הטכניון - מכון טכנולוגי לישראל
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY
הפקולטה להנדסת חשמל
המעבדה לבקרה רובוטיקה ולמידה חישובית

Control Robotics and Machine Learning Laboratory

NP-hard problem of team composition optimization

Tal Oved and Elad Mandinach

Supervisor: Dr. Daniel Sigalov

Spring 2023

Table of Contents

Figures

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 solutions – 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
- Generation – iterations or rounds of the optimization process in CE

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).

unfeasible.

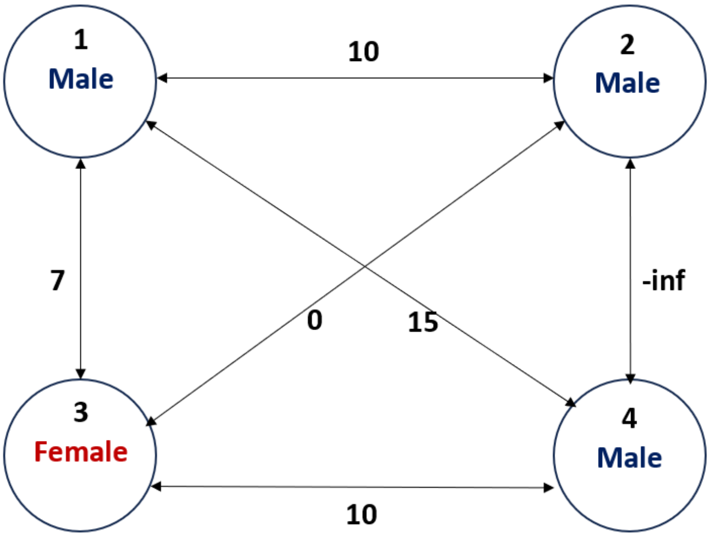

Figure 1: Fully connected Graph

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

Figure2 : 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 $M_{i,j}$ for all $i \neq j$ represents the suitability between athlete i and athlete j, when:

$$M_{i,j} = M_{j,i}\ \forall i \neq j \quad , \qquad M_{i,i} = 0\ \forall i$$

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:

$$Price = \sum_{paris} M_{i,j} + \sum_{triples} \frac{M_{i,j} + M_{i,k} + M_{j,k}}{2} + \sum_{fours} \frac{M_{i,j} + M_{i,k} + M_{j,k} + M_{i,l} + M_{j,k} + M_{k,l}}{3}$$

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.

Before running the algorithm, we need to initialize the following parameters:

1. Initial temperature
2. Final temperature

3. Temperature reduction rate
4. Probabilities for creating a new team or making a double change in an iteration

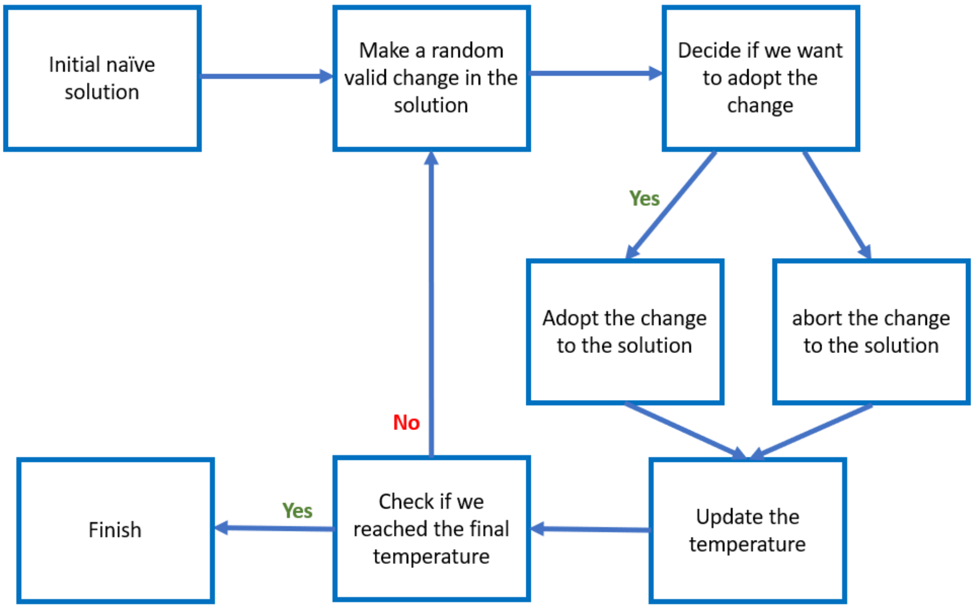

Figure 3: Simulated annealing algorithm scheme

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, $P_{i,j}$ 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 $P_{i,j}$. 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.

