IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 12, 2019, accepted May 9, 2019, date of publication May 29, 2019, date of current version June 25, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2919447*

# Team Composition in PES2018 Using Submodular Function Optimization

**YIFENG ZENG[1,2], GAOYANG SHEN[1], BILIAN CHEN[1], AND JING TANG[2]**
[1]Department of Automation, Xiamen University, Xiamen 361005, China
[2]School of Computing, Teesside University, Middlesbrough TS1 3BA, U.K.

Corresponding authors: Yifeng Zeng (yfzeng@xmu.edu.cn) and Bilian Chen (blchen@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61836005, Grant 61772442, and Grant 11671335.

**ABSTRACT** With the development of computer game technologies, gameplay becomes very realistic in many sports games, therefore providing appealing play experience to game players. To get the victory in a football pitch, the team composition is pretty important. There is little research on the automatic team composition in sports games particularly in a popular game of *Pro Evolution Soccer* (PES). In this paper, we consider the team composition as one team player recommendation problem since a team is composed of several players in a game. Subsequently, we aim to recommend a list of sufficiently good football players to game players. We convert the team player recommendation into one optimization problem and resort to the greedy algorithm-based solutions. We propose a coverage function that quantifies the degree of soccer skills to be covered by the selected players. In addition, we prove the submodularity of the coverage function and improve a greedy algorithm to solve the function optimization problem. We demonstrate the performance of our techniques in PES2018.

**INDEX TERMS** Team composition, recommender, submodularity, PES2018.

## I. INTRODUCTION

In the recent years, many sport games have appeared and attracted more and more players in game markets. Pro Evolution Soccer 2018 (PES2018)[1] is a popular football game which is produced and released by Konami,[2] it can be played on a personal computer, PS4 or XBOX. This game can be controlled by human or computer players, and can fully simulate a football match. In most cases, a human-player is offered an opportunity to compose a team of avatars each of which simulates a real-world football player, e.g. Lionel Messi, Harry Kane, *etc.*, in a competitive game. Subsequently the selection of team members becomes interesting and important in PES.

Currently the team composition mainly depends on preferences and knowledge of a human-player who, however, still expects inputs from the gaming system. In other words, the human-player would be better satisfied if the game could recommend a dream team that will succeed in a new match in PES. This is well aligned with entertainment spirit in the content recommendation in computer games [1]. Hence a team recommender becomes an important feature in a sport game not just limited in PES [2]. In PES, every football player is specified by a set of attributes, e.g. *attacking_prowess*, *ball_control*, *speed* and others, that represent his skills in a football match. Fig. 2 and Fig. 1 are screenshots of the PES game. Each attribute is associated with a specific value all of which decide the player performance in a match. The strength of a team is mainly influenced by the performance of individual players. The team is more likely to win a match if more skillful players are selected into the team. However, as each player has a specific position and a limited number of positions (a football match needs 11 players) exist in a pitch, the team composition is not straightforward given the known ratings of the players that indicate their performance. Things become more complicated since a human-player is often given a limited budget for purchasing a team of players each of which costs a certain value corresponding to his skills.

In this article, we aim to automate a team composition so that a game player can best the winning chance in the competitive PES. As a team is composed of a set of eleven football players, the core issue is about selecting the players into the team given their skillset. Subsequently we can convert the team composition into a player recommendation problem.

The associate editor coordinating the review of this manuscript and approving it for publication was Shuai Han.

[1]https://www.konami.com/wepes/2018/
[2]https://www.konami.com/


**FIGURE 1.** Football game interface of PES2018.

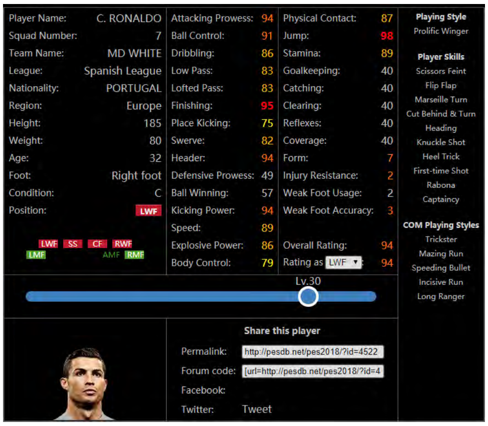



**FIGURE 2.** Player attributes in PES2018.

In other words, the recommended players will compose the team to be offered to a game player in PES. We will develop a new player recommendation method and demonstrate its utility in PES. We formulate the player recommendation as one optimization problem with constraints. In the case of ensuring that the player's position is appropriate in a football pitch, we develop an objective function in the optimization that represents the team's coverage of each skill in a match. Meanwhile, we consider the player's total salary as a cost constraint in the optimization. Hence, by solving the optimization problem, we can select a set of eleven players to compose a team so that their coverage of skills is to be maximized, which will provide a higher winning rate in a football match in PES.

Solving the constrained optimization problem for the player recommendation is hard since the objective function is not a linear combination of skill factors from all potential players. We make a further step to investigate the property of the objective function and prove that it is a submodular function since the property of diminishing marginal returns is satisfied in the function [3]. Subsequently we resort to a traditional greedy algorithm [4] to solve the optimization problem and improve the algorithm, namely Cost-Effective Forward selection Greedy (CEFG), to achieve better results.

The cost constraint is not fully exploited in the generalized greedy algorithm due to the limit of the number of players in a team. Hence we propose CEFG algorithm that combines the unit-cost greedy algorithm (ignoring the costs) and the traditional greedy algorithm. We find a middle point and use two different strategies in-between. Finally we implement the proposed algorithm in the PES simulation platform, and show convincing results in the experiments.

