ITAndroids 2D Soccer Simulation Team Description Paper 2023

Davi M. Vasconcelos, Nean Segura, and Marcos R. O. A. Maximo

Aeronautics Institute of Technology,
São José dos Campos, São Paulo, Brazil
{davi.muniz41,seguranean}@gmail.com,mmaximo@ita.br
itandroids-soccer2d@googlegroups.com
http://www.itandroids.com.br/

Abstract The ITAndroids 2D Soccer Simulation team is composed by undergraduate students of the Aeronautics Institute of Technology. The team is currently one of the strongest teams in Brazil, having won first place 4 times consecutively from 2012 to 2015, Vice Champion in 2018 and was the Champion of the 2019 Latin American Competition. Moreover, the team has qualified for the last ten editions of RoboCup, having participated in nine. This paper describes some of our advances in 2022 and our plans for 2023.

1 Introduction

ITAndroids is a competitive robotics team from Aeronautics Institute of Technology reestablished in 2011. The group participates in the following leagues: RoboCup 2D Soccer Simulation, RoboCup 3D Soccer Simulation, RoboCup Humanoid Kid-Size, IEEE Humanoid Robot Racing, IEEE Very Small Size, and RoboCup Small Size League.

Our Soccer 2D team, ITAndroids 2D, has continuously participated in Latin American Robotics Competition (LARC) and Brazilian Robotics Competition (CBR – acronym for *Competição Brasileira de Robótica*) since 2011. Moreover, ITAndroids 2D competed in RoboCup in 2012, 2013, 2015, 2016, 2017, 2018, 2019, 2021 and 2022. The team also was qualified for RoboCup 2014, but unfortunately it was not able to attend the competition. Our results in these competitions are represented in Table 1.

Lack of continuation and documentation of the project and the spreading of the team towards other fields slowed down its improvements. After a complete restructuring of the project [1], we won 9th place at RoboCup 2018, our best absolute place in the competition.

2 Previous Works

Our code base uses agent2d [2] as base team. Since the year 2012, we have focused on improving mechanisms already present in agent2d framework. We have

Table 1: Placement of ITAndroids 2D in past RoboCup and LARC competitions.

Year	RoboCup	LARC
2022	11th	3rd
2021	11th	5th
2020	—	4th
2019	13th	1st
2018	9th	2nd
2017	15th	3rd
2016	13th	2nd
2015	13th	1st
2014	—	1st
2013	13th	1st
2012	10th	1st

improved the action chain evaluator with Particle Swarm Optimization (PSO) [3]. Furthermore, we have developed heuristics [1] to increase attack and defense performance: type of formation (attack or defense) selection based on probability of scoring a goal, field evaluator selection based on opponent team, and defender optimal marking in opponent attack situations. Many early improvement ideas were inspired by Team HELIOS [4] and Team Nemesis [5].

The team proposed in 2018 a novel technique to determine the in-game ball possession [6]. The ball possession information without noise is valuable for any team, since it can be useful to create dynamic behaviors in players. We have used a Finite-State Machine called Possession Automaton that takes into account the current and the last game situations to infer the ball possession. The game situations are determined by InterceptTable simulations, the ball possession estimator in agent2d [2]. Since we do not estimate ball possession based on a single game cycle, we have obtained a classification accuracy 18% higher than the default estimator of the base team.

Lately we have experimented Deep Reinforcement Learning (DRL) techniques to improve the goalkeeper defense in penalty situations [7]. After five training experiments using Proximal Policy Optimization (PPO) algorithm, we have achieved a penalty defense rate of 40% against agent2d [2], twenty percent higher than the base team rate.

Our current efforts focused on improving the action chain evaluator with Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and on modeling the opponent formation strategy. Several teams of RoboCup 2D Soccer Simulation have proposed opponent formation classifiers. Fukushima, Nakashima, and Akiyama [8] have developed an online opponent formation classifier for defense situations of opponent. Faria *et al.* [9], Almeida *et al.* [10] have proposed a classifier for predefined soccer formations (e.g., 4-3-3, 3-5-2). Furthermore, due to lack of workforce, we decided to stop the development of our game log analysis tools [11].

3 Optimization of Field Evaluator Weights with CMA-ES

The agent2d [2] has a simple field evaluator that gives a value to each state in the action chain mechanism. The value helps to measure if a state is good or bad when we wish to score a goal in the future. We have modified it by multiplying the agent2d [2] values by weights and adding new hand-coded rules. Thus, we use six weights in the field evaluator, each one measures the influence of a state's characteristic when scoring a goal in the future.