The formula for calculating the probabilities of the next generation is as follows:

$$p_{ij_{new}} = \alpha \cdot p_{ij_{old}} + (1 - \alpha) \cdot p_{ij_{Elite}}$$

Here, $p_{ij_{Elite}} = \frac{N_{ij}}{\sum_k N_{ik}}$, where N_{ij} represents the number of times nodes i and j were in the same group in the elite solutions. The parameter 'α' is used for adjusting this calculation.

Before running the algorithm, we need to initialize the following parameters:

1. Number of solutions in each generation
2. Precision of the elite solution from all the solutions
3. Number of generations
4. Step size ('α') towards the last generation's result
5. Probability to stop attempting to combine a team in team building.

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".

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 reflect the complexity associated with evaluating the cost function and checking the legality of moves.

Result Analysis

To evaluate the effectiveness of Simulated Annealing in optimizing team compositions, we conducted extensive experiments and analyzed the results.

For the experiment we used the following connections matrix (13x13):

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	12	2	2	2	2	2	2	2	2	2	2	2
2	12	2	12	-Inf	2	2	2	2	2	2	2	9	2
3	2	12	2	12	2	2	2	2	2	2	11	2	2
4	2	-Inf	12	2	2	2	2	2	2	2	2	2	2
5	2	2	2	2	2	12	2	2	2	2	2	2	2
6	2	2	2	2	12	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	2	12	12	2	2	2	2
8	2	2	2	2	2	2	12	2	12	2	2	2	2
9	2	2	2	2	2	2	12	12	2	2	2	2	2
10	2	2	2	2	2	2	2	2	2	2	17	17	17
11	2	2	11	2	2	2	2	2	2	17	2	17	17
12	2	9	2	2	2	2	2	2	2	17	17	2	17
13	2	2	2	2	2	2	2	2	2	17	17	17	2

Figure 4: Experiment's connection matrix

With the following genders:

```
['M'  'M'  'M'  'M'  'F'  'F'  'F'  'F'  'F'  'F'  'F'  'F'  'M']
```

Figure 5: Experiment's team member's gender

Using the following parameters:

```
params.initial_temprature = 1e6;
params.final_temprature = 1;
params.pace = 0.995;
params.team_size = size(graph,1);
params.probability_to_rand_group = 0.1;
params.probability_to_double_change = 0.3;
```

Figure 6: SA parameters

Our primary focus was on the following aspects:

Convergence

We monitored the algorithm's performance improvement over iterations. As shown in Figure 7 below, the temperature decreased, indicating that the algorithm gradually converged towards a more optimal solution, demonstrating the effectiveness of the annealing process.