The remainder of this article is organized as follows. Section II describes related works of team composition and player recommendation, and then we brief a submodular function and its optimization in Section III. Section IV proves the property of the objective function regarding the submodularity. Section V develops a greedy algorithm and its improved version to solve the optimization problem. We conduct experiments to evaluate performance of our techniques in Section VI. Finally, Section VII concludes our work.

## II. RELATED WORK

Research of team composition is most relevant to team recommendation where a list of teams are recommended. Originally team recommendation comes from organizational and behavioral sciences and research on social web application has appeared for a team recommendation since 2012 [5]. The team recommendation was rarely studied in a game environment. Brocco and Woerndl [2] presented ideas for supporting team composition in different computer game scenarios, and explained how to integrate locations in their team composition model.

Most of the previous work on team recommendation was based on a model where attributes of individual group members are aggregated to generate group recommendation [6]. Some authors proposed an algorithm where individual user ratings were generated using nearest neighbor models for collaborative filtering [7] while others created a group profile by aggregating the profiles of individual users and used neighborhood models to generate recommendations for the newly created group profile [8].

Amer-Yahia *et al.* [9] advocated that a better team recommendation strategy could be devised by considering disagreements between the individual users in the group for the same item. Li *et al.* [10] studied the issue of recommending a replacer when a critical player becomes unavailable. Their basic ideas are to adopt a graph kernel to encode both skill and structural matching. Other techniques intend to find the best team in terms of communication costs within a network of experts [11]. In parallel, a team recommendation is studied as a special case of the budgeted social choice problem in economics. Lu and Boutilier [12] proposed a greedy algorithm using knapsack heuristics; however, the algorithm does not perform quite well in a group recommendation task with positional scoring rules. Skowron *et al.* [13] extended the theoretical framework for limited choice models with positional scoring rules using the ordered weighted average operators.

With a rapid development of mobile internet and online marketing models, many spatial crowdsourcing

**TABLE 1. Players attributes and possible values for the attributes.**

| | ID | 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|---|---|
| | player_name | C.RONALDO | L.MESSI | L.SUAREZ | M.NEUER | ... |
| | position | LWF | RWF | CF | GK | ... |
| | rating | 94 | 94 | 92 | 91 | ... |
| ability | attacking_prowess | 94 | 95 | 95 | 42 | ... |
| | ball_control | 91 | 96 | 86 | 68 | ... |
| | dribbling | 86 | 96 | 84 | 60 | ... |
| | low_pass | 83 | 88 | 82 | 65 | ... |
| | lofted_pass | 83 | 86 | 77 | 69 | ... |
| | finishing | 95 | 95 | 95 | 43 | ... |
| | header | 94 | 68 | 77 | 70 | ... |
| | defensive_prowess | 49 | 43 | 58 | 60 | ... |
| | speed | 89 | 86 | 78 | 71 | ... |
| | goalkeeping | 40 | 40 | 40 | 98 | ... |

platforms emerge. The problem of team formation for crowdsourcing becomes popular ascribed to requirements of massive human intelligence service-oriented applications. Some studies focused on crowdsourcing complex tasks through team formation in non-cooperative social networks [14], [15] while others developed a top-$k$ team recommendation in a spatial crowdsourcing problem [16]. We also notice that a team recommendation problem can be modeled as a submodular optimization problem. Parambath *et al*. [17] proposed a unified framework and an algorithm for the group recommendation where a fixed number of items or alternatives could be recommended to a group of users. They used a fast greedy algorithm with strong theoretical guarantee.

Most of the work on team/group recommendation mainly focuses on the improvement of service quality to satisfy a diverse set of preferences from a group of users who have different requirements. Our work in this article is to choose a set of players so that a comprehensive set of skills will be covered therefore leading to a successful match in PES.

## III. BACKGROUND: SUBMODULAR FUNCTION

For a set of objects $V = \{v_1, \ldots, v_n\}$ and a function $f : 2^V \rightarrow R$, if for each $A \subseteq B \subseteq V$ and $e \in V \setminus B$, it holds that $\triangle(e \mid A) \geq \triangle(e \mid B)$, then the function $f$ is submodular, where

$$\triangle(e \mid A) = f(A \cup e) - f(A) \quad (1)$$

means the discrete derivative of $f$ at $A$.

Equivalently, $f$ is submodular for each $A, B \subseteq V$, it holds that $f(A)+f(B) \geq f(A\cap B)+f(A\cup B)$. One important property of submodularity is diminishing marginal returns, i.e., adding an element to a small set is more influential than adding it to a large set.

A function $f$ is said to be monotone if $f(A) \leq f(B)$ for all $A \subseteq B \subseteq V$. There is a popular submodular function optimization problem: given an integer $k$, we aim to find a subset $T \subseteq V$ to maximize the monotone submodular function $f$, i.e., $argmax_{T\subseteq V} f(T)$, where $|T| \leq k$. A solution to the optimization problem is NP-hard [18]. A greedy algorithm can find an approximate solution that guarantees the solution quality within $\frac{e-1}{e}$ ($\approx$ 0.632) of the optimality [3]. Going beyond the $\frac{e-1}{e}$-approximation is NP-hard for many classes of submodular functions [3], [19].

In recent years, the submodular function optimization has been seen in many machine learning and computer vision applications and is usually applied to coverage issues such as video segmentation [20], [21], document summarization [22], advertisement allocation [23], and information gathering [24]. In this article, we will convert the team recommendation problem into a submodular function optimization with the constraint of salary cost, and seek for greedy algorithm based solutions to the problem.

