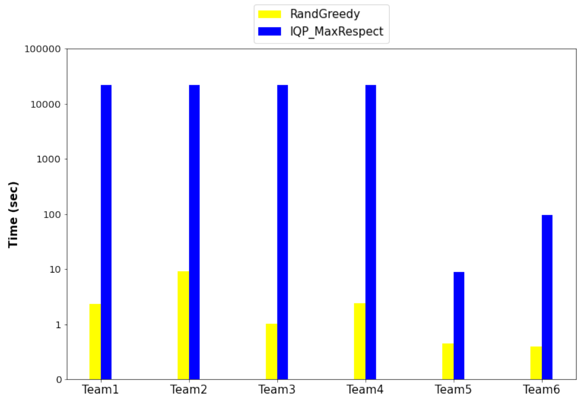
Results for RespectMaximization problem

Figure 3.1a shows the score of the *IQP_MaxRespect* algorithm with the DBLP dataset next to the score achieved by the *RandGreedy* algorithm. We can see that in most cases the *IQP_MaxRespect* algorithm outperforms the *RandGreedy* algorithm, and in the other cases it achieves the same score. We can therefore conclude that the *IQP_MaxRespect* algorithm does offer an advantage.

In Figure 3.1b the execution times of the *IQP_MaxRespect* algorithm and the *RandGreedy* algorithm are shown side by side. We observe that whilst *IQP_MaxRespect* performs greatly in regards to the score, the efficiency is substantially worse than the *RandGreedy* algorithm in regards of the execution time. A time limit of 6 hours had to



(a) Respect score



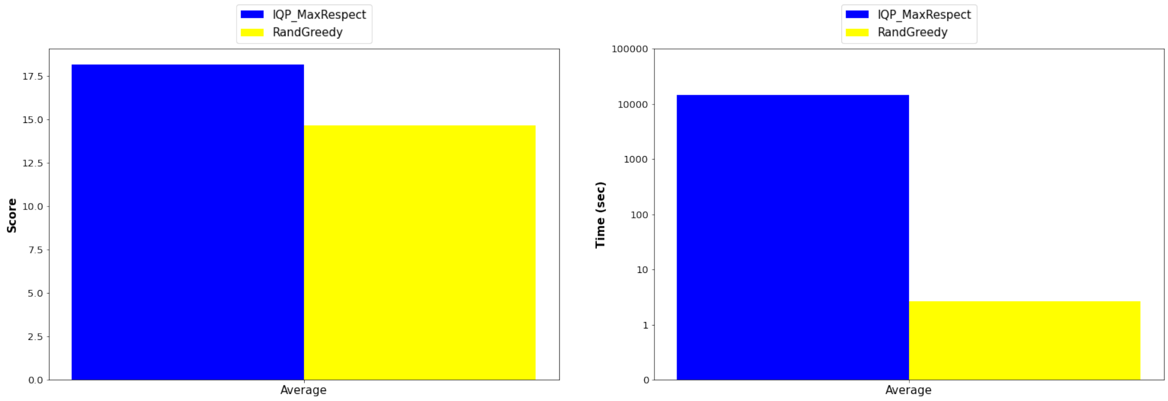
(b) Runtime analysis

Figure 3.1: Respect score and runtime analysis comparison of IQP_MaxRespect and RandGreedy algorithms.

be applied in order to obtain results for Teams 1 - 4 for the *IQP_MaxRespect* algorithm, otherwise the algorithm terminated unexpectedly without returning results.

Figures 3.2a and 3.2b show the average respect score and runtime values over all teams for each algorithm. We can see that our *IQP_MaxRespect* algorithm performs better overall with respect to the score, while with respect to the runtime the *RandGreedy* algorithm is much more efficient.

The workers selected for each team by the *IQP_MaxRespect* algorithm with the DBLP dataset can be seen in Table 3.1 in comparison to the experts selected by the *RandGreedy* algorithm. Rows 2 and 10 denoted as Top contain the scientists with the highest number of citations in each field, presented for calibration. We observe that for Teams 1, 2, 4 and 5, in most cases *IQP_MaxRespect* assigns different experts than *RandGreedy* does, and also that it never assigns the most cited author in any field. An interesting case is Team 2 where the *IQP_MaxRespect* algorithm produces a team that seems intuitively more appropriate than that of *RandGreedy*. However, for Team 6 the assignments of *IQP_MaxRespect* and *RandGreedy* have most experts in common, but assigned to different fields, the same goes for Team 3 where the two algorithms have some assignments in common and a few of the same experts assigned to different fields.



(a) Average respect score (b) Average runtime analysis

Figure 3.2: Average respect score and runtime analysis comparison of IQP_MaxRespect and RandGreedy algorithms.

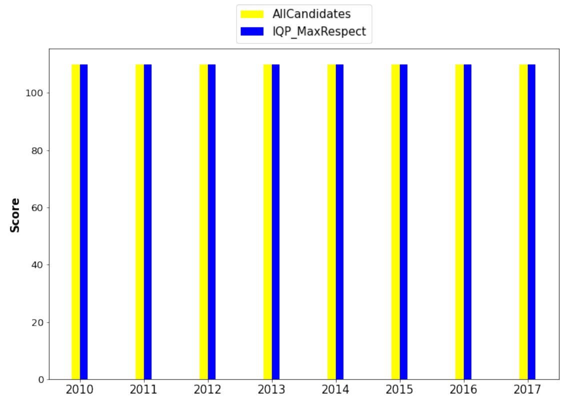
Table 3.1: Teams produced by the IQP_MaxRespect algorithm with the DBLP dataset.

		AI	**NN**	**NLP**	**Robotics**	**DM**	**Algorithms**	**DB**	**Theory**
Top		J.Lafferty	G.Hinton	E.Hovy	V.Kumar	C.Aggarwal	A.Goldberg	R.Agrawal	M.Szegedy
Team1	**RandGreedy**	W.Burgard	A.Ng	J.Pineau	S.Thrun	-	-	-	-
	IQP_MaxRespect	H.Lee	A.Ng	C.Manning	D.Fox	-	-	-	-
Team2	**RandGreedy**	-	-	-	-	D.Srivastava	S.Muthukrishnan	M.Hadjieleftheriou	N.Koudas
	IQP_MaxRespect	-	-	-	-	P.S.Yu	S.Muthukrishnan	A.Gionis	P.Indyk
Team3	**RandGreedy**	M.Jordan	B.Mirzasoleiman	Q.Yang	W.Wang	A.Krause	A.Badanidiyuru	C.Guestrin	A.Karbasi
	IQP_MaxRespect	A.Krause	B.Mirzasoleiman	A.Singla	J.Vondrk	C.Guestrin	A.Badanidiyuru	T.Joachims	A.Karbasi
		SE	**HPC**	**DPC**	**OS**	**SP**	**CN**	**IR**	**Wireless**
Top		G.Rothermel	I.Foster	L.Ni	M.Kaashoek	G.Giannakis	D.Towsley	C.Buckley	J.Polastre
Team4	**RandGreedy**	R.Gupta	D.Panda	Q.Gao	Y.Zhou	-	-	-	-
	IQP_MaxRespect	R.Iyer	W.Kramer	Z.Kalbarczyk	A.Arpaci-Dusseau	-	-	-	-
Team5	**RandGreedy**	-	-	-	-	C.Zhang	R.Zhang	B.Li	Q.Zhang
	IQP_MaxRespect	-	-	-	-	R.Zhang	Y.Chen	J.Zhang	Q.Zhang
Team6	**RandGreedy**	B.Li	J.Wu	Y.Liu	Z.Yang	J.Han	M.Li	J.Cao	X.Li
	IQP_MaxRespect	Z.Yang	J.Wu	M.Li	J.Liu	Z.Li	X.Li	J.Han	Y.Liu

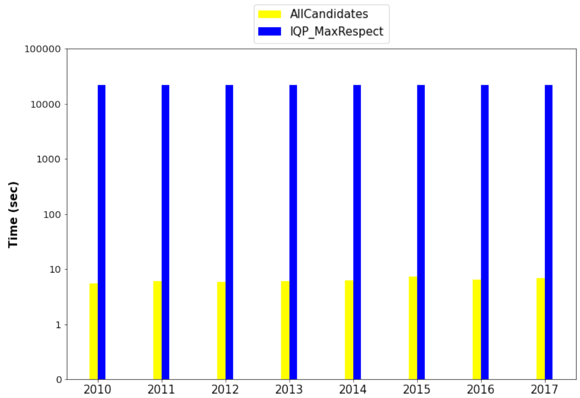
Results for MaxRankingRespect problem

In Figure 3.3a the respect score achieved by the *IQP_MaxRespect* algorithm with the NBA dataset is compared to the score achieved by the *AllCandidates* algorithm. It is easily observed that the *IQP_MaxRespect* algorithm produces teams with the same respect score as the *AllCandidates* algorithm, which is the maximum score possible, despite assigning different workers.

