**A close-up of a sign

Description automatically generated**

**Solving NP hard optimization problem using stochastic manners**

**Tal Oved and Elad Mandinach**

**Supervisor: Dr. Daniel Sigalov**

**Spring 2023**

Table of Contents

[**No table of figures entries found.** 3](#_Toc144839263)

[**Abstract** 4](#_Toc144839264)

[**Symbols list** 6](#_Toc144839265)

[**Introduction** 7](#_Toc144839266)

[Simulated Annealing 8](#_Toc144839267)

[Cross Entropy 10](#_Toc144839268)

[**Implementation** 11](#_Toc144839269)

[Simulated Annealing 11](#_Toc144839270)

[Cross Entropy 13](#_Toc144839271)

[**Summary and conclusions** 15](#_Toc144839272)

[**References** 16](#_Toc144839273)

# Figures

[Figure 1: Fully connected Graph 8](#_Toc146017145)

[Figure 2: Representative matrix 8](#_Toc146017146)

[Figure 3 17](file:////Users/eladmandinach/Desktop/Technion/Projects/ProjectA/ProjectATechnion/projectA%20final%20doc.docx#_Toc146017147)

# Abstract

In the realm of sports, the composition of teams plays a pivotal role in determining tournament outcomes. The intricate arrangement of athletes can significantly influence a team's performance. To address this challenge, our research project focuses on optimizing team compositions.

We tackle the NP-hard problem of team composition optimization, a complex task that cannot be efficiently solved with traditional methods. Drawing inspiration from established algorithms, we have developed our own customized solutions, primarily relying on two polynomial complexity methods: Simulated Annealing and Cross Entropy.

Our algorithms are tailored to the specific task of forming teams that encompass various combinations, including individual athletes, pairs, trios, and quartets, while adhering to predefined rules and considering the strength of connections between athletes. Additionally, we explore a novel approach by combining Simulated Annealing and Cross Entropy to further enhance team composition optimization.

In the implementation phase, mathematical models are used to represent the suitability of athletes for team formation, and we employ these models to evaluate the overall quality of each solution. Simulated Annealing iteratively refines team configurations by considering both legality and cost, while Cross Entropy utilizes a probability matrix to probabilistically generate teams, thereby increasing the likelihood of successful pairings.

One of the noteworthy achievements is our successful implementation of the algorithms, with polynomial complexity. This accomplishment is particularly important given the NP-hard nature of the problem we addressed. Through rigorous testing and experimentation, we observed that our algorithms consistently delivered efficient solutions while adhering to reasonable time frames. These results indicate that our tailored approaches are not only effective but also practical for real-world sports scenarios. Coaches and sports professionals can now harness the power of these algorithms to refine their team compositions and enhance performance, all within manageable computational timeframes.

# Symbols list

* CE – Cross Entropy
* SA – Simulated Annealing
* Price – the evaluation of solution
* Elite solution – top priced solutions
* Connectivity level – the strength of the connection between two athletes (nodes)
* NP hard problem – problem that couldn’t be solved by polynomial complexity
* Temperature - represents a control parameter that regulates the probability of accepting worse solutions

# Introduction

The focal challenge we address is the optimization of team compositions for athletes. This issue holds relevance in sports contexts where the configuration of a team can wield substantial influence over tournament outcomes. Our project aims to provide coaches with a tool to refine their team compositions, thereby attaining maximum efficiency from the group.

In essence, we are addressing a broader spectrum of optimization problems, with athlete partitioning being just one of the many domains that benefit from our algorithm, with the right adjustments.

Within this endeavor, our focus lies in identifying algorithms that yield optimal team configurations, while adhering to constraints governing team makeup. Specifically, we consider teams structured as follows:

- Individual athletes (male/female)  
- Pairs of female athletes  
- Pairs of male athletes  
- Trios of female athletes  
- Quartets of male athletes  
- Pairs comprising one male and one female athlete

Our approach involves utilizing a fully connected weighted graph, represented as a matrix. In this graph, nodes correspond to group members, and edges denote the strength of connectivity between two members (with higher weights indicating stronger connections).