## IV. PLAYER RECOMMENDATION AS A SUBMODULAR FUNCTION OPTIMIZATION

In this section, we formulate player recommendation into one optimization problem and prove the submodularity property of this function as well.

### A. SKILL COVERAGE FUNCTION

A team composition has a large influence on a match result since it decides the strength and complementarity of members' skills on a football pitch. To maximize the winning chance, we need to recommend a team of players that will cover a set of football skills.

Given the PES platform, we choose ten players' attributes as the most important skills for the team composition, i.e. *attacking_ prowess*, *ball_control*, *dribbling*, *low_pass*, *lofted_pass*, *finishing*, *header*, *defensive_prowess*, *speed*, and *goalkeeping*. In addition, we consider the player's number, name, position, salary and overall rating. Hence each player has 15 attributes. A sample of some player's attributes is shown in Table 1.

For each player $p_i \in U$, where $U = \{p_1, p_2, \ldots, p_n\}$ is a collection of players, we use $s$ to represent the player's ability such as *attacking_prowess*, *ball_control*, and *speed*, $s_j \in S = \{s_1, s_2, \ldots, s_m\}$ where $m$ is equal to 10 if ten attributes are considered in our work. We define the skill value of each player as $a_{s_j}(p_i)$, and the skill coverage function for a player, that is, the degree to which the player $p_i$ covers the set of skill $s_j$ is defined in Eq. 2.

$$cov_{s_j}(p_i) = a_{s_j}(p_i)/(\sum_{p_k \in U} a_{s_j}(p_k)) \tag{2}$$

Subsequently, we can define the skill coverage function for a set of team players, $T \subseteq U$, that is a subset of all potential players. Eq. 3 measures the degree to which the ability $s_j$ is covered by at least one player in $T$.

$$cov_{s_j}(T) = 1 - \prod_{p_i \in T}(1 - cov_{s_j}(p_i)) \tag{3}$$

Finally, the function of $T$ covering $S$ can be defined as $F(T)$ in Eq. 4.

$$F(T) = \sum_{s_j \in S} \beta cov_{s_j}(T) \tag{4}$$

where $\beta$ is used to weight the skill $s_j$.

### B. RECOMMENDER MODEL

We aim to find an optimal team that maximizes the coverage value in Eq. 4. Meanwhile, we need to consider the cost of composing the team of players in the optimization. Hence the recommendation is equivalent to solving the following optimization problem.

$$max_{T \in U} F(T) \ subject\ to\ |T| = 11\ and\ c(T) \leq C \tag{5}$$

where $c(T)$ is the sum of the salary of the total eleven players in $T$ and $C$ is the salary constraint for the entire team. The salary value is to be specified in a sport game; otherwise, as shown in our experiments, we can use the available players' ratings to estimate their salary.

Solving the above optimization problem sounds to be not easy and we proceed to investigate the submodularity of the skill coverage function below.

*Proposition 1: The monotone function $F(T)$ (in Eq. 4) is submodular.*

*proof:* We calculate the marginal gain of the skill coverage when one player is added into a potential team $\hat{T} \subseteq U$.

$$cov(\hat{T} \cup p_j) - cov(\hat{T})(1 - \prod_{p_i \in \hat{T}}(1 - cov(p_i)) * (1 - cov(p_j)))$$
$$-(1 - \prod_{p_i \in \hat{T}}(1 - cov(p_i)))$$
$$= cov(p_j) * \prod_{p_i \in \hat{T}}(1 - cov(p_i))$$

Similarly, for a small team $\check{T}$, we have

$$cov(\check{T} \cup p_j) - cov(\check{T}) = cov(p_j) * \prod_{p_i \in \check{T}}(1 - cov(p_i)),$$

where $\check{T} \subseteq \hat{T} \subseteq U$. Moreover, since $1 - cov(p_i) < 1$, $cov(\hat{T} \cup p_j) - cov(\hat{T}) \leq cov(\check{T} \cup p_j) - cov(\check{T})$, we have $cov(T)$ is submodular.

There is an important attribute of submodularity: if $g_1, \ldots, g_n : 2^V \rightarrow R$ are submodular, and $\alpha_1, \ldots \alpha_n \geq 0$, then $f(T) := \sum_{i=1}^{n} \alpha_i g_i(S)$ is submodular as well [25]. Hence $F(T)$ is submodular as $cov(T)$ is submodular.

Consequently, Eq. 5 becomes a maximum budget coverage problem with a monotonic cost constraint. The player recommendation formulated as the submodular function optimization is a NP-hard problem [18] and an approximate solution is to be investigated next.

## V. OPTIMIZATION ALGORITHMS

In this section, we introduce a greedy algorithm to solve the recommendation problem that is formulated as one submodular function optimization problem in Eq. 5, and improve the algorithm to solve the problem.

### A. GREEDY ALGORITHM

As mentioned in Section III, the greedy algorithm generally can solve the submodular function optimization problem. The solution reaches the approximation of optimality with the theoretical bound $F(\mathcal{T}) \geq (1 - 1/e) max\ F(T)$ [3]. Zhang and Vorobeychik have recently investigated it with a monotonic cost constraint [26] and propose the generalized greedy algorithm as shown below.

Generalized Greedy Algorithm

**Input:** an objective function $F$, a cost constraint $C$, and player database $U$

**Output:** a solution $T \subseteq U$ with $c(T) \leq C$

1: $T \longleftarrow \emptyset$;
2: **repeat**
3: $\quad p \longleftarrow argmax_{p \in U} \frac{F(T \cup p) - F(T)}{c(T \cup p) - c(T)}$
4: $\quad$ **if** $c(T \cup p) \leq C$ **then** $T = T \cup p$
5: $\quad$ **end if**
6: $\quad U = U \setminus p$
7: **until** $U = 0$
8: **return** $T$

The algorithm iteratively selects a player $p$ such that the ratio of the marginal gain for objective function $F$ and constraint $c$ is maximized by adding $p$ (lines 3-5). The best subset $T$ found is eventually returned.