Figure 3.3b shows the comparison of the execution time of the *IQP_MaxRespect* algorithm and the *AllCandidates* algorithm for the NBA dataset. We can see that the *IQP_MaxRespect* algorithm has a very long execution time compared to the AllCandidates algorithm. In this case, too, a time limit of 6 hours had to be applied in order



(a) Respect score



(b) Runtime analysis

Figure 3.3: Respect score and runtime analysis comparison of IQP_MaxRespect and AllCandidates algorithms.

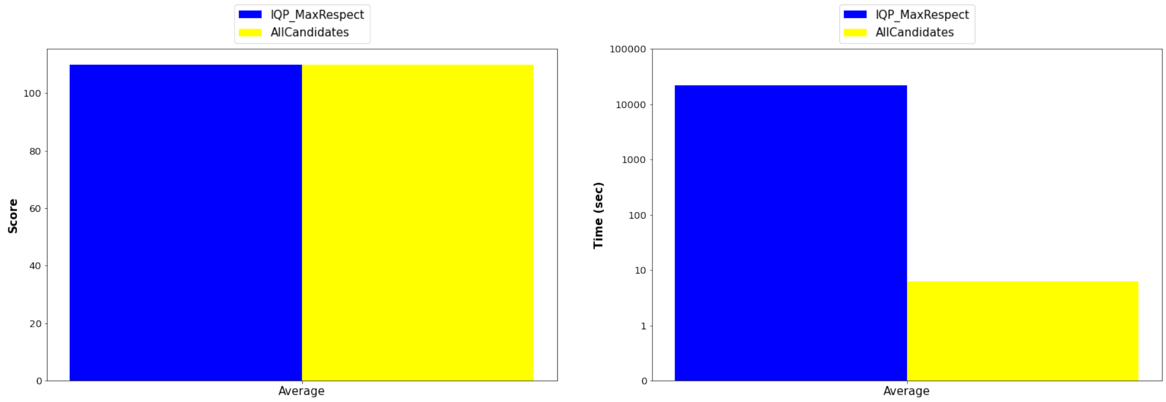
Table 3.2: Teams produced by the IQP_MaxRespect algorithm with the NBA dataset for seasons 2010 - 2013.

	2010		2011		2012		2013	
	IQP_MaxRespect	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**
STL	C.J.Watson	E.Watson	O.J.Mayo	D.Fisher	C.Delfino	M.Conley	C.Brewer	T.Allen
AST	J.Calderon	J.Calderon	E.Watson	J.Calderon	B.Udrih	S.Nash	J.Tinsley	E.Turner
FT	C.Landry	K.Lowry	M.Williams	R.Sessions	J.Crawford	K.Bryant	T.Hansbrough	D.DeRozan
BLK	C.Andersen	C.Andersen	W.Chandler	J.McGee	K.Seraphin	S.Ibaka	B.Biyombo	R.Hibbert
FG	O.J.Mayo	K.Martin	C.Villanueva	M.Beasley	T.Prince	R.Westbrook	A.Afflalo	J.Smith
TRB	E.Okafor	J.Noah	K.Brown	K.Humphries	U.Haslem	D.Howard	R.Evans	O.Asik
2P	A.Bargnani	N.Krstic	S.Young	D.DeRozan	G.Henderson	B.Griffin	J.Thompson	C.Boozer
3P	R.Butler	Q.Richardson	G.Neal	M.Belinelli	R.Foye	J.Terry	C.Butler	B.Gordon
DBPM	T.Thomas	M.Camby	A.McDyess	K.Thomas	B.Wallace	M.Camby	A.Bogut	A.Bogut
OBPM	G.Arenas	G.Arenas	B.Miller	L.Williams	J.Bayless	D.Wade	W.Chandler	A.J.Price
VORP	A.Kirilenko	B.Wallace	M.Dunleavy	S.Jackson	R.Allen	J.Noah	J.Butler	T.Duncan

to obtain an assignment for every season with the *IQP_MaxRespect* algorithm.

In Figures 3.4a and 3.4b the average respect score and runtime over all teams is shown. We can see that with respect to the score the difference between the algorithms is minimal, while with respect to the runtime the difference is substantial.

Tables 3.2 and 3.3 show the teams assigned by the *IQP_MaxRespect* algorithm for the NBA dataset next to the teams produced by the *AllCandidates* algorithm. We observe that the teams assigned by each algorithm differ vastly for every season.



(a) Average respect score (b) Average runtime analysis

Figure 3.4: Average respect score and runtime analysis comparison of IQP_MaxRespect and AllCandidates algorithms.

Table 3.3: Teams produced by the IQP_MaxRespect algorithm with the NBA dataset for seasons 2014 - 2017.

	2014		**2015**		**2016**		**2017**	
	IQP_MaxRespect	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**	**IQP_MaxRespect**	**AllCandidates**
STL	K.Caldwell-Pope	R.Rubio	B.Knight	T.Ariza	C.Brewer	R.Rubio	K.Caldwell-Pope	P.Beverley
AST	A.Rivers	J.Wall	D.Schroder	J.Wall	S.Mack	R.Rondo	D.Collison	R.Rondo
FT	D.Williams	J.Harden	J.Green	D.Cousins	A.Goodwin	D.DeRozan	D.Gallinari	J.Embiid
BLK	R.Kelly	S.Ibaka	D.Cunningham	R.Gobert	D.Cunningham	H.Whiteside	K.Porzingis	K.O'Quinn)
FG	O.J.Mayo	C.Anthony	G.Green	L.Aldridge	G.Henderson	C.J.McCollum	C.J.McCollum	E.Turner
TRB	Z.Pachulia	D.Jordan	L.Scola	D.Jordan	B.Portis	A.Drummond	C.Capela	B.Biyombo
2P	C.Kaman	B.Griffin	M.Speights	N.Vucevic	T.J.Warren	K.A.Towns	H.Barnes	K.Faried
3P	M.Teletovic	K.Thompson	T.Ross	K.Thompson	B.McLemore	J.R.Smith	K.Thompson	W.Ellington
DBPM	A.Kirilenko	A.Bogut	K.Bazemore	A.Bogut	L.Nance	A.Bogut	M.Muscala	L.Nogueira
OBPM	N.Robinson	C.Paul	J.Clarkson	K.Lowry	C.Landry	C.Anthony	B.Beal	J.Lin
VORP	C.J.Watson	J.Noah	M.Dunleavy	M.Gasol	C.Frye	D.Green	B.Griffin	A.Iguodala

Chapter 4

Antisymmetric Respect

4.1 Problem Definition

In this chapter we define a variation of the *RespectMaximization* problem, based on having antisymmetric respect $R_{ij} = -R_{ji}$, where R_{ij} denotes the amount of respect worker j has for worker i. The concept of antisymmetric respect is based on the idea that if worker j has respect for worker i, then a hierarchy exists between them, in which i is higher than j, and worker i will have negative respect for worker j, since he is beneath him in the hierarchy.

We define this variation of the problem as follows. Given a set of n workers X, a task requiring a set of k skills S and an antisymmetric respect matrix T, for each skill s, create a team of workers $F \subseteq X$, where each skill is assigned a worker and the total respect of the team is maximized.

An antisymmetric matrix T has the property that $T^T = -T$, and $T[x, y] = -T[y, x]$ for each x, y in the bounds of T. More specifically, a respect matrix T^i for skill i is of size $n \times n$ and contains the respect values for each pair of workers, such that $T^i[x, y] = R_{xy}$.