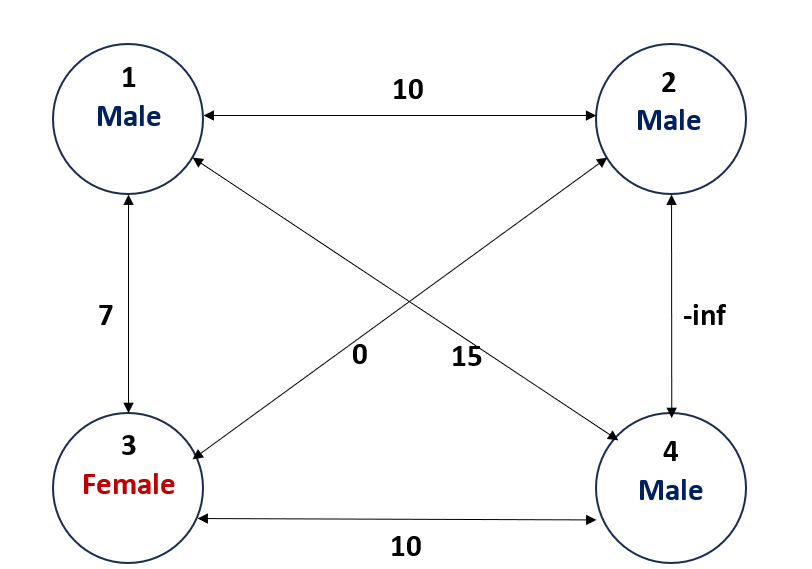


Figure 1: Fully connected Graph

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 10 | 7 | 15 |
| 10 | 0 | 0 | -inf |
| 7 | 0 | 0 | 10 |
| 15 | -inf | 10 | 0 |

Figure 2: Representative matrix

It's important to note that the challenge at hand is categorized as an NP-hard problem, implying that finding a solution in polynomial time is unfeasible.

In this project, our primary objective revolves around employing two polynomial complexity methods to solve the problem: Simulated Annealing and Cross Entropy.

Simulated Annealing

Simulated Annealing is a metaheuristic optimization algorithm that simulates the annealing process observed in metallurgy. Beginning with a random solution, the algorithm starts at a high-temperature state, initially emphasizing exploration. Over time, the temperature reduces, leading to reduced exploration and the hopeful convergence towards the optimal solution.

Cross Entropy

Cross Entropy is an iterative optimization algorithm that constructs a probabilistic model of potential solutions. Through iterative updates of model parameters, the algorithm aims to converge towards the optimal solution.

Our methodology first entailed adapting these algorithms to our specific problem context, using the MATLAB environment for implementation. Subsequently, we conducted comprehensive performance tests, evaluating factors such as algorithm complexity, accuracy in identifying known solutions, and constraints on the group size that the algorithm could effectively manage with fixed parameters.

Furthermore, we explored a novel approach by combining the strengths of both algorithms. Specifically, we generated an initial solution using the Simulated Annealing technique, and then fed this solution into the Cross-Entropy algorithm. The resulting hybrid algorithm's performance and effectiveness were analyzed extensively.

By undertaking these steps, our project strives to provide coaches and sports professionals with a valuable tool for optimizing team compositions, contributing to enhanced performance and strategic decision-making in the world of sports.

# Implementation

Firstly, in order to implement each of the approaches, we needed to determine the mathematical representation of the suitability between two different athletes. We chose a symmetric matrix, denoted as M, where for all represents the suitability between athlete i and athlete j, when:

To account for impossible matches, such as those with a significant age gap or conflicts between athletes, the suitability level of the athletes was assigned a value of negative infinity (-∞).

Secondly, we had to determine the method for evaluating the overall solution. Given our objective of maximizing our chances of winning competitions by achieving the highest suitability score, it was decided that the solution's cost would be evaluated as follows:

As we can see, there is ‘preference’ for bigger teams, which will be in more observed matches.

## Simulated Annealing

When employing Simulated Annealing (SA), our initial step involves selecting a random legally valid starting solution. For simplicity, we opt for the naive solution, which assigns each athlete to a different team.

Subsequently, during each iteration, we randomly select an athlete and a team number to consider for placement. If this proposed move violates any legality (e.g., rendering the original or new team illegal), we discard the change and generate a new proposal. However, if the move is deemed legal, we assess its impact on the overall team's cost. If the move increases the cost, we accept it and proceed to explore further changes. Conversely, if the move decreases the cost, we accept it with a probability that hinges on how much the cost reduction is and the current temperature of the system while making that decision.

Notably, in each iteration, there exists a possibility of transferring two athletes in a single step, which helps circumvent the constraint that allows groups of four men but disallows groups of three.

We halt the attempts to reassign athletes to different teams under two conditions: either when we repeatedly maintain the same solution for several consecutive iterations, or when we reach the predefined final temperature set at the outset.