As in our recommendation problem there are eleven players in a football team, the length of $T$ needs to be limited. In addition, $c(T \cup p) - c(T) = c(p)$ as the constraint is linear and discrete. Hence we adapt the generalized greedy algorithm into Limit Greedy Algorithm below.

In each iteration, we will select the player $p$ from a set of players $U$ with the largest ratio of the increase of the objective function to the wage cost under the cost constraint $C$ (lines 3-5), until the team length is equal to eleven.

### B. CEFG ALGORITHM

Due to the limit of the number of players in a team, the cost constraint is not fully exploited in the generalized greedy algorithm, which leads to a small cost of the selected team and the overall team rating is extremely low. The players recommended are cost-effective; however, the team of such

Limit Greedy Algorithm

**Input:** a submodular objective function $F$, a cost constraint $C$, and player database $U$

**Output:** a solution $T \subseteq U$ with $c(T) \leq C$ and $|T| = 11$

1: $T \longleftarrow \emptyset$;
2: **repeat**
3: $\quad p \longleftarrow argmax_{p \in U} \frac{F(T \cup p) - F(T)}{c(p)}$
4: $\quad$ **if** $c(T \cup p) \leq C$ **then** $T = T \cup p$
5: $\quad$ **end if**
6: $\quad U = U \setminus p$
7: **until** $|T| = 11$
8: **return** $T$

CEFG Algorithm

**Input:** a submodular objective function $F$, a cost constraint $C$, and player database $U$

**Output:** a solution $T \subseteq U$ with $c(T) \leq C$ and $|T| = 11$

1: $T \longleftarrow \emptyset$;
2: **repeat**
3: $\quad p \longleftarrow argmax_{p \in U} F(T \cup p) - F(T)$
4: $\quad$ **if** $c(T \cup p) \leq C$ **then** $T = T \cup p$
5: $\quad$ **end if**
6: $\quad U = U \setminus p$
7: $\quad$ **if** $C - c(T \cup p) < \varepsilon$ **then**
8: $\qquad$ **repeat**
9: $\qquad\quad p \longleftarrow argmax_{p \in U} \frac{F(T \cup p) - F(T)}{c(p)}$
10: $\qquad\quad$ **if** $c(T \cup p) \leq C$ **then** $T = T \cup p$
11: $\qquad\quad$ **end if**
12: $\qquad\quad U = U \setminus p$
13: $\qquad$ **until** $|T| = 11$
14: $\quad$ **end if**
15: **until** $|T| = 11$
16: **return** $T$

players is not strong enough to win a match. The results of the simulated competition in Section VI will illustrate this problem.

Inspired by the classification selection in [27], we first use the unit-cost greedy algorithm (ignoring the costs) in the early player selection. When the total cost is close to the upper-bound constraint, we adopt the Limit Greedy Algorithm. We find a middle point and use two different strategies in-between. The new approach is framed as the CEFG (Cost-Effective Forward selection Greedy) Algorithm.

Give the submodular coverage function $F$, a set of players $U$ and a salary cost constraint $C$, we first use the unit-cost greedy algorithm to select the player $p$ with the maximum increment of the objective function (lines 3-5), which means the best player is added to the team $T$. Hence, we can make the most of the cost space and choose the player who is outstanding enough in the initial selection stage. We will not select a player twice in each iteration (line 6).

If $C - c(T \cup p) < \varepsilon$ as $\varepsilon = \sum c_i$ where

$$i = \{1, 2, \ldots 11\}, \; c_i \textit{ is the lowest value in } c \tag{6}$$

**TABLE 2. Positions of players and their equivalent numbers.**

| Position | Specific position | Number |
|---|---|---|
| Goalkeeper | GK | 1 |
| Guard | CB | 2 |
| | LB | 3 |
| | RB | 4 |
| Midfielder | DMF | 5 |
| | CMF | 5 |
| | RMF | 5 |
| | LMF | 5 |
| | AMF | 5 |
| Forward | LWF | 6 |
| | RWF | 7 |
| | CF | 8 |
| | SS | 8 |

Then we enter the second selection stage and use the greedy algorithm for the consideration of the remaining cost (lines 8-14). By doing this, we have a team of players that meets the cost constraint and contains sufficiently good players, which generates better results than the generalized greedy algorithm in V-A. It is apparent that the CEFG Algorithm will degenerate into unit-cost greedy algorithm if $C$ is large enough.

## VI. EXPERIMENTS AND GAME RESULTS

We implement the algorithms in Matlab2018 and conduct all the numerical computations on a Windows PC with a 4-core Intel i7-6700 3.40GHz CPU and 16GB memory. All the games are simulated in a quick game of PES2018 that is downloaded from a platform *Steam* on Windows10 computer system.

### A. DATA ANALYSIS

We collect the match data from the official website of PES2018[3] by using a Python crawler. There are a total of 9,563 football players in the database and a sample of data is shown in Table 1.