For this variation too, we define a skill assignment as an injective function $f : S \to X$, where $f(i) = x_i$ is the worker assigned to skill i. We let $F = f(S)$ denote the selected team of experts. The respect $R^i(f)$ that worker $f(i)$ receives from his team members is defined as:

$$R^i(f) = \sum_{x_j \in F} R_{x_i x_j} \tag{4.1}$$

Therefore, the total respect for an assignment is:

$$R(f) = \sum_{j=1}^{k} R^j(f) \tag{4.2}$$

The problem can be broken into two cases based on the input on which the respect matrix is derived from, a ranking case and a general case.

4.2 The Ranking Case

In the ranking case, we assume that for each skill i, every worker x has a weight W_x^i. The weights give a partial or full order of the workers. We define the amount of respect that worker y has for worker x with respect to skill i as:

$$R_{xy}^i = W_x^i - W_y^i \tag{4.3}$$

If instead of weights we are given a ranked order of the nodes for skill i, then we derive these weights as a decreasing function of the position of x in the ranking of i, $W_x^i = n - rank^i(x)$. We refer to this problem as the *MaxRankingAntisymmetricRespect* problem.

We observe that in this case the respect for skill j is computed as:

$$\begin{aligned} R^j(f) &= \sum_{x_i \in F} \left[W_{x_j}^j - W_{x_i}^j \right] \\ &= kW_{x_j}^j - \sum_{x_i \in F} W_{x_i}^j \end{aligned} \tag{4.4}$$

Consequently, the total respect of an assignment f is computed as:

$$
\begin{aligned}
R(f) &= \sum_{j=1}^{k} R_j(f) && (4.5)\\
&= \sum_{j=1}^{k} kW^{j}_{x_j} - \sum_{j=1}^{k} \sum_{x_i \in F} W^{j}_{x_i} && (4.6)\\
&= \sum_{j=1}^{k} kW^{j}_{x_j} - \sum_{x_i \in F} \sum_{j=1}^{k} W^{j}_{x_i} && (4.7)\\
&= \sum_{x_i \in F} kW^{i}_{x_i} - \sum_{x_i \in F} \sum_{j=1}^{k} W^{j}_{x_i} && (4.8)\\
&= \sum_{x_i \in F} \left[kW^{i}_{x_i} - \sum_{j=1}^{k} W^{j}_{x_i} \right] && (4.9)\\
&= \sum_{x_i \in F} V(i, x_i) && (4.10)
\end{aligned}
$$

The value $V(i, x_i) = \left[kW^{i}_{x_i} - \sum_{j=1}^{k} W^{j}_{x_i} \right]$ is the contribution to the respect value of f for assigning x_i to skill i. Note that the function $V(i, x)$ is independent of the rest of the assignment, and depends only on the pair (i, x). Therefore, we can now approach the problem as a Maximum Weight Bipartite Matching problem [25].

In a Maximum Weight Bipartite Matching problem, given a bipartite graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}$, we are called to find a matching of maximum weight where the weight of matching M is given by $w(M) = \sum_{e \in M} w(e)$.

For our problem, given a bipartite graph $G = (N, E)$, where $N = (X + S)$, with bipartition (X, S), the weight function is $w(i, x_i) = V(i, x_i)$, where i is a skill in S and x_i is a worker in X. Our goal is to find a matching M with maximum weight, which is given by: $w(M) = \sum_{(i,x_i) \in M} w(i, x_i)$.

The maximum weight bipartite matching problem can be solved optimally in polynomial time using the Hungarian method, or approximately using a greedy approach. We now describe these two algorithms below.

4.2.1 Algorithms

Hungarian

The Hungarian method is commonly used to solve linear assignment problems optimally. For this method we created the *Hungarian* based on [26]. The *Hungarian* receives as input the weights derived over the ranking and computes the score values $V(i, x_i)$ for each skill-worker pair. Given these values the *Hungarian* creates a matrix P of size $n \times k$, where each row represents a worker x, each column a skill i, and each cell contains the value $V(i, x_i)$, which is the benefit of assigning worker x to skill i. The Hungarian method can only be applied to square matrices, thus matrix P is modified by adding rows or columns as needed and the empty cells are filled with the minimum value of matrix P. Now matrix P contains the benefit of assigning any worker to any skill and is called a profit matrix. The Hungarian method, though, works by minimizing the cost of an assignment, therefore *Hungarian* creates a cost matrix C. Matrix C is of size $n \times n$ and is a product of matrix P, $C = max(P) - P$. The *Hungarian* proceeds by modifying matrix C according to the following steps, as described in [27]:

1. Subtract minimum of each row from all elements in respective row and subtract minimum of each column from all elements in respective column.

2. Draw minimum number of horizontal and vertical lines to cover all zeros in the matrix.

   (a) Let N denote the number of lines needed and n denote the order of matrix C. If $N = n$, an optimal assignment can be made. Continue to step 5.

   (b) If $N < n$, continue with next step.

3. Find the smallest element x in C, that is not covered by lines, and subtract it from all elements not covered and add it to elements at intersection points of lines.

4. Repeat steps 2 & 3 until $N = n$.

5. Examine rows successively and find row containing a single zero element and mark the zero. Examine the column of marked zero and cross any zero found. Repeat until all rows have been examined, then repeat for all columns.

   (a) If no unmarked or uncrossed zero is left, an optimal solution has been found and corresponds to the workers and skills at the rows and columns of the marked zeros.

   (b) If unmarked or uncrossed zeros are left, continue with next step.

6. Randomly mark an unmarked and uncrossed zero and cross remaining zeros in its row and column.
   (a) If no unmarked or uncrossed zero is left, an optimal solution has been found and corresponds to the workers and skills at the rows and columns of the marked zeros.
   (b) If unmarked or uncrossed zeros are left, repeat current step until no more zeros are left.

MatchingGreedy

For the greedy approach we created the *MatchingGreedy* algorithm, whose outline can be seen in Algorithm 4.1. *MatchingGreedy* receives as input the weights derived by the ranking, computes the score values $V(i, x_i)$ for each skill i and each worker x, stores them in a list B as tuples in the form of $(score, worker, skill)$, and sorts them in a descending order based on the score value. The *MatchingGreedy* algorithm keeps a dictionary F that stores the assignments of workers to skills it makes, and lists W and S containing the workers and skills assigned respectively. The first assignment of the algorithm is the first value $V(i, x_i)$ in the sorted list. For each following value $V(i, x_i)$ in the list we examine if skill i has already been assigned a worker and if worker x has already been assigned a skill. If both of those statements are false, *MatchingGreedy* assigns worker x to skill i and moves on to the next value. If any of those statements is true, it moves on to the next value without making an assignment. The algorithm terminates when each skill has been assigned a worker or when the end of the list has been reached. The greedy approach may not always accomplish an optimal assignment due to assigning workers to skills as it encounters them without being able to change them later on if they find a better assignment later on.

4.3 The General Case

We now consider a more general case where the respect matrix is computed as follows. The input is, again, a directed graph G^i, for each skill. We define the respect for a pair of workers x, y for a skill i as:

$$R^i_{xy} = d^i(x, y) - d^i(y, x), \tag{4.11}$$

Algorithm 4.1 MatchingGreedy

Input: A dictionary W containing the weight of each worker in each skill.
Output: Assignment F.

```
B ← compute_score(W)
sort B in descending order (key: score)
F ← {}
W ← []
S ← []
for tuple(score, worker, skill) in B do
    if worker not in W and skill not in S then
        add tuple to F
        add worker to W
        add skill to S
    end if
    if length of F = number of skills then
        break
    end if
end for
return F
```

where function d^i denotes the shortest-path distance between the two workers in the graph. Intuitively, a large distance from x to y implies that x is "higher" than y and thus commands more respect. If there is no path from x to y in the graph, then the distance is zero. To define the respect between the two nodes, we take the difference of their distances in the graph. If the distance from worker x to worker y is greater than the distance from worker y to worker x for skill i, then x commands more respect from y than y demands from x, and thus R^i_{xy} is positive, while R^i_{yx} is negative.

We refer to this problem as the *MaxMutualAntisymmetricRespect* problem.