## Cross Entropy

In the implementation of the Cross Entropy approach, a crucial consideration is how to represent the probability of two athletes being on the same team. To address this, we employ a stochastic matrix, with the requirement that the rows sum to 1. It's noteworthy that this matrix isn't necessarily symmetric. Here, signifies the probability of athlete i being paired with athlete j. The lack of symmetry arises from the fact that different athletes possess distinct options for team memberships, and this approach relies on probabilities.

Additionally, we've devised an efficient method for generating solutions. The team formation process operates in a regressive manner. Each team's creation begins with a random permutation of all team members. We begin with the first athlete, denoted as i, and then sequentially evaluate all other athletes according to the random permutation. Athlete j is added to the developing team with a probability governed by . If including athlete j in the team would violate any rules, we proceed to the next athlete option. This cycle continues, considering all athletes cyclically until the team is complete or a random 'Stop Flag' is triggered with a certain probability.

After generating numerous teams for a given set of probabilities, we select elite solutions and extract the subsequent set of probabilities from them. For each athlete, we determine the athletes they most frequently formed teams with among these elite teams. We then increase the probability of the athlete being paired with these specific athletes in the next iteration.

# Summary and conclusions

# In our project, we managed to implement both solutions, one that is inspired by cross entropy and one that is inspired by simulated annealing. More than that, we combined our two solutions to make a solution that takes “the best of both worlds”.

## Cross Entropy results

## Simulated Annealing:

# Simulated Annealing (SA) is a powerful optimization algorithm that we applied to the challenging problem of team composition optimization in the realm of sports. In this section, we will provide a summary of our SA implementation, analyze its time complexity, and discuss the results we obtained through extensive experimentation.

# **Summary of Simulated Annealing Implementation:**

# Our implementation of Simulated Annealing started with a random initial solution, which assigned athletes to teams. The algorithm then iteratively explored possible moves, such as reassigning athletes to different teams, while considering both the legality of the move and the cost associated with it. The cost function aimed to maximize the overall suitability of team compositions, which directly impacted tournament outcomes. We also introduced the concept of temperature, which controlled the probability of accepting worse solutions early in the optimization process but decreased over time to converge towards an optimal solution.

# **Time Running Complexity Analysis:**

# SA's complexity depends on several factors, including problem size, the cooling schedule, and the number of iterations. Typically, SA exhibits a time complexity of O(N∙I), where N represents the number of iterations, and I reflects the complexity associated with evaluating the cost function and checking the legality of moves.

# During our experiments, we observed that the algorithm's execution time exhibited notable variations based on the problem size and the number of iterations. For instance, in our specific model involving 13 individuals (resulting in a 13x13 matrix), it took approximately 4000 iterations to reach convergence. On average, over 50 runs of the algorithm, the execution time averaged 3.399226 seconds on the MacBook Pro with an Intel Core i7 processor that we used for testing.

# Need to add another example

# This analysis provides valuable insights into the algorithm's performance under different conditions and serves as a reference for understanding its computational demands in real-world applications.

# Result Analysis

# To evaluate the effectiveness of Simulated Annealing in optimizing team compositions, we conducted extensive experiments and analyzed the results. Our primary focus was on the following aspects:

# **Convergence**:

# We monitored the algorithm's performance improvement over iterations. As shown in Figure 3 below, the temperature decreased, indicating that the algorithm gradually converged towards a more optimal solution, demonstrating the effectiveness of the annealing process.

תמונה שמכילה טקסט, קו, תרשים, עלילה

התיאור נוצר באופן אוטומטי

Figure 3

# **Final Result**:

# We generated a result graph that depicted the suitability (price) of team compositions as a function of iteration, [Figure 4](#_Figures). This graph allowed us to visualize how the algorithm's performance evolved over time. Typically, we observed a decreasing trend in the price, indicating that SA effectively improved team compositions.

# Additionally, we created a final result graph to provide a more intuitive visualization of the outcomes, as demonstrated in [Figure 5](#_Figures).