For the position of each player on the football pitch, we consider equivalence of positions and normalize the position as shown in Table 2. Based on the PES game experience, we choose the team of 4-3-3 formations which means there are one *Goalkeeper*, four *Guard*, three *Midfielder* and three *Forward* in the team. For the position $g$ of player $p$, the recommended player's position in the team meets the formula in Eq. 7, where $n$ refers to the total number of players.

$$\begin{cases} n_p = 1, & \text{where } g_p = 1, 3, 4, 6, 7, 8 \\ n_p = 2, & \text{where } g_p = 2 \\ n_p = 3, & \text{where } g_p = 5 \end{cases} \tag{7}$$

For the cost constraint, there is no player's salary data in the official website. Considering that the player's salary is often positively correlated with his rating, we fit the wages with scores of some players based on the existing data. We find that

[3] http://pesdb.net/pes2018

**TABLE 3. The CEFG result on the selected players with their numbers in the database.**

| Player Number | 1 | 2 | 9 | 22 | 1508 | 107 | 16 | 21 | 19 | 53 | 89 | total cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | 62.6415 |

**TABLE 4. Match results of dream team v.s. random teams.**

| Battle | Score | | | | | | | | | | Win Number | Goal Difference | Actual Cost Comparison |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AMIENS | 4:1 | 4:2 | 6:0 | 1:1 | 0:2 | 3:1 | 4:1 | 4:0 | 3:0 | 2:2 | | | 62.64 |
| VS | 2:0 | 4:2 | 2:0 | 4:0 | 0:1 | 3:0 | 3:1 | 1:0 | 3:0 | 4:2 | 25.5 | 1.97 | : |
| RAND1 | 0:1 | 4:0 | 4:1 | 1:0 | 3:0 | 1:1 | 3:1 | 4:2 | 2:1 | 3:0 | | | 19.29 |
| AMIENS | 3:0 | 1:0 | 3:0 | 0:0 | 1:1 | 2:0 | 0:1 | 2:2 | 4:0 | 3:1 | | | 62.64 |
| VS | 1:1 | 1:0 | 3:0 | 1:0 | 2:0 | 1:0 | 1:1 | 3:0 | 3:0 | 1:0 | 24.5 | 1.37 | : |
| RAND2 | 3:1 | 1:0 | 0:1 | 2:0 | 4:1 | 1:1 | 1:0 | 3:0 | 2:1 | 0:0 | | | 20.75 |
| AMIENS | 1:0 | 4:1 | 1:0 | 2:2 | 2:2 | 2:0 | 2:0 | 3:1 | 3:0 | 1:1 | | | 62.64 |
| VS | 4:0 | 3:1 | 1:1 | 3:0 | 0:0 | 3:0 | 3:1 | 4:1 | 1:0 | 1:0 | 26.5 | 1.7 | : |
| RAND3 | 3:0 | 2:1 | 2:1 | 1:1 | 4:0 | 3:1 | 5:1 | 1:0 | 0:0 | 2:0 | | | 24.04 |

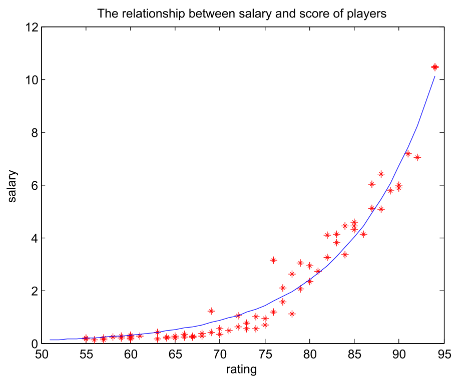



**FIGURE 3. Cost of players as a function of their ratings.**

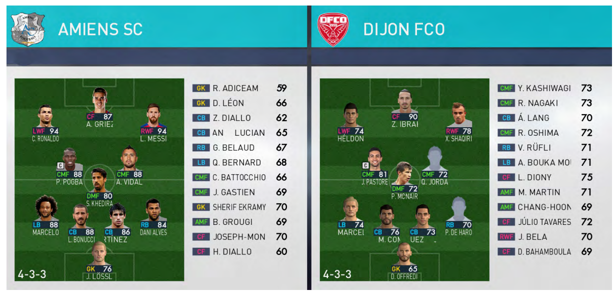



**FIGURE 4. Recommended players to compose the dream team v.s. the random team.**

the data is exponentially distributed and therefore use the least squares method for regression. The fitting curve formulated below is shown in Fig. 3.

$$y(i) = \eta \cdot e^{\theta x(i)} \tag{8}$$

where $\eta = 6.375 \times 10^{-4}$, and $\theta = 0.1029$. Then through the curve, we can find the $y$-axis of the corresponding point based on the $x$-axis, which means we can get a player's salary based on his ratings.

### B. EXPERIMENTAL RESULTS

To ensure the credibility of the results, we select a total of 8,762 players with the ratings larger than or equal to 60 in the database, and recommend a team including 11 players. We use the CEFG Algorithm to solve the optimization problem and set a sufficiently large cost as retrieved from the curve in Fig. 3, the recommended results of team formation are shown in Table 3 below.

Based on the recommended players, we compose a "Dream Team". To conduct comparison of the algorithm performance, we randomly generate a team in the PES without any cost constraint and then simulate the battle between the two teams (including the players) as shown in Fig. 4. AMIENS represents the Dream Team and DIJON represents the random team. The final result from all the five matches is 4:1 and the dream team dominates most of the competitions.

To verify the strength of the Dream Team, we get three random teams, and use the Dream Team to battle with each one for 30 matches. The random teams are randomly generated from the game PES, which limits the position of 11 players, and the ratings of players are high and low. So random teams have great reference significance. If the game ends in a tie, we set the win number to 0.5. The results are listed in Table 4 including specific results of every match, the Dream Team's wins of 30 matches and the average goal difference.