A weight vector that improves the team's performance can be found using black-box optimization. In fact, we have previously optimized the field evaluator weights with Particle Swarm Optimization (PSO) [3] [7]. In this section, we show the results of field evaluator weights optimization using Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which outperformed PSO in several benchmark functions [12].

Our Soccer 3D team has developed an open-source distributed optimization tool for RoboCup 3D Soccer Simulation [13]. The tool can run in the Intel® DevCloud environment, which provides free computational resources for computing-intensive applications. Furthermore, the tool supports the pycma optimizer [14], a Python implementation of CMA-ES. Therefore, we adapted the tool for running optimizations in RoboCup 2D Soccer Simulation.

We defined the fitness function as

$$f(G_{our}, G_{opp}, P_{our}) = \frac{\tanh\left(\frac{G_{our}-G_{opp}}{3}\right)+1}{3} + \frac{P_{our}}{3}, \tag{1}$$

where G_{our} is the number of goals scored by our team, G_{opp} is the number of goals scored by opponent team, and P_{our} is the ball possession of our team in `play_on` game mode. We estimated the ball possession from game log by assigning the possession to the team with the player closest to the ball. If the distance between the ball and the player closest to it is greater than a threshold, then the ball possession is unknown. However, if a sequence of unknown possessions from consecutive game cycles has the previous known possession and the next known possession to be equal, then we set the sequence of unknown possessions to the next known possession. Thus, we can better assign ball possession when a player successfully passes the ball.

Since the RoboCup Soccer Simulator is a stochastic environment, we need to find the weight vector that maximizes the expected value of (1). Hence, we run 10 matches to estimate the expected value. The optimization parameters are represented in Table 2. The weights of the field evaluator were initialized as w_0. The σ_0 parameter changes the range of the search space, and we assigned it a value that expects the optimum weights to lie within $[0, 1]$. The popsize parameter sets the population size of the evolutionary algorithm. We selected it using the suggested heuristic in [15], which computes the population size for a search space of dimension n by the expression $4 + 3\ln(n)$.

Finally, we run the optimization against Team RoboCIn for 200 steps of the algorithm. The matches were run with no penalties and no extra time. Figure

1 shows the best solution found at each step. Table 3 exhibits the performance of our team against Team RoboCIn before and after optimization. The performance of ITAndroids decreased when playing with the best weights found in the optimization. Therefore, we should increase the number of matches when estimating the expected value of (1), since the randomness in RoboCup Soccer Simulator may affect the search of CMA-ES for better solutions.

Table 2: Optimization configurations in the Intel® DevCloud environment.

Parameter	Value
σ_0	1/6
w_0	1/2
popsize	9

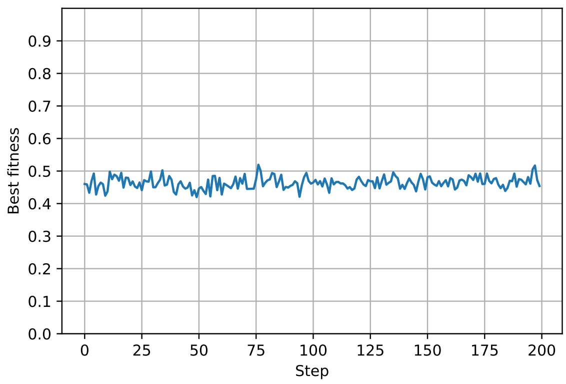

Figure 1: Best value of the fitness function at each optimization step.

Table 3: Performance of Team ITAndroids (2023) against Team RoboCIn (2022) in 500 matches. We started on the left side of the soccer field.

Team	0-0	0-1	0-2	0-3	0-4	0-6	1-0	1-1	1-2	1-3	1-4	1-5	2-0	2-1	2-2
Before optimization	139	148	68	25	6	1	37	34	24	6	1	2	5	3	1
After optimization	136	196	106	49	4	0	3	3	2	0	0	1	0	0	0

4 Analysis of Opponent Soccer Formations

The development of effective dynamic behaviors against different teams is a major challenge in the RoboCup 2D Soccer Simulation domain. In order to efficiently change the playing style of an agent, we should identify the current opponent strategy first. Although team strategy can be addressed in many ways, we could approximate it to the team formation strategy, since the team positioning is a major factor in the team strategy. Thus, once we have identified the formation strategy of the opponent, we can dynamically rearrange our players to increase the performance in attack or defense situations.