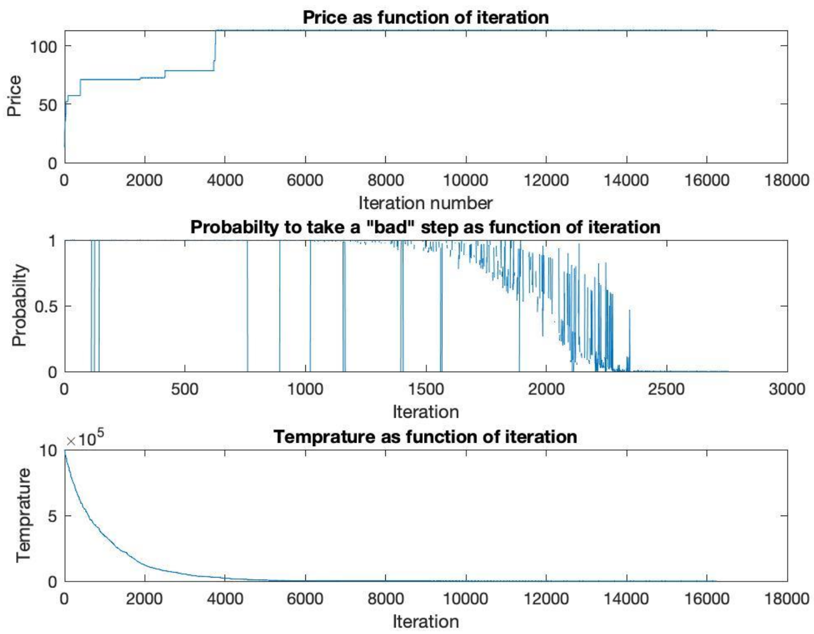

Figure 7: SA parameters and price as function of iteration

Final Result

We generated a result graph that depicted the suitability (price) of team compositions as a function of iteration, Figure 6. This graph allowed us to visualize how the algorithm's performance evolved over time, while each color represents a different team. 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 8, when each color represents different group, so we can see the partition of the groups for each iteration.

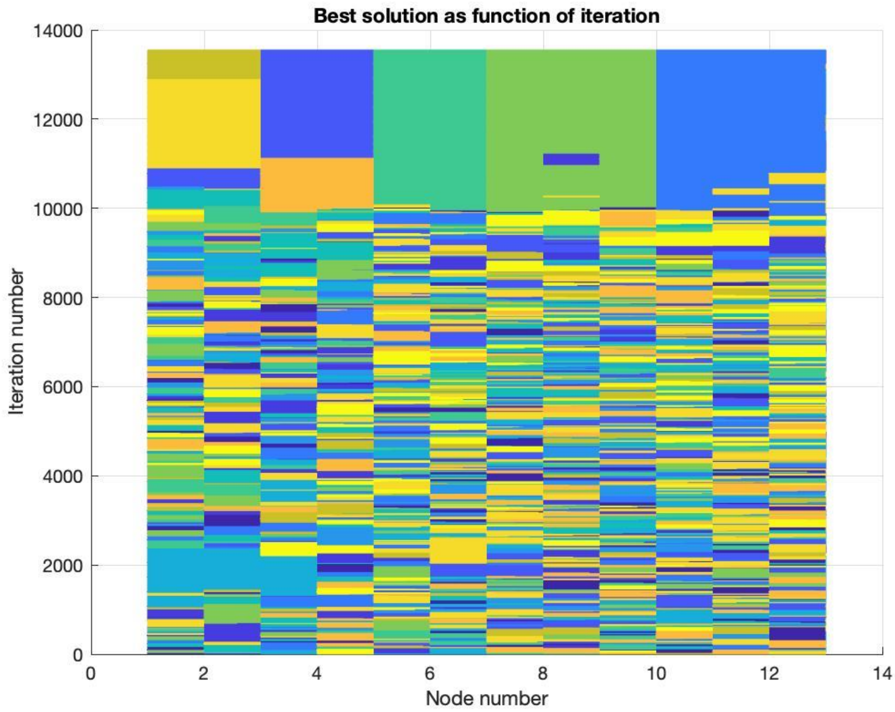

Figure 8 : the different groups as function of iteration

In the following graph, the gender represented by color and the number represent the ID of the person. Additionally, we can observe the level of suitability between different individuals within each group.