We find that the Dream Team performs pretty well against the three random teams. If we set the values of win, draw, and lose of the match to 1, 0, and -1 respectively, we can analyze the results from another perspective in Fig. 5. The $x$-axis has different random teams while the $y$-axis are the average values and variances of 30 match results. Obviously, AMIENS wins a lot and has a stable performance.

We also recommend a team using the Limit Greedy Algorithm (represented by the team MAN in the game) and have

**TABLE 5.** Results of the selected players through the limit greedy algorithm.

| Player Number | | | | | | | | | | | total cost |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8407 | 8434 | 8618 | 8500 | 8410 | 8450 | 8664 | 8602 | 8488 | 8628 | 8461 | 3.3667 |

**TABLE 6.** Match results of the teams (generated by the limit greedy algorithm) v.s. random teams.

| Battle | Score | | | | | | | | | | Win Number | Goal Difference | Actual Cost Comparison |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAN | 2:2 | 1:0 | 2:1 | 2:2 | 0:0 | 3:1 | 0:2 | 0:1 | 2:3 | 1:3 | | | 3.37 |
| VS | 0:0 | 3:0 | 1:3 | 0:3 | 0:2 | 1:1 | 0:1 | 4:0 | 0:1 | 1:3 | 11.5 | -0.33 | : |
| RAND1 | 1:4 | 2:2 | 2:2 | 0:1 | 2:2 | 1:0 | 4:1 | 0:2 | 0:0 | 1:3 | | | 19.29 |
| MAN | 1:0 | 4:1 | 0:1 | 1:0 | 2:2 | 2:2 | 3:0 | 0:0 | 0:2 | 0:1 | | | 3.37 |
| VS | 2:2 | 1:0 | 4:1 | 2:4 | 0:0 | 1:1 | 0:1 | 3:1 | 0:2 | 1:0 | 16 | 0.27 | : |
| RAND2 | 1:1 | 2:2 | 2:0 | 2:4 | 0:1 | 0:0 | 4:0 | 0:0 | 1:0 | 1:3 | | | 20.75 |
| MAN | 1:3 | 3:1 | 2:0 | 4:2 | 1:1 | 0:1 | 0:0 | 2:4 | 1:3 | 0:2 | | | 3.37 |
| VS | 0:3 | 2:0 | 0:3 | 4:0 | 0:3 | 2:4 | 0:0 | 1:4 | 0:1 | 1:0 | 12 | -0.55 | : |
| RAND3 | 2:0 | 1:3 | 1:1 | 0:1 | 4:2 | 2:2 | 0:3 | 1:1 | 1:0 | 0:2 | | | 24.04 |

**TABLE 7.** Match results of the CEFG v.s. the limit greedy algorithm.

| Battle | Score | | | | | | | | | | Win Number | Goal Difference | Cost Constraint | Actual Cost Comparison |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AMIENS1 | 5:1 | 5:0 | 1:0 | 3:0 | 5:0 | 1:0 | 5:0 | 1:0 | 3:0 | 2:0 | | | | 62.64 |
| VS | 2:0 | 4:1 | 3:0 | 1:1 | 2:0 | 1:0 | 4:1 | 3:0 | 1:0 | 4:1 | 29 | 2.43 | 70 | : |
| MAN1 | 1:0 | 2:0 | 3:0 | 1:1 | 1:0 | 4:1 | 3:1 | 3:0 | 5:0 | 2:0 | | | | 3.37 |
| AMIENS2 | 3:0 | 1:0 | 0:1 | 2:0 | 3:0 | 3:0 | 2:2 | 2:0 | 4:2 | 1:1 | | | | 44.89 |
| VS | 2:2 | 1:1 | 4:2 | 4:1 | 1:0 | 2:0 | 4:2 | 1:0 | 0:2 | 1:1 | 23.5 | 1.17 | 50 | : |
| MAN2 | 0:0 | 1:0 | 0:1 | 4:1 | 4:1 | 1:0 | 3:1 | 1:0 | 1:1 | 1:0 | | | | 3.37 |
| AMIENS3 | 3:0 | 0:1 | 0:1 | 4:1 | 3:1 | 1:0 | 3:1 | 0:0 | 2:2 | 4:2 | | | | 23.00 |
| VS | 2:0 | 4:2 | 1:0 | 3:1 | 4:2 | 2:2 | 3:0 | 0:1 | 1:0 | 1:3 | 22 | 1.03 | 30 | : |
| MAN3 | 1:1 | 1:4 | 2:0 | 1:1 | 1:0 | 4:1 | 4:0 | 1:0 | 1:1 | 2:0 | | | | 3.37 |

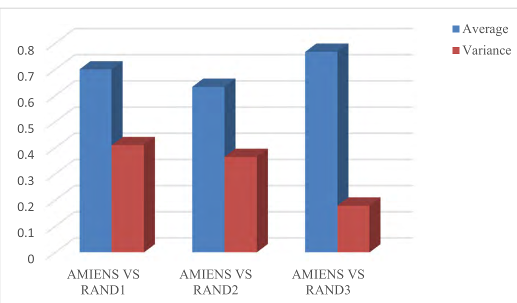



**FIGURE 5.** Normalized results of dream team v.s. random teams.

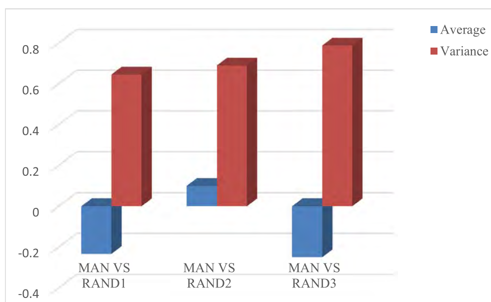



**FIGURE 6.** Normalized results of the teams (generated by the limit greedy algorithm) v.s. random teams.

the team compete with the above three random teams. The results are shown in Table 5, Table 6 and Fig. 6. We find that the teams recommended by the greedy algorithm have a poor performance, and the randomness of their performance is very large. But on the other hand, we can find that the strength of the team does not depend entirely on cost or rating (e.g. MAN VS RAND2).

