

CIS 559 PROJECT 1: PARALLEL FOOTBALL

IAN SIBNER, DERICK OLSON, SPRIHA BARUAH, ANTHONY HSIEH

22 September 2015

CONTENTS

1	Introduction	2
2	Initial Insights and Observations	3
3	Strategies and Concepts	4
3.1	Movement & Benefit Function	4
3.2	Benefit Function	4
3.3	Cell clustering	4
3.4	Kicking	4
4	Implementation	5
4.1	Composability	5
4.2	Locking Mechanism	6
4.3	Auxiliary Board	6
5	Results	6
6	Contributions	6
7	Future Directions and Limitations	7
7.1	Strengthen Early Game Through Sweeping	7
7.2	Generalized Step Distance	7
7.3	Pathing	8

8	Acknowledgments	8
9	Conclusion	9

LIST OF FIGURES

Figure 1	Initial board of a parallel football game	3
Figure 2	State of a parallel football board after 150 turns	5

1 INTRODUCTION

Parallel Soccer is a game in which four teams of P (which can vary from 1 to 250) players compete to kick soccer balls into their teams' goal. The goals are arranged in each corner of a 32×32 grid, and the rest of the squares on the board are initialized to contain one ball each, for a total of 1,020 balls on the board to start off with. To start off with, each team can place each of their P players anywhere on the board; multiple players can potentially occupy the same square. After that, each player (with full knowledge of the board) must choose to either *kick* a ball (in their square) up to K distance away (with K being a constant whose value may vary between one and 45), or *move* up to 1 square in any direction (including diagonals). When a ball is kicked into a team's goal, that team is awarded a point and the ball is removed from play. The game ends when all balls are removed from play, and the team with the most points wins.

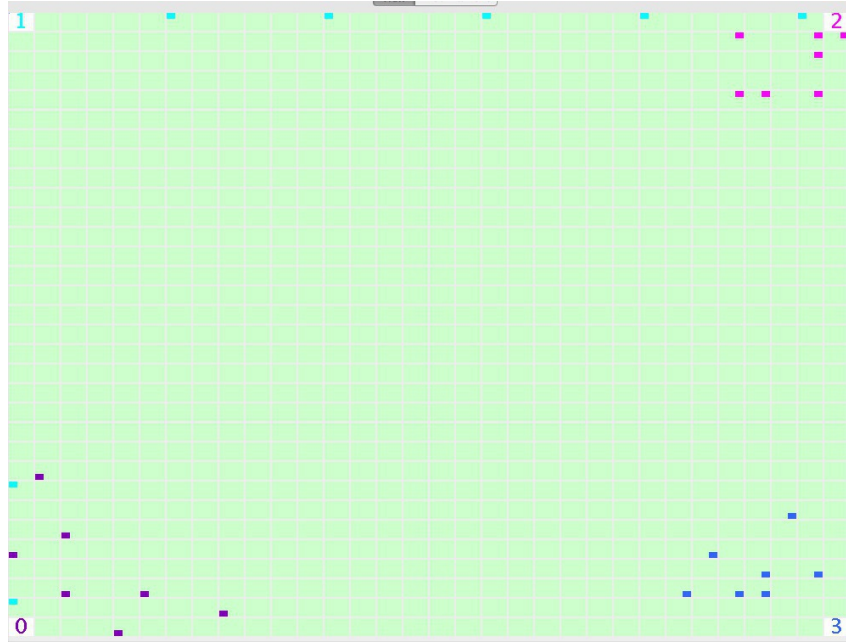


Figure 1: An initial positioning of the board where $P = 7$

2 INITIAL INSIGHTS AND OBSERVATIONS

Immediately, our team observed several key features about the game.

1. The game is zero-sum. That is, any ball that is scored by our player is denied to all of our opponents, and vice versa.
2. Efficiency is not a constraint. Because the game would not progress to the next step until all of our players had calculated their next move, we could take the entire board into account without worrying about the runtime of our solution.
3. Strategies may work well for certain combinations of P and K and not others. A smart player would change its strategy depending on what value these constants took on.
4. Certain strategies may work particularly well in the early phase of the game, when all the balls are on the board and there are not clusters, while others may work better in the middle phase (where there are fewer balls and more clusters) or in the end game (when there are only a few balls which every team is competing for).
5. Efficient strategies (i.e. those which required to fewest moves to kick a ball into the goal) would generally favor kicking over moving, since $K \geq 1$. This naturally leads to a sort of “bucket brigade” where players kick balls close to one another in order to “pass” them towards the goal.

These insights guided our strategy throughout and allowed us to focus on the most important aspects of the game.

3 STRATEGIES AND CONCEPTS

3.1 Movement & Benefit Function

In order for a player to decide which cell was the best move for any given board state, we created a benefit function that assigned a score to each cell for every given player. Since we had access to the entire board, we used many factors in this function, including the number of balls in a cell (and surrounding cells), distance from the cell to our goal, distance from the cell to the player, and the number of opponents around the cell. Given the loose efficiency constraints given the problem, we decided that a globally optimal benefit function would be both feasible and desirable. The primary gain from this decision was a team optimized for late gameplay, where our players were able to seek out and steal balls toward the end of the game. The reason was that once we and other teams cleared the balls near our respective goals, balls became scarce. If any team had accumulated balls, even if they were far from our players, they would seek them out and attempt to score with them.