4.3.1 Algorithms

Landmark Algorithms

In this section, we propose a landmark-based method for the *MaxMutualAntisymmetricRespect* problem, aiming to reduce the general case to the *MaxRankingAntisymmetricRespect* problem.

Landmarks have been used to estimate distances between nodes in a graph (e.g., see [15]). The idea is that given a landmark node ℓ, we precompute the distance from ℓ to all other nodes in the graph, and we estimate the distance between two nodes x, y as $d(x,y) = d(x,\ell) + d(\ell,y)$. Multiple landmarks are used for more accurate estimation.

In our problem, if for a pair of nodes x, y we had a perfect landmark, such that, $d^i(x,y) = d^i(x,\ell) + d^i(\ell,y)$ and $d^i(y,x) = d^i(y,\ell) + d^i(\ell,x)$, then it would hold that $R^i_{xy} = R^i_{x\ell} - R^i_{y\ell}$. If this idealized landmark worked for all pairs of nodes in the graph, then we could assign to each node a weight $W^i_x = R^i_{x\ell}$ and our problem would reduce to the *MaxRankingAntisymmetricRespect* problem.

This idealized landmark does not exist, but we build on this idea to propose the following landmark-based heuristic algorithm. First, select a landmark ℓ^i for each skill i. Use this landmark to compute the respect $R^i_{x\ell^i}$ of the landmark ℓ^i to all nodes in the graph. Use these values as the weights $W^i_x = R^i_{x\ell^i}$ and apply the *MatchingGreedy* and *Hungarian* algorithms.

We examine three different ways of choosing the landmark.

1. *LowLandmark*: the worker with the lowest out-degree is assigned as landmark ℓ. Choosing the worker that is least respected by others for a skill as landmark is based on the idea that in a ranked order of the workers with regard to their incoming respect, such a worker would be placed at the bottom of the ranking. Then the distance from other workers to the landmark looks similar to the weight assigned to workers in the case where we are given a ranked order, as described in 4.2.

2. *RandomLandmark*: a worker is chosen uniformly at random to be assigned as landmark ℓ. The *RandomLandmark* variation is repeated $t = 100$ times and the assignment with the highest score is reported.

3. *AverageRandomLandmark*: initially a set L of $t = 100$ landmarks are chosen uni-

formly at random and $W_x^i = \frac{\sum_{\ell \in L} R_{x\ell}^i}{t}$. This is a more efficient variant of the random landmark selection, since we need to run the algorithm only once.

IQP_MaxRespect

For the *MaxMutualAntisymmetricRespect* problem, the *IQP_MaxRespect* algorithm defined in 3.3.2 can be applied, by replacing adjacency matrix A, with respect matrix T.

Greedy

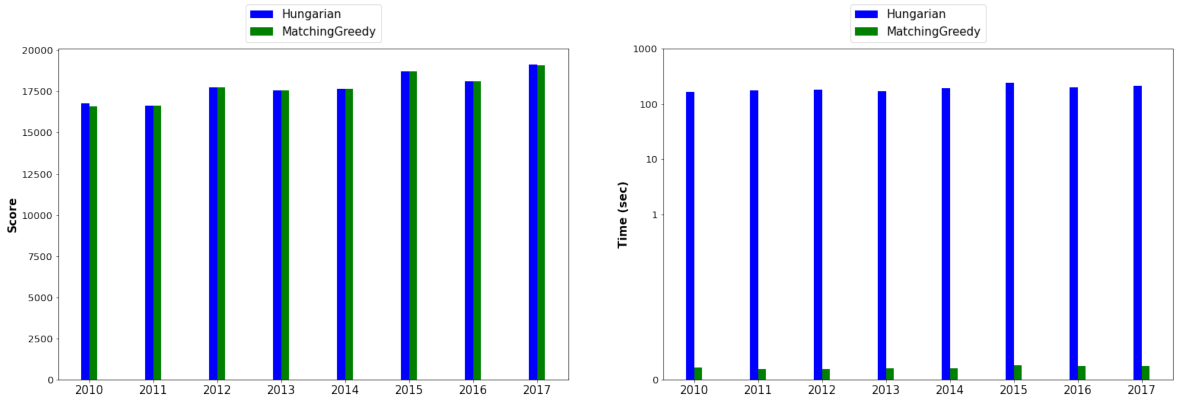
Additionally, based on the algorithms developed in [2] for their definition of the *MaxMutualRespect* problem, we created our version of the *Greedy* and *RandGreedy* algorithms. The *Greedy* algorithm initially assigns a score to every skill-worker pair, and makes the assignment with the highest score. For each next assignment an updated score value is computed, based on the already assigned workers, and the assignment with the highest score is made. The initial score value for a skill i and a worker x is computed as follows:

$$s(i,x) = \sum_{y \in X : y \neq x} R_{xy}^i + \frac{1}{k-1} \sum_{j \in S : j \neq i} \sum_{y \in X : y \neq x} R_{yx}^j, \tag{4.12}$$

where a high value in the first part of the above equation means that worker x is highly respected for skill i, and a high value in the second part of the equation means that worker x has high average respect for the remaining workers in the remaining skills. After the initial assignment is made, the updated score value for a skill i and a worker x is computed as:

$$s_F(i,x) = \sum_{y = f(j) \in F} R_{xy}^i + \sum_{y = f(j) \in F, j \in S} R_{yx}^j + \frac{1}{k - |F|} \sum_{j : f(j) = \emptyset, j \neq i, j \in S, y = f(j)} R_{yx}^j, \tag{4.13}$$

where a high value in the first part of the equation means that worker x is highly respected for skill i by the workers already assigned to team F, a high value in the second part that worker x highly respects the workers assigned to team F for their corresponding skills, and a high value in the third part means worker x has high average respect for the unassigned workers in the unassigned skills. The terms in the above values are normalized to be in the same scale.



(a) Score comparison between MatchingGreedy and Hungarian.

(b) Runtime comparison between MatchingGreedy and Hungarian.

Figure 4.1: Comparison of algorithms for the MaxRankingAntisymmetricRespect problem.

RandGreedy

The *RandGreedy* algorithm computes the score the same way as *Greedy*, but instead of selecting the pair (i, x) with the highest score for each assignment, it selects a skill $i \in S : f(i) = \emptyset$ uniformly at random, and then assigns the pair (i, x) with the highest score value. *RandGreedy* is repeated $t = 50$ times and the assignment with the highest score is reported.

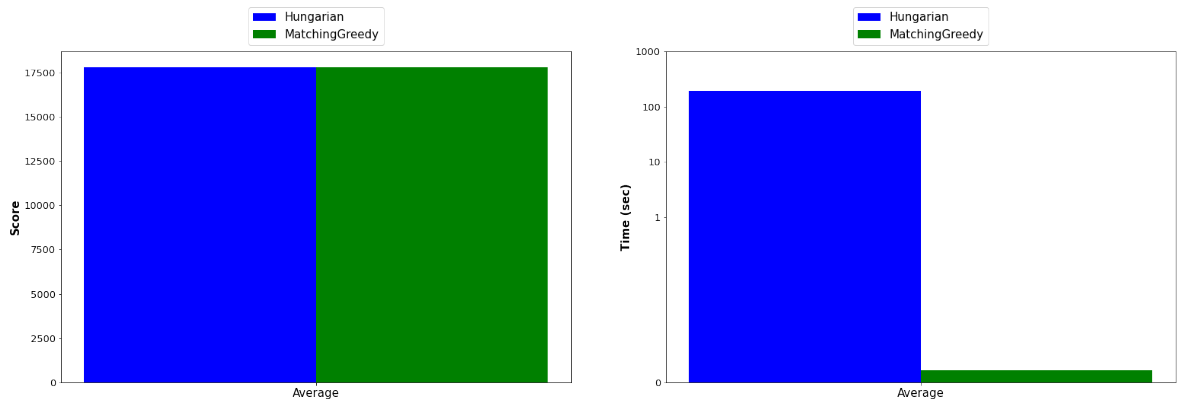
4.4 Experiments

4.4.1 Experiments for MaxRankingAntisymmetricRespect

To solve the *MaxRankingAntisymmetricRespect* problem, the same dataset as in 3.3.3 is being used.