Under different cost constraints in the CEFG algorithm, we recommend the Dream Teams and have battles between the teams generated by the Limit Greedy Algorithm and the CEFG algorithm. The match results are shown in Table 7 and Fig. 7. The teams recommended by the CEFG algorithm perform significantly better than those by the Limit Greedy Algorithm. We notice that a larger cost value generates better teams, which leads to more winning results for the teams.

Finally, in order to verify the superiority of the CEFG algorithm, we select MAB, MSB and HER teams in the game all of which exist in the real gameplay. We calculate the costs of the three teams, and use them as constraint to recommend

**TABLE 8. Match results of the CEFG v.s. the actual teams.**

| Battle | Score | | | | | | | | | | Win Number | Goal Difference | Cost Constraint | Actual Cost Comparison |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AMIENS4<br>VS<br>MAB | 3:1<br>0:0<br>0:2 | 1:4<br>1:0<br>1:0 | 0:0<br>2:0<br>4:1 | 2:2<br>1:1<br>2:0 | 1:3<br>2:4<br>0:0 | 3:0<br>0:0<br>1:1 | 3:1<br>4:2<br>0:2 | 1:0<br>2:2<br>2:0 | 2:1<br>2:0<br>2:2 | 0:0<br>0:1<br>3:1 | 19 | 0.47 | 36.95 | 35.21<br>:<br>36.95 |
| AMIENS5<br>VS<br>MSB | 0:0<br>2:0<br>1:0 | 2:2<br>2:2<br>2:0 | 0:1<br>0:1<br>0:0 | 3:1<br>0:1<br>3:1 | 0:0<br>1:1<br>0:0 | 4:0<br>1:0<br>0:1 | 2:2<br>0:0<br>0:2 | 1:3<br>4:2<br>2:2 | 2:2<br>2:0<br>1:0 | 1:0<br>0:0<br>1:1 | 17.5 | 0.4 | 26.38 | 23.00<br>:<br>26.38 |
| AMIENS6<br>VS<br>HER | 2:4<br>0:1<br>0:3 | 1:0<br>2:0<br>1:0 | 1:4<br>4:2<br>0:0 | 3:1<br>0:2<br>1:1 | 0:0<br>1:1<br>1:0 | 1:0<br>4:0<br>2:0 | 3:0<br>0:2<br>0:0 | 2:3<br>3:0<br>3:0 | 0:0<br>0:0<br>1:3 | 0:0<br>1:1<br>1:1 | 17 | 0.3 | 15.70 | 13.18<br>:<br>15.70 |

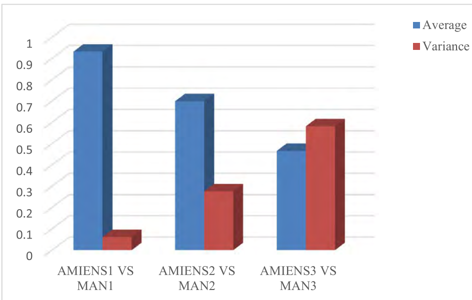



**FIGURE 7. Normalized results of the CEFG v.s. the limit greedy algorithm.**

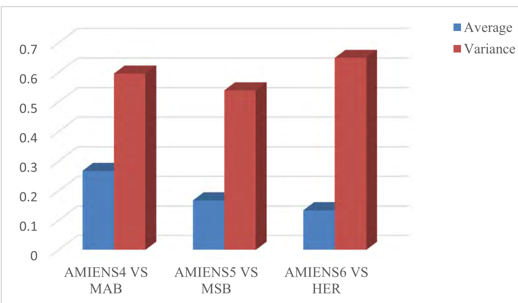



**FIGURE 8. Normalized results of the CEFG v.s. the actual teams.**

a team of players based on the CEFG algorithm. We then match the recommended team with the existing three teams and show the results in Table 8 and Fig. 8.

We can find that under the same cost constraint (or in a sense of rating), the teams recommended by CEFG algorithm are stronger than actual teams. The teams generated by the CEFG algorithm dominate the play in the football pitch.

## VII. CONCLUSION

In this paper, we make an in-depth analysis of team composition in PES that can be converted into a player recommendation problem. As there is no clear approach for football player recommendation in a game, we propose a skill coverage function to quantify the complementary capability of a proper team. We then improve the greedy algorithm to solve the recommendation problem. We conduct empirical study of the proposed recommendation techniques in a game platform of PES2018. The results demonstrate the strength of the team as well as the effectiveness of our approach. Although we investigate our techniques in the context of PES, the proposed recommendation model based on the submodular function is rather general and can be adapted to solve other team composition problems. We notice that the player recommendation technique can also be used to improve a game engine by suggesting a good team to computer-controlled characters in a sport game.

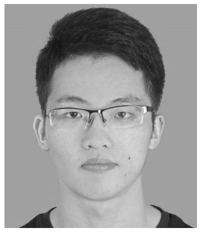
In the future work, we will research more attributes of players and consider their interactions in a football pitch. In addition, improving the CEFG algorithm is a great challenge. We will seek for a better bound so as to improve the player recommendation quality.