On the other hand, as teams improved, the game time grew shorter and quicker. In such a game, it is possible that a globally optimal zone will change into a depleted zone before a far-away team can reach it. It is likely that our model did not fully capture the constantly changing state of the game board. We addressed this by heavily weighting our own goal in benefit calculations.

3.2 Benefit Function

3.3 Cell clustering

On the second iteration of our benefit function, we accounted for ball and opponent clustering by keeping track of the number of balls and opponents surrounding a particular cell. This allowed our players to seek out areas of high benefit, rather than single cells.

After initial trials with a large radius, we settled on a radius of 3. We observed that larger radii caused the players to move back and forth somewhat inefficiently, and hypothesize that these larger zones changed too quickly for players to capitalize on them.

3.4 Kicking

Initially, our players would kick the maximal distance towards the goal. Although this will guarantee that the ball will be as close to the goal as possible, it does not necessarily mean that the number of steps to kick that ball in the goal is minimized when there are teammates around. With this insight, we improved the kicking algorithm with a simple calculation. A player will look at all possible cells it can kick to and minimize the number of steps to kick that ball into the goal for anyone on its team. What this translates to

in practice is that a player will now pass the ball as close to a teammate as possible if the teammate is closer to the goal than itself. We observed that after implementing this kicking algorithm, the players naturally form a bucket brigade to minimize the number of steps a player needs to go to for the ball to score.

4 IMPLEMENTATION

4.1 Composability

Because we realized that certain strategies might work very well in the early phase while others might excel in the late phase, we set about building a player that would *compose* other players by switching between its component players' strategies. We met with a lot of success the first week by composing Group 1's GridPlayer for the first 200 turns and our own mid-game-optimized DerickPlayer for the remainder of the game. This worked about as expected; GridPlayer was far more efficient than DerickPlayer in the early game, but after about 200 turns, the board had grown sparser and our own player was much better at seeking out balls and scoring them. We believed this was the way forward and set about finding a more intelligent way to determine when to switch (factoring in P, K, and the number of balls left on the board).

However, during the third week of gameplay, the rest of the teams (and our own DerickBrigadePlayer) had become so sophisticated that the early stage of the game completely dominated. Essentially, the winner was determined by how well the early phase played out, and the middle/end phases were nearly nonexistent. Thus we decided to play a pure strategy in the final round, focused on early-game optimization, rather than continuing to compose multiple strategies.

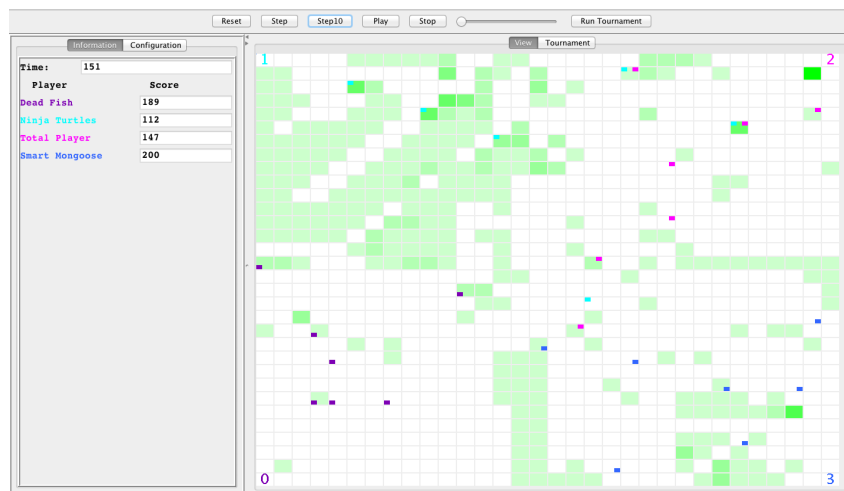


Figure 2: The state of the board after 150 turns with $P = 7$, $K = 6$. Note how over 60% of the balls are already scored; this illustrates the great importance of the early game when sophisticated strategies are placed in competition.

4.2 Locking Mechanism

As we found through our trials, using a benefit function to determine where a player should go runs into problems when players want to go to the same cell. The players will eventually converge and go to the same cells the rest of the game. To prevent this major inefficiency, we implemented a locking mechanism where only one player can go after a single cell at a time. However, we found that although this works great through most of the game, when the number of balls is very small, it becomes better for multiple players to go after the same ball to have a higher likelihood of kicking the ball to the goal. As such, when the number of balls reaches the number of players on the field, we turn off the locking mechanism.