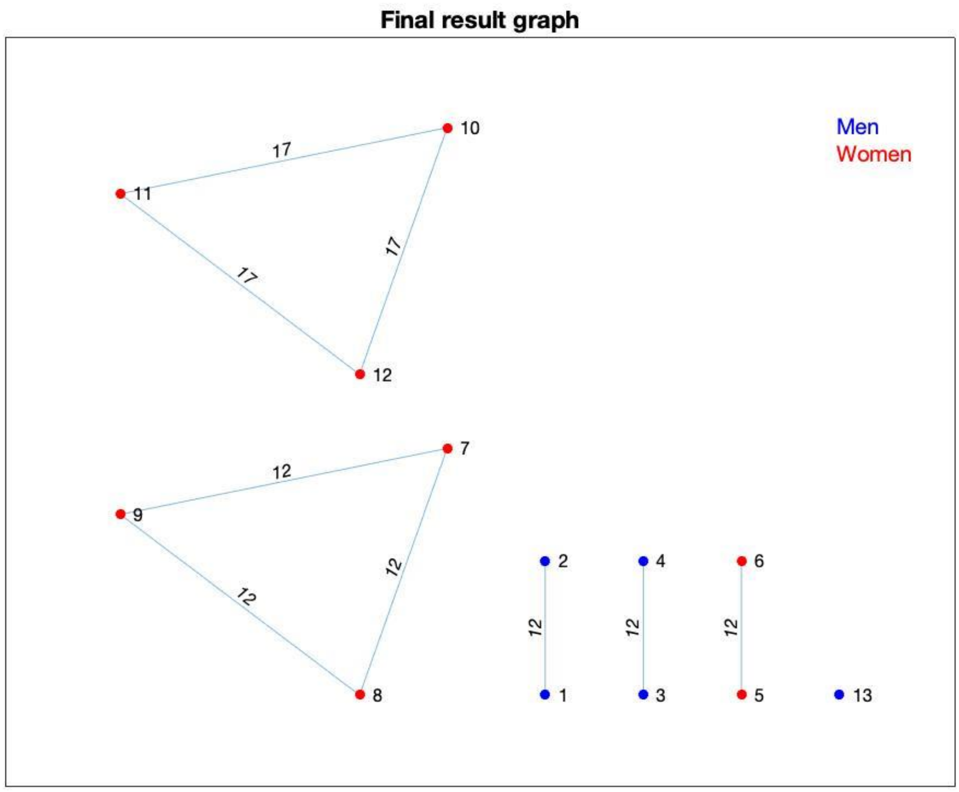

Figure 9: SA final Result Graph (13 individuals)

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 5, 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 5.
SA, with its controllable temperature and iterative refinement, can greatly benefit coaches and sports professionals seeking to enhance team performance through optimized compositions.

Cross Entropy:

Cross Entropy (CE) is a commonly used method for finding optimal solutions in classification problems and evaluating metrics. As mentioned earlier, this method employs probabilities to classify teams for each node. However, in our case, where we have more than two teams, it becomes significantly more challenging to create a solution based on this method.

Summary of Cross -Entropy implementation
Our implementation of the CE method began by initially forming teams randomly, with an equal chance of pairing legal nodes together. We then evaluated these initial solutions and updated the probabilities of pairing nodes based on the performance of elite solutions from the previous generation.

Implementing this technique proved to be exceptionally challenging due to the nature of our problem, which involves more than two groups and necessitates additional constraints on these groups. As a result, we chose a regressive approach in which we add or remove group member at each step until we have completed the composition of the group.

Time running complexity analysis

The complexity of the CE method depends on several factors, including problem size, the number of iterations, and the probability of halting attempts to form specific teams.

Typically, the CE method exhibits a time complexity of $O(N \cdot (I+C))$, where N represents the number of iterations, I reflects the complexity associated with evaluating the cost function and checking the legality of moves, and C reflects the complexity associated with assigning nodes to teams.

In our scenario, assembling teams can have an indefinite expectation, especially when only unmatched nodes remain. As iterations progress, the likelihood of pairing these unmatched nodes approaches zero. To address this, we introduced a percentage-based stopping criterion for each team assembly loop, ensuring that C remains a finite number on average.

Through our implementation, we successfully achieved a solution with polynomial complexity.

CE Results

Using the same example as we used for SA solution, and the following parameters:

```
params.num_of_lotteries_per_iteration = 350;
params.team_size = size(graph,1);
params.perc_best_solutions = 0.01;
params.max_iteration = 200;
params.alpha = 0.1;
params.probabilty_to_stop_build_team = 0.01;
```

Figure 10: CE parameters

We got the results that are illustrated in figures 8, 9 and 10.

Having an average of final result that is almost right, but completely right just once in 3 Runnings in average.

Result analysis

Experiencing significantly worse running times and less accurate solutions with our chosen regressive approach, we came to realize that this method incurred a high computational cost, primarily due to the demanding team-building conditions and the presence of more than two groups in our problem.