Figure 4.1a shows the performance of the *MatchingGreedy* and the *Hungarian* algorithms. We observe that in most cases the assignment given by *MatchingGreedy* achieves the same score as the *Hungarian* assignment, which is the maximum score, except for the seasons 2010 and 2017.

In Figure 4.1b we compare the performance of *MatchingGreedy* and *Hungarian* regarding their running time. As expected, *MatchingGreedy* is more efficient achieving



(a) Average score comparison between MatchingGreedy and Hungarian.

(b) Average runtime comparison between MatchingGreedy and Hungarian.

Figure 4.2: Comparison of the average values of the algorithms for the MaxRankingAntisymmetricRespect problem.

Table 4.1: Teams produced by the Hungarian algorithm.

	2010	2011	2012	2013	2014	2015	2016	2017
STL	C.J.Watson	D.Fisher	C.Singleton	R.Rubio	P.Pressey	S.Larkin	T.J.McConnell	T.J.McConnell
AST	C.Duhon	J.Flynn	B.Udrih	J.Tinsley	K.Marshall	S.Blake	S.Blake	S.Rodriguezl
FT	J.Bayless	C.Maggette	C.Magette	T.Hansbrough	R.Stuckey	N.Young	S.Muhammad	S.Muhammad
BLK	J.Anthony	E.Udoh	B.Biyombo	B.Biyombo	E.Udoh	B.Biyombo	J.Henson	J.Grant
FG	C.Kaman	D.DeRozan	D.DeRozan	D.DeRozan	C.Boozer	T.Parker	D.Rose	Z.Randolph
TRB	U.Haslem	R.Evans	U.Haslem	R.Evans	O.Asik	O.Asik	O.Asik	T.Chandler
2P	T.Parker	J.J.Hickson	L.Scola	K.Seraphin	E.Kanter	C.Boozer	J.Okafor	D.Rose
3P	P.Stojakovic	J.Jones	S.Novak	S.Novak	M.Teletovik	T.Hardaway	H.Thompson	T.Daniels
DBPM	T.Ratliff	E.Dampier	A.Biedrins	A.Bogut	R.Turiaf	S.Dalembert	J.Noah	I.Mahinmi
OBPM	N.Robinson	B.Davis	J.Bayless	J.Calderon	J.J.Redick	S.Muhammad	C.Landryl	J.J.Barea
VORP	B.Wallace	J.Anthony	M.Ginobili	J.Kidd	A.Bogut	A.Bogut	A.Bogut	L.Nogueira

a running time of 1ms, while *Hungarian* is noticeably slower.

Figures 4.2a and 4.2b show the average respect score value and the average runtime value over all teams respectively. We can see that with respect to the score the two algorithms perform very closely, while with respect to the runtime *MatchingGreedy* is much more efficient than *Hungarian*.

Tables 4.1 and 4.2 show the assignments created by the *Hungarian* and *MatchingGreedy* algorithms respectively. We can observe that for the years 2011 - 2016 the assignments made by each algorithm are identical, while for the years 2010 and 2017 they differ for the roles FG and 2P, which we have highlighted in yellow for easier recognition.

Table 4.2: Teams produced by the MatchingGreedy algorithm.

	2010	2011	2012	2013	2014	2015	2016	2017
STL	C.J.Watson	D.Fisher	C.Singleton	R.Rubio	P.Pressey	S.Larkin	T.J.McConnell	T.J.McConnell
AST	C.Duhon	J.Flynn	B.Udrih	J.Tinsley	K.Marshall	S.Blake	S.Blake	S.Rodriguezl
FT	J.Bayless	C.Maggette	C.Magette	T.Hansbrough	R.Stuckey	N.Young	S.Muhammad	S.Muhammad
BLK	J.Anthony	E.Udoh	B.Biyombo	B.Biyombo	E.Udoh	B.Biyombo	J.Henson	J.Grant
FG	A.Stoudemire	D.DeRozan	D.DeRozan	D.DeRozan	C.Boozer	T.Parker	D.Rose	D.Rose
TRB	U.Haslem	R.Evans	U.Haslem	R.Evans	O.Asik	O.Asik	O.Asik	T.Chandler
2P	C.Kaman	J.J.Hickson	L.Scola	K.Seraphin	E.Kanter	C.Boozer	J.Okafor	Z.Randolph
3P	P.Stojakovic	J.Jones	S.Novak	S.Novak	M.Teletovik	T.Hardaway	H.Thompson	T.Daniels
DBPM	T.Ratliff	E.Dampier	A.Biedrins	A.Bogut	R.Turiaf	S.Dalembert	J.Noah	I.Mahinmi
OBPM	N.Robinson	B.Davis	J.Bayless	J.Calderon	J.J.Redick	S.Muhammad	C.Landryl	J.J.Barea
VORP	B.Wallace	J.Anthony	M.Ginobili	J.Kidd	A.Bogut	A.Bogut	A.Bogut	L.Nogueira

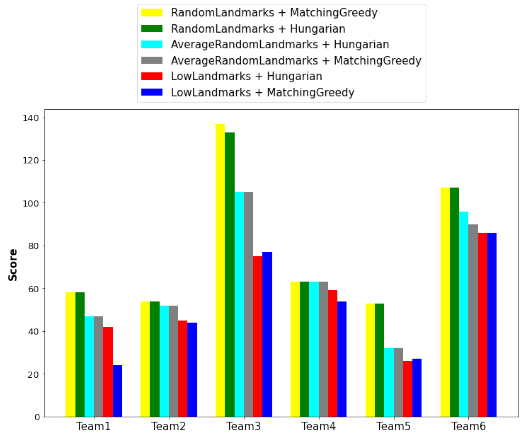
4.4.2 Experiments for MaxMutualAntisymmetricRespect

To solve the *MaxMutualAntisymmetricRespect* problem, we use the same dataset as in 3.3.3.

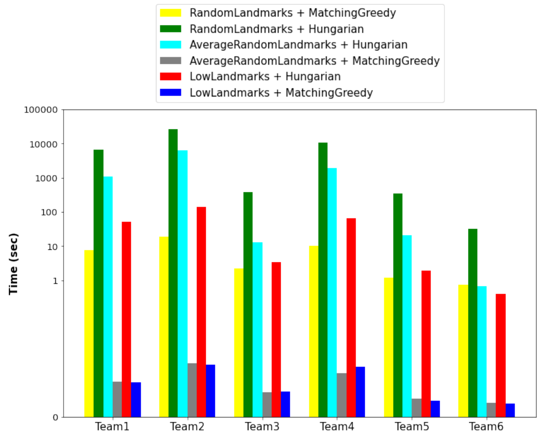
For the *MaxMutualAntisymmetricRespect* problem we first compare the *LowLandmark*, *RandomLandmark* and *AverageRandomLandmark* variations using the *MatchingGreedy* and *Hungarian* algorithms. Table 4.3 shows the assignments made by the landmark variations paired with the algorithms for the *MaxRankingAntisymmetricRespect* problem. We observe that *MatchingGreedy* and *Hungarian* make vary similar assignments for each variation. We also observe that often the same worker is assigned to a specific skill across different landmark variations, especially for the *RandomLandmark* and *AverageRandomLandmark* variations.

In Figure 4.3a the performance of all variation-algorithm pairs is shown. It is easily noticeable that the *RandomLandmark* variation performs best, for both algorithms used. Between the two algorithms though, it might seem surprising that the *MatchingGreedy* algorithm performs slightly better, since one would expect it to perform worse than *Hungarian*, due to its greedy nature. This happens because these algorithms use an approximation of the score to give an assignment, and given the assignment, the real score is computed, based on which they are evaluated. The *AverageRandomLandmark* variation performs slightly worse for both algorithms used, even though in some cases it gets very close to the performance of the *RandomLandmark* variation, while the *LowLandmark* variation performs poorly compared to the others.

Figure 4.3b shows the runtime analysis comparison for the *LowLandmark*, *RandomLandmark* and *AverageRandomLandmark* variations using the *MatchingGreedy* and



(a) Score comparison between landmark variations.