## REFERENCES

[1] D.-C. Zhan, J. Tang, and Z.-H. Zhou, "Online game props recommendation with real assessments," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 1–15, 2017.
[2] M. Brocco and W. Woerndl, "Location-based team recommendation in computer gaming scenarios," in *Proc. Acm SIGSPATIAL Int. Workshop Querying Mining Uncertain Spatio-Temporal Data*, Chicago, IL, USA, 2011, pp. 21–28.
[3] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Programm.*, vol. 14, no. 1, pp. 265–294, 1978.
[4] J. Edmonds, "Matroids and the greedy algorithm," *Math. Program.*, vol. 1, no. 1, pp. 127–136, 1971.
[5] S. Braghin, J. T. T. Yong, A. Ventresque, and A. Datta, "SWAT: Social Web application for team recommendation," in *Proc. IEEE 18th Int. Conf. Parallel Distrib. Syst.*, Singapore, Dec. 2012, pp. 845–850.
[6] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 677–702.
[7] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: A recommender system for groups of users," in *Proc. 7th Eur. Conf. Comput. Supported Cooperation Work*, Bonn, Germany, 2001, pp. 199–218.
[8] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu, "A group recommendation system for online communities," *Int. J. Inf. Manage.*, vol. 30, no. 3, pp. 212–219, 2010.
[9] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 754–765, 2009.
[10] L. Li, H. Tong, N. Cao, K. Ehrlich, Y.-R. Lin, and N. Buchler, "Replacing the irreplaceable: Fast algorithms for team member recommendation," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 636–646.
[11] C.-T. Li and M.-K. Shan, "Team formation for generalized tasks in expertise social networks," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, Minneapolis, MN, USA, Aug. 2010, pp. 9–16.
[12] T. Lu and C. Boutilier, "Budgeted social choice: From consensus to personalized decision making," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 280–286.

[13] P. Skowron, P. Faliszewski, and J. Lang, "Finding a collective set of items: From proportional multirepresentation to group recommendation," *Artif. Intell.*, vol. 241, pp. 191–216, Dec. 2016.
[14] W. Wang, Z. He, P. Shi, W. Wu, and Y. Jiang, "Truthful team formation for crowdsourcing in social networks: (Extended abstract)," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, Singapore, 2016, pp. 1327–1328.
[15] W. Wang, J. Jiang, B. An, Y. Jiang, and B. Chen, "Toward efficient team formation for crowdsourcing in noncooperative social networks," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4208–4222, Dec. 2017.
[16] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu, "Top-$K$ team recommendation and its variants in spatial crowdsourcing," *Data Sci. Eng.*, vol. 2, no. 2, pp. 136–150, 2017.
[17] S. A. P. Parambath, N. Vijayakumar, and S. Chawla, "SAGA: A submodular greedy algorithm for group recommendation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 3900–3908.
[18] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.
[19] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proc. Conf. Uncertainty Artif. Intell.*, Edinburgh, U.K., 2005, pp. 324–331.
[20] L. Su, J. Tang, D. Liang, and N. Wang, "A video co-segmentation algorithm by means of maximizing submodular function and RRWM," *Acta Automatica Sinica*, vol. 42, no. 10, pp. 1532–1541, 2016.
[21] Y. Zhang, X. Chen, J. Li, W. Teng, and H. Song, "Exploring weakly labeled images for video object segmentation with submodular proposal selection," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4245–4259, Sep. 2018.
[22] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA, 2011, pp. 510–520.
[23] N. R. Devanur, Z. Huang, N. Korula, V. S. Mirrokni, and Q. Yan, "Whole-page optimization and submodular welfare maximization with online bidders," *ACM Trans. Econ. Comput.*, vol. 4, no. 3, pp. 1–20, 2016.
[24] G. Radanovic, A. Singla, A. Krause, and B. Faltings, "Information gathering with peers: Submodular optimization with peer-prediction constraints," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 1603–1610.
[25] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge, U.K.: Cambridge Univ. Press, 2014, pp. 71–104.
[26] H. Zhang and Y. Vorobeychik, "Submodular optimization with routing constraints," in *Proc. 13th AAAI Conf. Artif. Intell.*, Palo Alto, CA, USA, 2016, pp. 819–826.
[27] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Jose, CA, USA, 2007, pp. 420–429.

**YIFENG ZENG** received the Ph.D. degree from the National University of Singapore, Singapore, in 2006. He is currently a Professor of artificial intelligence with the School of Computing, Teesside University, U.K., and also an Affiliate Professor with Xiamen University, Xiamen, China. Most of his publications appear in the most prestigious international academic journals and conferences, including the *Journal of Artificial Intelligence Research*, the *Journal of Autonomous Agents and Multi-Agent Systems*, the International Conference on Autonomous Agents and Multi-Agent Systems, the International Joint Conference on Artificial Intelligence, and the Association for the Advancement of Artificial Intelligence. His research interests include intelligent agents, decision making, social networks, and computer games. He received the EPSRC New Investigator Award, in 2018.



**GAOYANG SHEN** received the B.S. degree in automation from Xiamen University, in 2016, where he is currently pursuing the master's degree in artificial intelligence and system engineering. His research interests include data mining and recommendation systems.

**BILIAN CHEN** received the Ph.D. degree from The Chinese University of Hong Kong, in 2012. She is currently an Associate Professor with Xiamen University. Her publications have appeared in the *SIAM Journal on Optimization* and the *Journal of Global Optimization.* Her research interests include operational research, optimization theory, and recommendation systems.

**JING TANG** received the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. She is a Senior Lecturer with the School of Computing, Teesside University. Her publications have appeared in CEC, GECCO, and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATIONS. Her research interests include evolutionary algorithms, memetic algorithms, and artificial intelligence.