Utilizing Heuristic Based Depth First Search with Pruning to Solve EA Sports FIFA Squad Building Challenges

Introduction

EA Sports' FIFA series is among the most globally recognized video games, where its Ultimate Team mode enables the creation of customized squads with individual players from various leagues, ratings, and nationalities. A popular feature in the game lies in Squad Builder Challenges (SBCs) - a functionality that allows gamers to submit a squad of eleven players that meet a specific criteria, which can vary from straightforward rating requirements to advanced demands of players from specific nationalities or leagues. These challenges are an integral component of the game with hundreds of millions of SBC solutions completed annually, although deriving these solutions can often be a difficult process entailing trials and error and can consume a lot of time. Skilled players who crack these challenges sometimes share their solutions on platforms like Futbin and Futwiz, aiding other players. Yet, this system begs the question - Is there a more systematic, less trial-and-error-dependent method to solve these challenges? This project, therefore, aims to address this dilemma by engineering a programmatic solution to reduce the guesswork and provide more efficient resolutions to FIFA's SBCs.

As of the time of writing, there are 18, 619 different player cards in the Ultimate Team game mode. On this principle alone, there are many different combinations of players that can be combined to make a squad. Each squad is made up of 11 starting players, which leaves us with an astronomical amount of different permutations of a squad.

$$P(n,\ r)\ =\frac{n!}{(n-r)!}$$
$$P(18619,\ 11)\ =\frac{18619!}{(18619-11)!}$$
$$9.002E35$$

As in most computing scenarios, brute force is not a feasible solution shown by the unruly amount of unique permutations of players that are present. A core component of a squad is the chemistry rating each individual player receives. For those unaware, chemistry can be thought of as the bond between players that play for the same club, in the same league or hail from the same nation. This allows the players to perform more together as a unit. Chemistry requirements are often a core requirement for an SBC. Table 1 displays the chemistry system in FIFA. Like the chemistry system, every other requirement can be modeled as a formula or a mathematical notation. Thus, utility functions were developed for each requirement, to easily calculate if players meet the given requirements.

Table 1. ***Chemistry System***

	1 Chemistry	2 Chemistry	3 Chemistry
Club	2 players	4 players	7 players
Nationality	2 players	5 players	8 players

League	3 players	5 players	8 players

Heuristic Based Depth First Search with Pruning

Heuristic-based Depth First Search (DFS) with pruning is a derivative of the standard DFS algorithm, using a heuristic evaluation function to guide the search process and pruning to reduce the search space. In the context of Squad Building Challenges (SBCs), each player is treated as a node in a graph. A path from the root node to a node with no children represents a possible team, and the algorithm aims to find a valid team that meets the SBC's requirements. In a basic DFS, the algorithm starts at a root node (an empty team in our case), and systematically explores each branch to its fullest extent before backtracking and moving on to the next branch. It does this by adding a player to the team, then recursively calling DFS to explore all teams that can be made with that player included. If the algorithm finds a valid solution (a team that meets all the SBC's requirements), it terminates and returns that solution. If not, it removes the player from the team and moves on to the next player.

However, the basic DFS algorithm doesn't always perform efficiently due to the large search space involved. Given the enormous number of possible teams, DFS could potentially take an inordinate amount of time to find a solution. Hence, we introduce a heuristic and pruning to guide and limit our search. Heuristics are employed to guide the search process by ranking nodes based on their potential to lead to a solution. In our case, a heuristic could be the player's rating or chemistry potential. A node (player) with a higher heuristic value would be explored before a node with a lower value. This method allows the algorithm to prioritize promising paths, thereby potentially finding a valid solution faster.