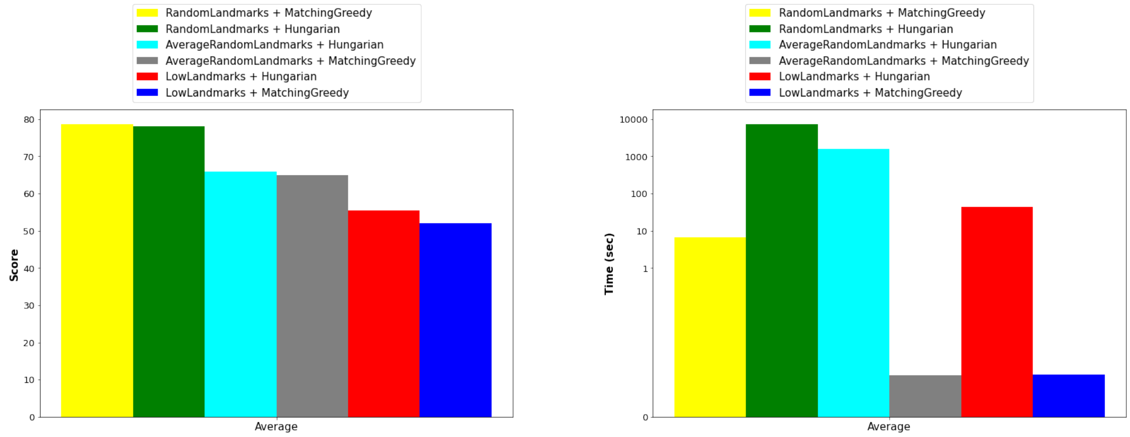
(b) Runtime comparison between landmark variations.

Figure 4.3: Comparison of landmarks variations using the MatchingGreedy and Hungarian algorithms.

Hungarian algorithms. We observe that overall all landmark variations paired with the *MatchingGreedy* algorithm perform better than the *Hungarian* algorithm. Specifically, the *LowLandmark* and *AverageRandomLandmark* variations combined with the *MatchingGreedy* algorithm are very efficient, having execution times of less than a second. The longer execution time of the *RandomLandmark* variation is explained by the number of $t = 100$ times the algorithm is repeated.

In Figures 4.4a and 4.4b the average respect score and runtime over all teams for each variation is shown. We observe that with respect to the score overall the *RandomLandmark* variation performs best, paired either with the *MatchingGreedy* or the *Hungarian* algorithm. Between, the two algorithms *MatchingGreedy* performs slightly better in this case. With respect to the runtime, we can see that overall the *MatchingGreedy* algorithm performs more efficiently.

Next we compare the best performing landmark variation-algorithm pair with the other algorithms for the *MaxMutualAntisymmetricRespect* problem. In Table 4.4 the assignments made by the *RandomLandmark* variation combined with the *MatchingGreedy* algorithm are shown, along with the assignments made by the *IQP_MaxRespect*, *Greedy* and *RandGreedy* algorithms. We observe that the assignments by the *IQP_MaxRespect* algorithm and the *RandGreedy* algorithm are very similar and even identical in the case of Team 2, Team 4 and Team 5. The assignments made by the *RandomLandmark* variation paired with the *MatchingGreedy* algorithm have some skill-worker pairs in



(a) Average score comparison between landmark variations.

(b) Average runtime comparison between landmark variations.

Figure 4.4: Comparison of the average values of the landmarks variations using the MatchingGreedy and Hungarian algorithms.

common with the assignments made by the *IQP_MaxRespect* algorithm, while the assignments made by the *Greedy* algorithm are the most different from the others.

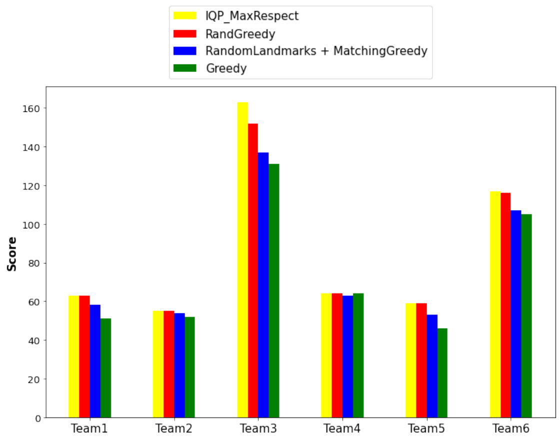
In Figure 4.5a the performance of the *RandomLandmark* variation paired with *MatchingGreedy* algorithm is shown compared to the *IQP_MaxRespect*, *Greedy* and *RandGreedy* algorithms. The results show that the *IQP_MaxRespect* algorithm performs best, slightly surpassing the *RandGreedy* algorithm, which also performs very good. The combination of the *RandomLandmark* variation with the *MatchingGreedy* algorithm performs quite close to the *RandGreedy* algorithm, while the *Greedy* algorithm performs slightly worse overall.

In Figure 4.5b the runtime analysis of the *RandomLandmark* variation combined with *MatchingGreedy* algorithm is shown compared to the *IQP_MaxRespect*, *Greedy* and *RandGreedy* algorithms. The *Greedy* algorithm and the pairing of the *RandomLandmark* variation with *MatchingGreedy* algorithm perform the best, followed by the *RandGreedy* algorithm. The *IQP_MaxRespect* algorithm performs quite poorly regarding the execution time, reaching the time limit of six hours that was applied in most cases.

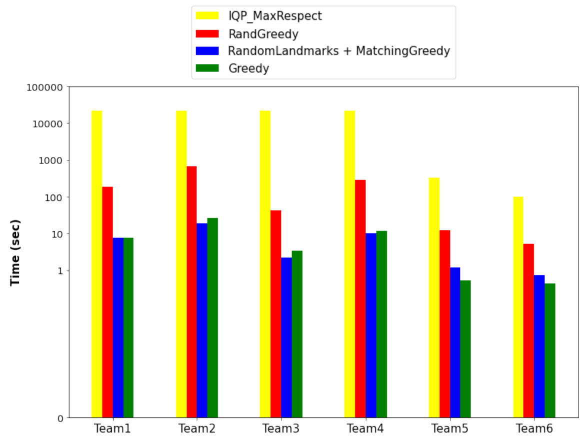
Figures 4.6a and 4.6b show the average respect score and runtime over all teams respectively. We observe that with respect to the score, the algorithms perform quite closely, with the *IQP_MaxRespect* performing the best. With respect to the runtime we see that the differences between the algorithms are more prominent, with the combination of *RandomLandmark* and *MatchingGreedy*having the lowest runtime.

Table 4.3: Teams produced by the landmark variations combined with the algorithms for the MaxRankingAntisymmetricRespect problem.