Throughout our experiments, we observed that in the initial iterations, there was a notable improvement in the average price, and over many iterations, the probabilities of matching between two nodes converged as expected. However, the maximal average price didn't converge and fluctuated around the optimal solution.

These results, heavily influenced by our team assembly method (i.e., the regressive approach), led us to the conclusion that the CE method was not well-suited for our problem. It couldn't provide a straightforward solution for assembling teams.

To illustrate the solution, we employed two different graphs. The first graph, shown in Figure 11, displays the average price of each generation. The second graph depicts the suitability of each member with other members as a function of the iteration number.

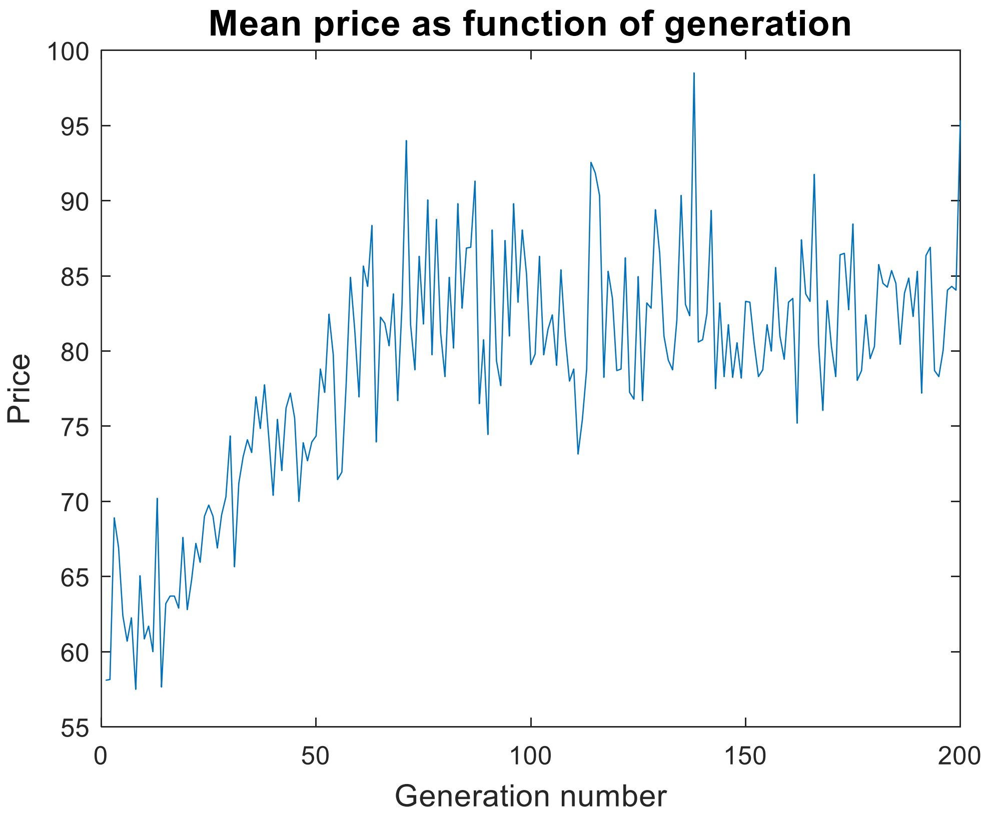

Figure 11: Mean price as function of generation

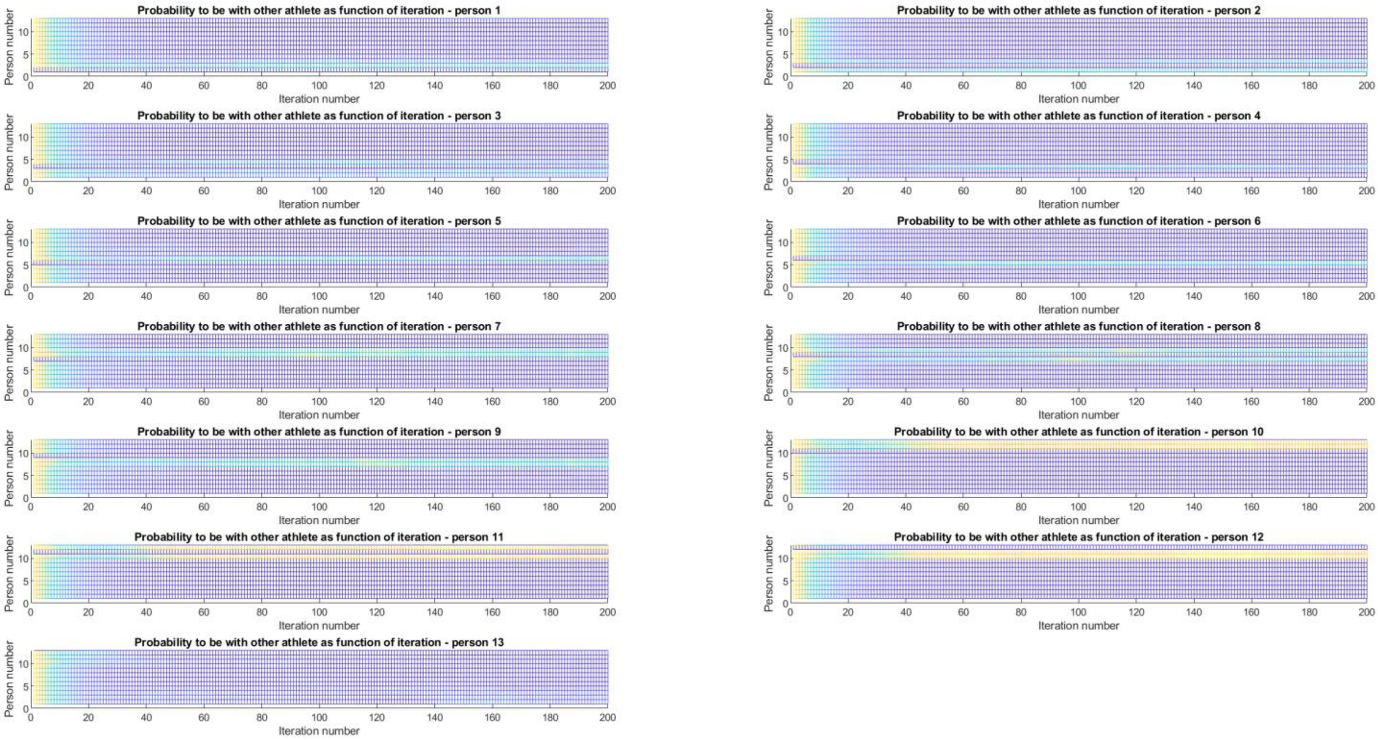

Figure12 : Suitability as function of iteration

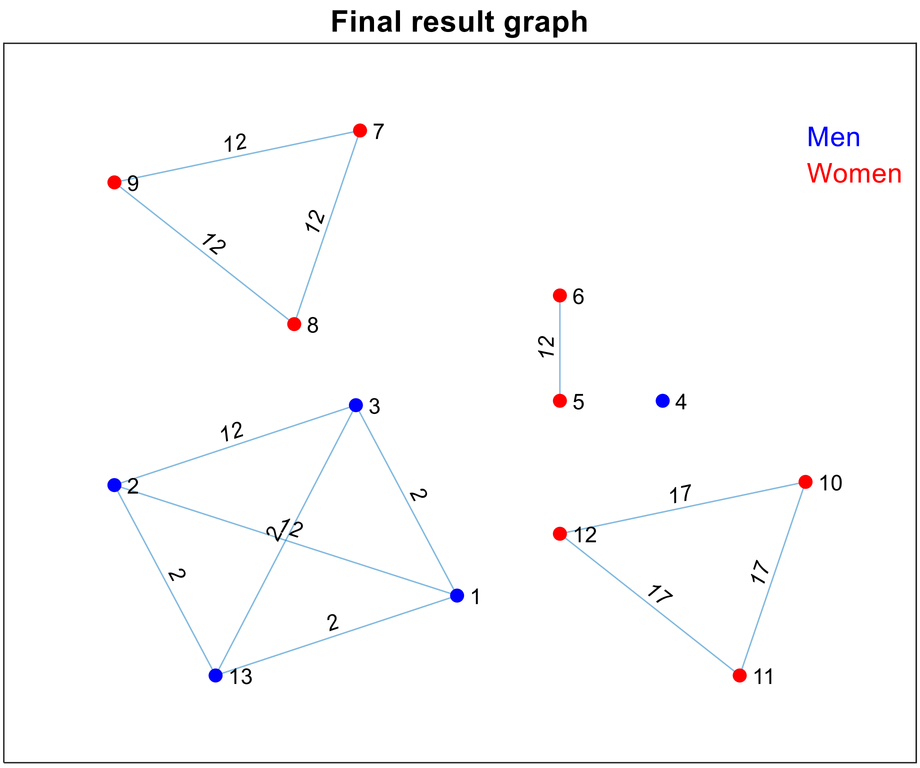

Figure 13: Result of CE on Example 1

Based on these findings, we began contemplating a mixed solution. This new approach involved initiating the process with the CE method, which initially converged rapidly, and then transitioning to the SA method to achieve a highly accurate solution.