Pruning, on the other hand, reduces the search space by eliminating paths that cannot possibly lead to a valid solution. In the context of the SBCs, our pruning function considers the current team and the SBC's requirements. Before exploring a path, the function checks if it is theoretically possible for the current team to meet the requirements. If the team's current chemistry plus the maximum possible chemistry from adding more players is less than the chemistry requirement, or if the team's current rating plus the maximum possible rating from adding more players is less than the rating requirement, the path is pruned, meaning it's not explored. This function also checks whether the first six players meet the minimum club, league, and nationality requirements, providing another opportunity for early pruning.The combination of heuristic-based DFS and pruning drastically reduces the search space, thereby making the task of solving SBCs significantly more efficient and less time-consuming. It leverages the strengths of DFS's systematic exploration while mitigating its weakness of potentially large search spaces through the use of heuristics and pruning, culminating in a solution that can efficiently and reliably solve complex SBCs.

Conclusion and Time Results

The results of the comparison between the brute force and heuristic-based DFS with pruning approaches are presented in Table 2. It's evident that the latter significantly outperforms the former in terms of efficiency and speed. The runtime was compared with the SBC requirements from Table 3.

The brute force algorithm, serving as our baseline, generates teams by randomly picking 11 players and checking whether the resulting team satisfies the SBC requirements. As this method doesn't incorporate any strategic guidance or pruning, it tends to be relatively slow and inefficient. From Table 2, it can be seen that the average runtime for brute force is approximately 5.71 seconds. Furthermore, the number of iterations is around 578, it's clear that the brute force algorithm is conducting a significant amount of unnecessary computation.

On the other hand, the heuristic-based DFS with pruning approach introduces a more structured and efficient way to solve the SBCs. The DFS algorithm explores the search space methodically, guided by a heuristic evaluation function. Additionally, the pruning mechanism further enhances the efficiency by discarding the paths that cannot possibly lead to a valid solution early in the search process. This combination reduces the search space substantially, resulting in a more efficient and faster algorithm. As shown in Table 2, the DFS approach exhibits a remarkable improvement in runtime, with an average of approximately 0.12 seconds. The dramatic reduction in runtime compared to brute force signifies that the heuristic-based DFS with pruning circumvents the need for unnecessary brute force calculations. The number of iterations is not applicable here as the DFS approach finds the solution path more directly without needing to iterate through all possible permutations.

In conclusion, the results affirm that the heuristic-based DFS with pruning provides a more practical and efficient solution for solving a SBC. It substantially reduces the computational load and time compared to the brute force approach, thereby offering a powerful strategy for navigating the large search space presented by the SBCs. This improvement enhances the feasibility of programmatic solutions for complex SBCs in FIFA.

Table 2. ***Average Runtime (100 runs)***

	Brute Force	DFS
Runtime	5.707340	0.123610
Iterations	578	N/A

Table 3. ***SBC Requirements***

Min Squad Rating	Min Chemistry	Formation	Min Unique Clubs	Min Unique Nations	Min Unique Leagues
70	*15*	*3-4-1-2*	*6*	*4*	*2*

Future Work

Every player card in FIFA Ultimate Team carries an associated in-game monetary value and can be traded on the transfer market. Various factors influence the market value of a player card, including better in-game attributes, affiliation with popular leagues and clubs, or high demand due to an ongoing Squad Building Challenge. The current algorithm does not incorporate this dynamic of player card values into its heuristic, leaving room for further optimization and practicality. By considering a player card's market value in the algorithm, I can introduce a new dimension to the problem-solving process and refine the heuristic function.

In future work, I intend to adjust the algorithm to reflect this aspect of the game. This enhancement will add a layer of practicality to the algorithm, making it not just about meeting the SBC's requirements, but also about achieving a cost-effective solution. This additional aspect of the algorithm will provide players with a more practical solution for tackling SBCs.

References

Futbin. “FIFA 23 Ultimate Team Prices, Squad Builder, Draft and Players Database.” *FUTBIN*, www.futbin.com/. Accessed 9 June 2023.

“FIFA 23 Ultimate Team Prices, Draft Simulator, Squad Builder.” *FUTWIZ*, www.futwiz.com/. Accessed 9 June 2023.

Appendix of Code

All the code for this project can be found at this github repository, https://github.com/woublion/FIFA-SBC-Solver. Feel free to run the code yourself, if your curiosity permits!