			AI	**NN**	**NLP**	**Robotics**	**DM**	**Algorithms**	**DB**	**Theory**
Team1	**MatchingGreedy**	**LowLandmarks**	L.Chen	M.Wang	Y.Chen	Y.Zhang	-	-	-	-
		RandomLandmarks	A.I.Rudnicky	J.Wang	C.Cerisara	T.Chen	-	-	-	-
		AverageRandomLandmarks	Y.Wang	J.Wang	C.Cerisara	I.Noda	-	-	-	-
	Hungarian	**LowLandmarks**	L.Chen	J.Wang	Y.Chen	G.Z.Grudic	-	-	-	-
		RandomLandmarks	A.I.Rudnicky	J.Wang	C.Cerisara	T.Chen	-	-	-	-
		AverageRandomLandmarks	Y.Wang	J.Wang	C.Cerisara	I.Noda	-	-	-	-
Team2	**MatchingGreedy**	**LowLandmarks**	-	-	-	-	S.Zilles	M.Purohit	M.Tang	A.Nichterlein
		RandomLandmarks	-	-	-	-	D.P.Miranker	B.D.Sullivan	Y.Shavitt	R.Bredereck
		AverageRandomLandmarks	-	-	-	-	D.P.Miranker	M.Karppa	D.S.Papailiopoulos	R.Bredereck
	Hungarian	**LowLandmarks**	-	-	-	-	K.M.Hammouda	A.Zelikovsky	A.Faria	A.Nichterlein
		RandomLandmarks	-	-	-	-	D.P.Miranker	B.D.Sullivan	A.Stupar	A.Nichterlein
		AverageRandomLandmarks	-	-	-	-	D.P.Miranker	Y.Kohonen	D.S.Papailiopoulos	R.Bredereck
Team3	**MatchingGreedy**	**LowLandmarks**	S.Parthasarathy	H.Cheng	R.Liu	R.Motwani	W.K.Wong	Y.Wang	K.Sun	A.Das
		RandomLandmarks	Y.Cheng	S.Pandey	X.Li	T.Jebara	W.K.Wong	D.Agarwal	H.Zhang	J.Langford
		AverageRandomLandmarks	T.Roughgarden	S.Pandey	Qiang Li	Qian Li	Y.Aumann	X.Li	H.Zhang	A.Das
	Hungarian	**LowLandmarks**	S.Parthasarathy	H.Cheng	J.Chen	R.Motwani	W.K.Wong	C.V.Jawahar	K.Sun	A.Das
		RandomLandmarks	Y.Cheng	S.Pandey	X.Li	K.P.Sycara	W.K.Wong	D.Agarwal	H.Zhang	J.Langford
		AverageRandomLandmarks	T.Roughgarden	S.Pandey	Qiang Li	Qian Li	Y.Aumann	X.Li	H.Zhang	A.Das
			SE	**HPC**	**DPC**	**OS**	**SP**	**CN**	**IR**	**Wireless**
Team4	**MatchingGreedy**	**LowLandmarks**	K.Razavi	R.Prodan	Z.Li	W.Dong	-	-	-	-
		RandomLandmarks	F.Zhang	R.Prodan	Z.Li	W.Dong	-	-	-	-
		AverageRandomLandmarks	F.Zhang	R.Prodan	Z.Li	W.Dong	-	-	-	-
	Hungarian	**LowLandmarks**	J.Li	R.Prodan	Z.Prodan	W.Dong	-	-	-	-
		RandomLandmarks	F.Zhang	R.Prodan	Z.Li	W.Dong	-	-	-	-
		AverageRandomLandmarks	F.Zhang	R.Prodan	Z.Li	W.Dong	-	-	-	-
Team5	**MatchingGreedy**	**LowLandmarks**	-	-	-	-	Y.Gao	J.Li	Q.Zheng	L.M.Kaplan
		RandomLandmarks	-	-	-	-	Y.Chen	Y.Li	Y.Zhang	R.Zhang
		AverageRandomLandmarks	-	-	-	-	J.Liu	H.Li	Y.Zhang	L.M.Kaplan
	Hungarian	**LowLandmarks**	-	-	-	-	J.Liu	J.Li	Q.Zheng	L.M.Kaplan
		RandomLandmarks	-	-	-	-	Y.Chen	Y.Li	Y.Zhang	R.Zhang
		AverageRandomLandmarks	-	-	-	-	J.Liu	H.Li	Y.Zhang	L.M.Kaplan
Team6	**MatchingGreedy**	**LowLandmarks**	C.Wang	Y.Li	Yu.Zhang	Q.Yang	W.Zhang	Yi.Zhang	B.Zhang	W.Li
		RandomLandmarks	C.Wang	Y.Li	L.Zhang	X.Li	R.Zhang	Yi.Zhang	Ya.Zhang	W.Li
		AverageRandomLandmarks	C.Wang	Yu.Zhang	L.Zhang	Q.Yang	R.Zhang	Yi.Zhang	Y.Li	Y.Yang
	Hungarian	**LowLandmarks**	C.Wang	Y.Li	Yu.Zhang	Q.Yang	W.Zhang	Yi.Zhang	B.Zhang	W.Li
		RandomLandmarks	Y.Li	Ya.Zhang	L.Zhang	X.Li	R.Zhang	Yi.Zhang	C.Wang	W.Li
		AverageRandomLandmarks	C.Wang	Y.Li	L.Zhang	Q.Yang	R.Zhang	Yi.Zhang	Yu.Zhang	Y.Yang

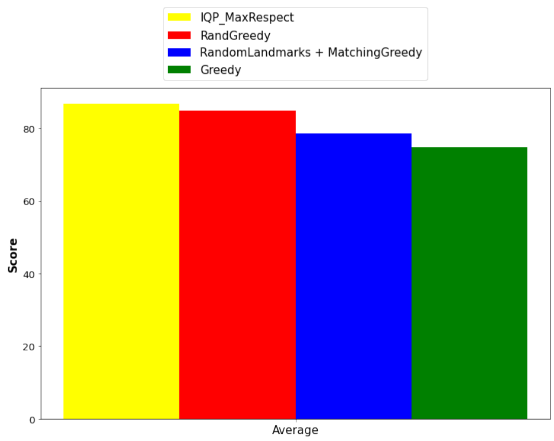


(a) Score comparison between algorithms for MaxMutualAntisymmetricRespect.

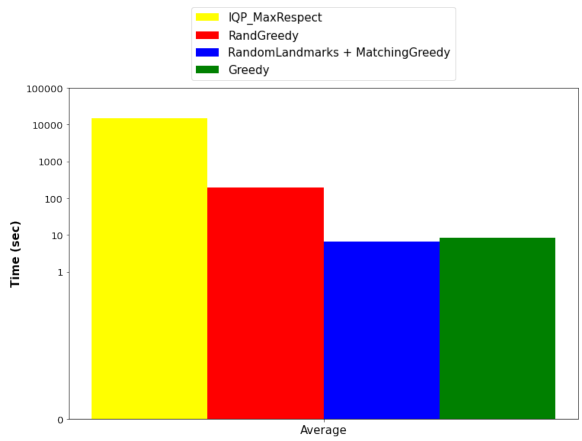


(b) Runtime comparison between algorithms for MaxMutualAntisymmetricRespect.

Figure 4.5: Comparison of algorithms for the MaxMutualAntisymmetricRespect problem.



(a) Avrege score comparison between algorithms for MaxMutualAntisymmetricRespect.



(b) Average runtime comparison between algorithms for MaxMutualAntisymmetricRespect.

Figure 4.6: Comparison of the average values of the algorithms for the MaxMutualAntisymmetricRespect problem.

Table 4.4: Teams produced by the algorithms for the MaxMutualAntisymmetricRespect problem.

		AI	**NN**	**NLP**	**Robotics**	**DM**	**Algorithms**	**DB**	**Theory**
Team1	**RandomLandmarks + MatchingGreedy**	A.I.Rudnicky	J.Wang	C.Cerisara	T.Chen	-	-	-	-
	RespectMaximization	A.I.Rudnicky	D.Wu	C.Cerisara	W.Wei	-	-	-	-
	Greedy	Y.Wang	Y.Zhao	C.Cerisara	T.Chen	-	-	-	-
	RandGreedy	K.Komatani	D.Wu	C.Cerisara	W.Wei	-	-	-	-
Team2	**RandomLandmarks + MatchingGreedy**	-	-	-	-	D.P.Miranker	B.D.Sullivan	Y.Shavitt	R.Bredereck
	RespectMaximization	-	-	-	-	D.P.Miranker	B.D.Sullivan	J.Liu	A.Nichterlein
	Greedy	-	-	-	-	Q.He	J.Kohonen	A.Stupar	R.Bredereck
	RandGreedy	-	-	-	-	D.P.Miranker	B.D.Sullivan	J.Liu	A.Nichterlein
Team3	**RandomLandmarks + MatchingGreedy**	Y.Cheng	S.Pandey	X.Li	T.Jebara	W.K.Wong	D.Agarwal	H.Zhang	J.Langford
	RespectMaximization	T.Roughgarden	Y.Li	M.I.Jordan	L.Zhang	C.Chen	D.Agarwal	H.Zhang	J.Langford
	Greedy	S.Mannor	S.Pandey	T.Huang	T.Jebara	G.Li	YWang	J.Vondrk	A.Karbasi
	RandGreedy	A.K.Jain	Y.Li	Q.Li	T.Roughgarden	Y.Aumann	D.Agarwal	Y.Chen	J.Langford
		SE	**HPC**	**DPC**	**OS**	**SP**	**CN**	**IR**	**Wireless**
Team4	**RandomLandmarks + MatchingGreedy**	F.Zhang	R.Prodan	Z.Li	W.Dong	-	-	-	-
	RespectMaximization	F.Zhang	R.Prodan	B.Balasubramanian	W.Dong	-	-	-	-
	Greedy	F.Zhang	R.Prodan	B.Balasubramanian	W.Dong	-	-	-	-
	RandGreedy	F.Zhang	R.Prodan	B.Balasubramanian	W.Dong	-	-	-	-
Team5	**RandomLandmarks + MatchingGreedy**	-	-	-	-	Y.Chen	Y.Li	Y.Zhang	R.Zhang
	RespectMaximization	-	-	-	-	K.Wang	Y.Li	Y.Zhang	D.Wang
	Greedy	-	-	-	-	H.Chen	H.V.Poor	Y.Zhang	L.M.Kaplan
	RandGreedy	-	-	-	-	K.Wang	Y.Li	Y.Zhang	D.Wang
Team6	**RandomLandmarks + MatchingGreedy**	C.Wang	Y.Li	L.Zhang	X.Li	R.Zhang	Yi.Zhang	Ya.Zhang	W.Li
	RespectMaximization	Ya.Zhang	J.Wang	L.Zhang	Y.Li	R.Zhang	Yi.Zhang	Yu.Zhang	W.Li
	Greedy	Ya.Zhang	H.Wang	L.Zhang	Y.Li	R.Zhang	Yi.Zhang	Yu.Zhang	Y.Yang
	RandGreedy	Ya.Zhang	H.Wang	L.Zhang	Y.Li	R.Zhang	Yi.Zhang	Yu.Zhang	W.Li