SA and CE comparison:

To compare the two algorithms, we executed the same test with the previously mentioned parameters on the same computer, resulting in the following runtimes:

$$SA - 4.78[Sec]$$
$$CE - 18.40[Sec]$$

Clearly, the SA algorithm outperforms the CE algorithm, exhibiting a 384% improvement in terms of runtime efficiency. Additionally, we observed that the SA algorithm yields a higher success rate in terms of accurate solutions. In our analyzed problem, the SA success rate was 90%, whereas the CE's success rate was only 33%. This observation supports our claim that the SA method is better suited for our problem.

Combined solution:

Our combined solution leverages the strengths of both Cross Entropy (CE) and Simulated Annealing (SA) to create a powerful algorithm for team composition optimization. In this section, we provide a summary of our revised combined solution's implementation, analyze its time complexity, and discuss the results we obtained through extensive experimentation.

Summary of Revised Combined Solution Implementation

In our combined solution, we begin by utilizing Cross Entropy to generate an initial solution. This initial solution is then passed on to the Simulated Annealing algorithm for further refinement.

Time Running Complexity Analysis

The time complexity of our revised combined solution is influenced by the number of CE iterations required to generate the initial solution and the subsequent SA iterations for refinement, along with the complexity associated with evaluating the cost function and checking the legality of moves. In general, our revised combined solution exhibits a time complexity of $O((N_{CE} + N_{SA}) \cdot I)$, where N_{CE} represents the number of CE iterations, N_{SA} represents the number of SA iterations, and I reflects the complexity associated with cost evaluation and legality checks.

Final Result

We generated a result graph that depicted the suitability (price) of team compositions as a function of Cross-Entropy iterations used to generate the initial solution for the simulated annealing, as shown in Figure 12. It appears that, based on the graph below, after 50 iterations, we achieve a relatively high price.

Figure 14: Cross Entropy stage in the combined solution

Feeding the following solution to SA stage:

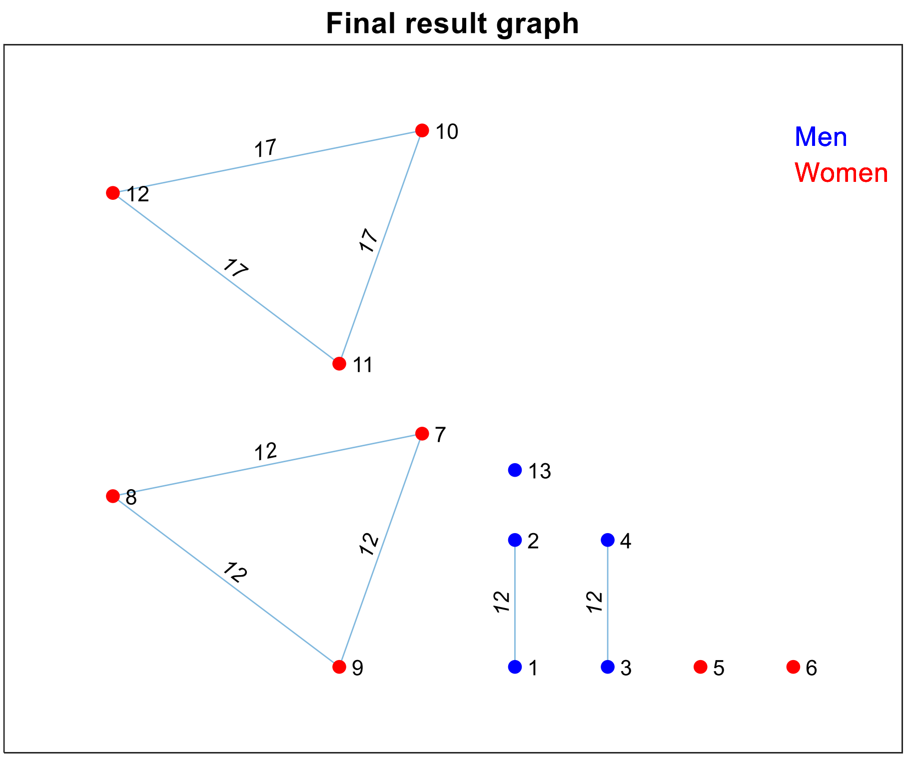

Figure15 : Initial solution for SA stage

As illustrated in Figure 13 below, when comparing the Simulated Annealing (SA) part, it becomes evident that introducing the advanced solution rather than the naïve one leads to a significant reduction in convergence time. The combined solution exhibits convergence at the 1700th iteration (in addition to the 50 iterations of the CE), whereas the SA algorithm necessitated almost double the iterations, converging only after 4000 iterations.

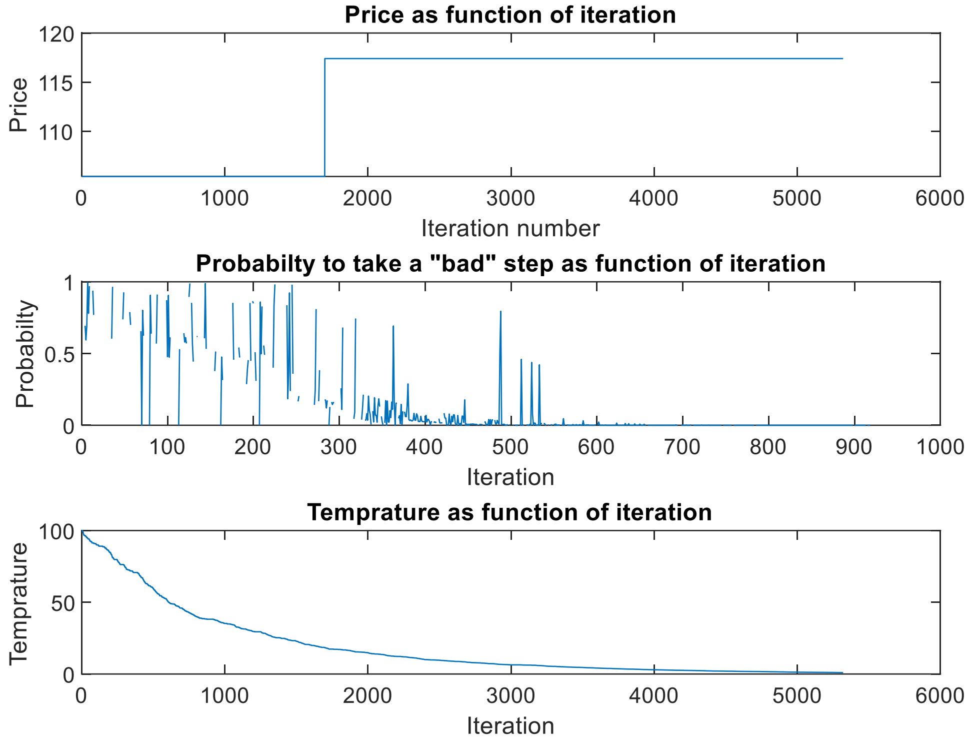

Figure 16: SA parameters as function of iteration for combines solution

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

Simulation and the monte Carlo method

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

Monte Carlo Book.pdf (dropbox.com)