Defining a representation for soccer formation may not be an easy task. We need to guarantee that the permutation of teammates does not change the formation, since it does not modify the shape of the team. Team HELIOS [8] has represented a formation as the number of players on each cell of a discrete soccer field and has made the assumption that opponent formation is static. Therefore, as a first experiment, we used the formation representation of HELIOS with a discrete soccer field with 20 rows and 30 columns. Moreover, we considered that a formation of a team in a given match is the average formation of all `play_on` cycles.

We obtained the formation data from 140 half-matches per team of ITAndroids against the following teams: Persepolis (2021), RoboCIn (2021), Ri-one (2021), Titans (2020), HELIOS (2018), CYRUS (2021), Futvasf (2021), ThunderLeague (2021), JyoSen (2021), Hades2D (2021), and Alice (2021). Then, we converted the rcg log files to a tabular format with `rcg2csv`, which is available in librcsc [2]. Later, we removed formation data when the game mode was not `play_on`.

Similar to HELIOS approach [8], we used the Gaussian Mixture Model (GMM) clustering algorithm to find similar opponent formations. Since GMM is sensitive to its initial parameters, we run 2,000 clustering rounds with different random seeds. We modified the number of clusters from 2 to 15 in each clustering round. Furthermore, we selected the optimal number of clusters of each clustering round with the Calinski-Harabasz Index.

Table 4 gives the most probable optimal number of clusters equals two. Hence, as the best cluster, we selected the cluster with the highest Calinski-Harabasz Index among optimal clusters from clustering rounds with optimal number of clusters equals two. The final label distribution is represented in Table 5, where L_i represents label i. We can see that each team is assigned to a single label.

Figures 2 and 3 show the average of 140 mean formations for each team with labels L_0 and L_1, respectively. All formations but CYRUS and Persepolis are symmetric with respect to the horizontal axis. CYRUS and Persepolis mostly move to positions with negative y-coordinate when playing against us. Furthermore, the positioning of Hades2D is different from all the teams.

Table 4: Distribution of optimal number of clusters in 2,000 clustering rounds.

Number of clusters	Percentage
2	89.4%
3	9.8%
4	0.8%

Table 5: Label distribution of the mean formations in the final clustering result.

Team	Percentage (L_0)	Percentage (L_1)
Persepolis (2021)	100.0%	0.0%
RoboCIn (2021)	100.0%	0.0%
Ri-one (2021)	100.0%	0.0%
Titans (2020)	100.0%	0.0%
HELIOS (2018)	100.0%	0.0%
CYRUS (2021)	100.0%	0.0%
Futvasf (2021)	0.0%	100.0%
ThunderLeague (2021)	0.0%	100.0%
JyoSen (2021)	0.0%	100.0%
Hades2D (2021)	0.0%	100.0%
Alice (2021)	0.0%	100.0%

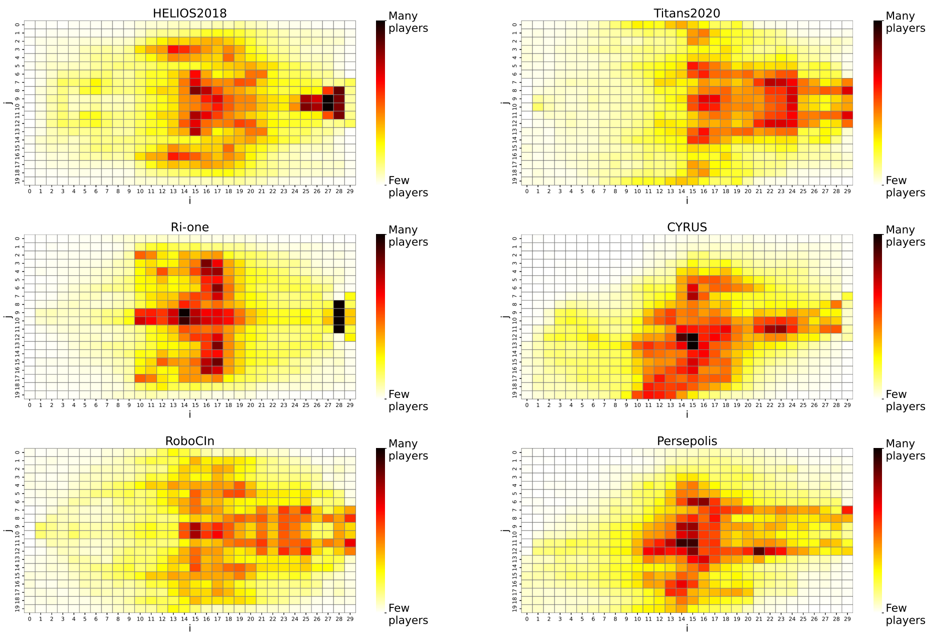

Figure 2: Average of mean formations for each team with label L_0 in 140 half-matches.