Chapter 5

Conclusion

5.1 Future Work

In this thesis we studied and extended a variation of the Team Formation problem [1], the *RespectMaximization* problem, that has been previously defined in [2]. The *RespectMaximization* problem takes into consideration the fact that social relations are not always reciprocal and may vary depending on the criteria. It also incorporates the concept of respect between workers, which is to be maximized in the assigned teams.

Our contribution to that work is the proposition of an IQP formulation of the *RespectMaximization* problem, and the heuristic algorithm that solves it. We showed that our algorithm achieves the assignment of teams with higher respect score than previous algorithms, albeit with much higher computational cost. Our heuristic algorithm was also applied to the *MaxRankingRespect* problem, where the assigned teams were of maximum respect, but the high computational cost renders our algorithm unnecessary in this case, since we do not gain in any aspect.

Thereafter, we introduce a variation of the *RespectMaximization* problem, with anti-symmetric respect, and implement polynomial algorithms to solve it. For the ranking case we showed that our *MatchingGreedy* algorithm performs very close to the *Hungarian* algorithm. For the general case we showed that our heuristic landmark algorithm performs very efficiently compared to the IQP and other heuristic algorithms, assigning teams with respect score close to that of the other algorithms, with lower computational cost.

5.1 Future Work

In the future it would be interesting to explore the IQP formulation with respect to the ranking case more deeply, in order to obtain a more efficient IQP program that does not have such high computational cost.

Additionally, studying more variations of landmark selection in a graph for the *MaxMutualAntisymmetricRespect* problem, could lead to improving the effectiveness of this approximation approach. The incorporation of more than one landmark could also improve the ability to approximate the true respect score value.

An extension of the *RespectMaximization* problem worth considering is the introduction of the concept of respect to the Template-Driven Team Formation (TDTF) problem defined in [23]. In the TDTF problem, the teams wanted to accomplish a task have a certain structure with a hierarchy among the workers. Incorporating the concept of respect so that the workers lower in the hierarchy respect those above them would make the problem even more realistic.

Lastly, we suggest the examination of a case, where given a task, the subgraph induced over the assigned workers should be a Directed Acyclic Graph (DAG). Such a DAG creates a hierarchy among the workers based on their respect relationships.

Bibliography

[1] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *KDD*, 2009.

[2] S. M. Nikolakaki, E. Pitoura, E. Terzi, and P. Tsaparas, "Finding teams of maximum mutual respect," in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020.

[3] A. Zakarian and A. Kusiak, "Forming teams: an analytical approach," *IIE Transactions*, vol. 31, no. 1, pp. 85–97, 1999.

[4] A. Baykasoglu, T. Dereli, and G. Daş, "Project team selection using fuzzy optimization approach." *Cybernetics and Systems*, vol. 38, pp. 155–185, 02 2007.

[5] C.-T. Li, M.-K. Shan, and s.-d. Lin, "On team formation with expertise query in collaborative social networks," *Knowledge and Information Systems*, vol. 42, 02 2015.

[6] A. Gajewar and A. Das Sarma, "Multi-skill collaborative teams based on densest subgraphs," *Computing Research Repository - CORR*, 02 2011.

[7] R. Bredereck, J. Chen, F. Hüffner, and S. Kratsch, "Parameterized complexity of team formation in social networks," *Theoretical Computer Science*, vol. 717, pp. 26–36, 2018.

[8] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Machine Learning and Knowledge Discovery in Databases*, P. A. Flach, T. De Bie, and N. Cristianini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 483–498.

[9] M. Kargar, M. Zihayat, and A. An, *Finding Affordable and Collaborative Teams from a Network of Experts*. SIAM, 05 2013, pp. 587–595.

[10] B. Golshan, T. Lappas, and E. Terzi, "Profit-maximizing cluster hires," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1196–1205. [Online]. Available: https://doi.org/10.1145/2623330.2623690

[11] A. Majumder, S. Datta, and K. Naidu, "Capacitated team formation problem on social networks," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1005–1013.

[12] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Power in unity: Forming teams in large-scale community systems," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 599–608.

[13] ——, "Online team formation in social networks," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 839–848.

[14] S. M. Nikolakaki, A. Ene, and E. Terzi, "An efficient framework for balancing submodularity and cost," *KDD*, 2021.

[15] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis, "Fast shortest path distance estimation in large networks," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 867–876.

[16] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1077–1088.

[17] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 985–994.

[18] X. Yin, C. Qu, Q. Wang, F. Wu, B. Liu, F. Chen, X. Chen, and D. Fang, "Social connection aware team formation for participatory tasks," *IEEE Access*, vol. 6, pp. 20 309–20 319, 2018.

[19] A. Bhowmik, V. Borkar, D. Garg, and M. Pallan, "Submodularity in team formation problem *," in *SDM*, 12 2013.

[20] I. Kouvatis, K. Semertzidis, M. Zerva, E. Pitoura, and P. Tsaparas, "Forming compatible teams in signed networks," in *Proceedings of the 23nd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*. OpenProceedings.org, 01 2020, pp. 363–366.

[21] S. M. Nikolakaki, O. Dibie, A. Beirami, N. Peterson, N. Aghdaie, and K. A. Zaman, "Competitive balance in team sports games," *2020 IEEE Conference on Games (CoG)*, pp. 526–533, 2020.

[22] Z. Cheng, Y. Yang, C. Tan, D. Cheng, A. Cheng, and Y. Zhuang, "What makes a good team? a large-scale study on the effect of team composition in honor of kings," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2666–2672.

[23] S. Apostolou, P. Tsaparas, and E. Terzi, "Template-driven team formation," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2020, pp. 258–265.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[25] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, 1st ed. Prentice Hall Englewood Cliffs, N.J, 1993.

[26] Eason. Hungarian algorithm introduction & python implementation. [Online]. Available: https://python.plainenglish.io/hungarian-algorithm-introduction-python-implementation-93e7c0890e15

[27] J. Dutta and S. C. Pal, "A note on hungarian method for solving assignment problem," in *Journal of Information and Optimization Sciences*, 2015.

Short Biography

I began my studies at the Ionian University in the Department of Informatics. There I followed their undergraduate Information Systems program. Through the lectures and the projects I discovered my interest in Information Security, which led me to writing my thesis with the title "Analysis of Information Security Risk Analysis and Management Methods and Software".

After the completion of my undergraduate studies, I wanted to expand my knowledge in the direction of Data Analysis, leading me to follow the postgraduate program of the University of Ioannina in the Department of Computer Science and Engineering, with the title "Master of science in Data and Computer Systems Engineering" and specialization in "Data Science and Engineering". During my studies there, I attended a multitude of courses and got to know the professors and their academic interests. Thus, I decided to conclude my studies by writing my thesis with the title "Team Formation with Mutual Respect", having the Associate Professor Panayiotis Tsaparas as my advisor.