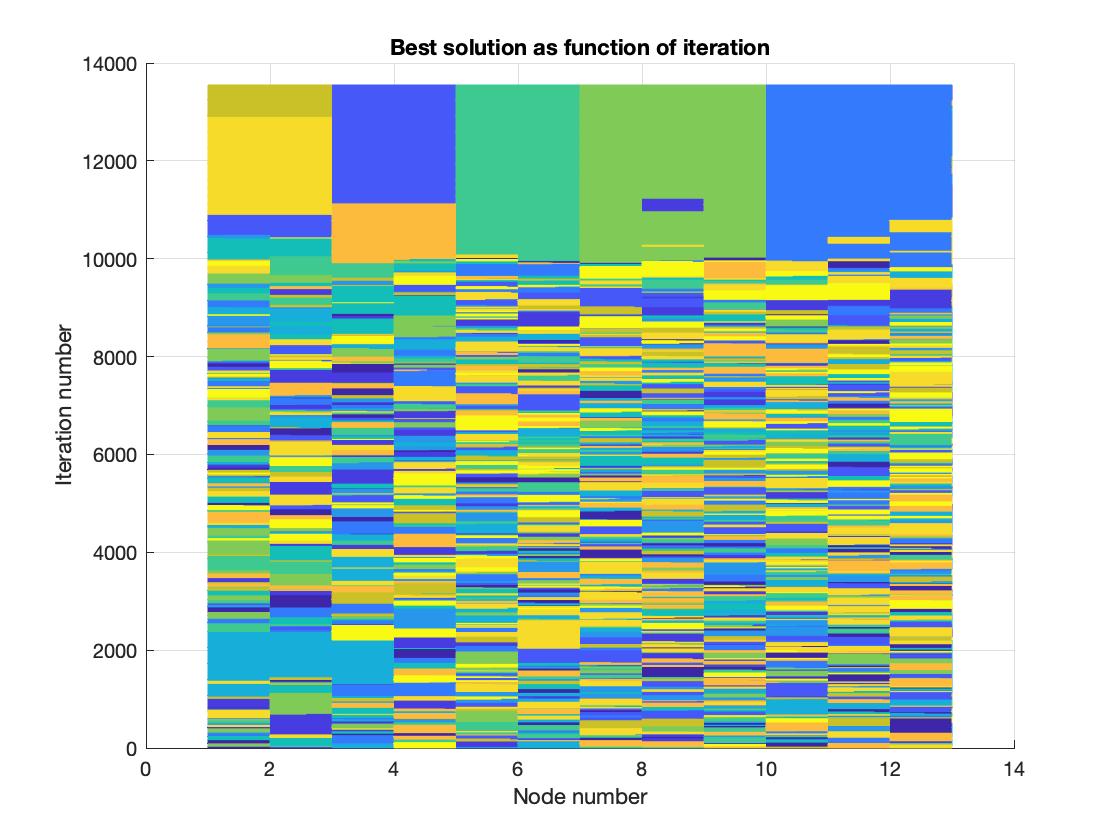


Figure 4

תמונה שמכילה צילום מסך, תרשים, קו, טקסט

התיאור נוצר באופן אוטומטי

Figure 5: Final Result Graph (13 individuals)

# **Time Efficiency**:

# We assessed the algorithm's ability to deliver efficient solutions within reasonable time frames. SA consistently provided good solutions even for large problem sizes, making it a practical choice for real-world sports scenarios.

# **Temperature Control**:

# We analyzed the temperature as a function of iteration to understand how the algorithm transitioned from exploration to exploitation. The temperature decreased gradually, as illustrated in [Figure 3](#_Figures), indicating that the algorithm effectively balanced exploration and exploitation to reach an optimal solution.

# In conclusion, Simulated Annealing proved to be a valuable tool for team composition optimization in sports. It effectively balanced exploration and exploitation, converging towards optimal solutions while efficiently managing computational time constraints. The algorithm's performance improved over iterations as illustrated in [Figure 3](#_Figures).

# SA, with its controllable temperature and iterative refinement, can greatly benefit coaches and sports professionals seeking to enhance team performance through optimized compositions.

## Combines solution results

# References

A Tutorial on the Cross-Entropy Method

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, Reuven Y. Rubinstein,

September 2, 2003

[http://l.academicdirect.org/Horticulture/GAs/Refs/Boer&others\_2005.pdf](https://eur01.safelinks.protection.outlook.com/?url=http%3A%2F%2Fl.academicdirect.org%2FHorticulture%2FGAs%2FRefs%2FBoer%26others_2005.pdf&data=05%7C01%7Ctal.oved%40campus.technion.ac.il%7C9ebdc6799a56431902a908db3c524096%7Cf1502c4cee2e411c9715c855f6753b84%7C1%7C0%7C638170096669480721%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=GbWsgQbGzdc8sgbghJUu8FvCXN501YW4sT9vSwr9PNg%3D&reserved=0)

Simulation and the monte Carlo method

Reuven Y. Rubinstein, Dirk P. Kroese from WILEY-INTERSCINECE

[Monte Carlo Book.pdf (dropbox.com)](https://www.dropbox.com/s/4w81jahy9x2d25v/Monte%20Carlo%20Book.pdf?dl=0)