After only locking the cell that the player is going towards, we also lock a radius around that cell. This allows us to create zones such that only one player clearing a zone. Through experimentation, we found that choosing a radius of $\max(0, 3 - P/10)$ works well. With higher number of players, the smaller the zones should become because there is not enough space to create so many zones. Additionally, it is more likely that opponents will steal balls from a particular zone before a single player can clear that zone out.

4.3 Auxiliary Board

5 RESULTS

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

6 CONTRIBUTIONS

Just as we adopted many good ideas by other teams (see Acknowledgements section), we contributed two major ideas to the group discussion over the course of the project:

1. **Benefit function** - HI SPRIHA PLZ WRITE THIS
2. **Composable Players and Game Phases** - Although we realized early on that the game would likely be characterized by distinct phases in which certain strategies

would dominate, it was not until week 2 when other groups really started to pay attention to this idea. This was largely because our performance that week was extremely dominant after we composed `GridPlayer` in the early game and `DerickPlayer` for the rest of the round! Other groups began to think about how they could tailor their strategy to perform well in all phases of the game, or at least hold their own. One group even took this idea to the extreme, creating a `ReinforcementLearningPlayer` that essentially composed *all* players using a 1NN algorithm!

7 FUTURE DIRECTIONS AND LIMITATIONS

7.1 Strengthen Early Game Through Sweeping

Among teams that placed highly in the tournament, we noticed that their players would tend to start the game by sweeping balls inwards towards their own goal, maximizing their early-game wins before beginning to spread out across the board. Our players, which started close to our own goal, did not exhibit this behavior. Adapting our placement strategy so that they started farther away might be sufficient to achieve this result; however, we could also look into a composability solution in which a sweeping strategy is used for the very first portion of the game before switching to our current players' behavior. This would require significantly tweaking our composability behavior, particularly the point at which the behavior switch occurred, in order to ensure that our players did not continue sweeping for too long before switching to mid-game behavior.

7.2 Generalized Step Distance

This idea stems from our attempts to take advantage of global-scope "hotspots," or high-density ball clusters that are not near the goal. The existing implementation weights balls close to the home goal particularly high, which is good in the early game, but may not be ideal overall.

As we can see in the figure above, certain strategies tend to cluster balls near an opponent goal, where they stay relatively untouched for several cycles. A smarter global-cluster detection would drop the current strategy

The biggest disadvantage to this approach is that the time it takes to travel to such a cluster may be greater than the time the cluster exists. Even if the seeking team is able to reach the cluster in time, the balls lost in the transit time may not be worth it.

It would be possible to determine, by factoring in the number of balls left B_{total} , the number of balls in close range B_{nearby} , and the number of balls expected to remain in the cluster $B_{cluster}$. Such a strategy would decide to go for the cluster if and only if:

$$B_{total} - B_{nearby} < B_{cluster}$$

One way of implementing this strategy would be to generalize the `numberOfStepsToGoal()` benefit function to be `numberOfStepsToPosition(p)`, where the latter takes in any valid position `p` as its argument, causing the cells around it to get a score boost. Then, upon deciding to pursue a cluster, this position would be updated to the cluster center, until a new position was found.

7.3 Pathing

It becomes clear through watching our players that a pathing algorithm could improve our players drastically. We found that our players will often take the same path to their desired locations even though their destinations are distinct. This creates inefficiencies as if there are balls on that path, only one player is needed to clear those balls. In the future, we would look into minimizing the number of cells such that two players would through to reach their destination.

Another problem with our players with pathing is that they arbitrarily choose a minimum path to its destination. So, if two paths are equally minimal, the player will not choose the path with the most balls, thus missing opportunities to score more points. In the same vein, if two paths are slightly different in the number of steps but the longer path has way more balls, the player will choose the shorter path, causing inefficiencies. To fix this problem, The path the player should take should additionally take into consideration how many balls the player can kick on that path.

8 ACKNOWLEDGMENTS

Our progress was largely a result of the class discussions and adopted strategies. Through the course of the project, we adopted starting strategies from Groups 1 (Grid Player), as well as experiments in stealing opponent balls with the opening placements. We would like to give special thanks to Groups 2 and 6 for the important heuristics we adopted from them and eventually used in our final implementation.

Group 2 (The Swarm) was the first group to successfully implement a kind of benefit function with the emergent behavior of a bucket brigade. This function was based on the number of steps from the ball to the goal, and we adopted it as a core feature of our ensemble benefit function.

Group 6 (Bucket Brigade) contributed a smarter kicking function that favored passing to players over kicking as far to the goal as possible. We adopted this idea as a core part of our improved kicking strategy.

9 CONCLUSION

Overall we were happy with the progression of this project and the contributions that our group was able to make to the class, including the benefit function and the composable players. We were also quite pleased with our performance during the second week of the project when our player came out on top!

We hope to take the lessons learned on this project and apply them to later projects in the class. In particular, the idea of optimizing for the early stage of the game (which turned out to be key in parallel football) may prove extremely useful later on as well. Especially in zero-sum games like this one, it can be difficult to stage a come from behind victory when one team starts out with a convincing lead.