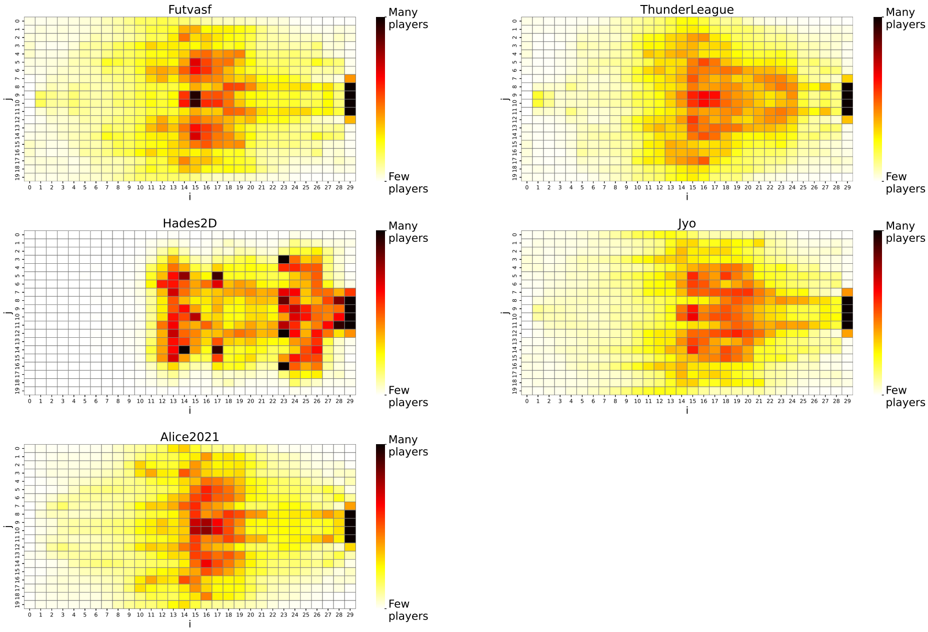

Figure 3: Average of mean formations for each team with label L_1 in 140 half-matches.

5 Conclusions and Future Work

This paper presented the most recent efforts of team ITAndroids 2D. We developed a tool for running optimizations in RoboCup 2D Soccer Simulation built upon a tool developed by our Soccer 3D team [13]. Then, as a first optimization experiment, we optimized our field evaluator weights against Team RoboCIn. Later, we analyzed the mean formation of several opponents against us.

In the future, we will run optimizations with a greater number of matches per step to deal with the randomness of RoboCup Soccer Simulator. Furthermore, we will search for a formation representation that includes temporal information when analyzing the opponent formations.

6 Acknowledgements

We would like to acknowledge the RoboCup 2D Soccer Simulation community for sharing their developments. In particular, we would like to acknowledge Hidehisa Akiyama for agent2d, librcsc, soccerwindow2, and fedit2 [2]. Finally, we acknowledge our sponsors: Altium, Cenic, Intel, ITAEx, Mathworks, Metinjo, Micropress, Neofield, Polimold, Rapid, Solidworks, STMicroelectronics, WildLife, and Virtual Pyxis.

References

[1] Felipe Coimbra et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2017" (2017).

[2] Hidehisa Akiyama and Tomoharu Nakashima. "HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation". In: *The 17th annual RoboCup International Symposium*. July 2013.

[3] Fábio Mello et al. "ITAndroids 2D Soccer Simulation Team Description 2012" (2012).

[4] Hidehisa Akiyama and Hiroki Shimora. "HELIOS2010 Team Description" (2010).

[5] Mehrab Norouzitallab et al. "Nemesis Team Description 2010" (2010).

[6] Felipe Coimbra and Lucas Lema. "ITAndroids 2D Soccer Simulation Team Description 2018" (2018).

[7] Diego Fidalgo et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2020" (2020).

[8] Takuya Fukushima, Tomoharu Nakashima, and Hidehisa Akiyama. "Online Opponent Formation Identification Based on Position Information". In: *RoboCup 2017: Robot World Cup XXI*. Ed. by Hidehisa Akiyama et al. Cham: Springer International Publishing, 2018, pp. 241–251. ISBN: 978-3-030-00308-1.

[9] Brígida Faria et al. "Machine Learning algorithms applied to the classification of robotic soccer formations and opponent teams". In: July 2010, pp. 344–349. DOI: 10.1109/ICCIS.2010.5518540.

[10] Rui Almeida, Luís Paulo Reis, and Alípio Mário Jorge. "Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain". In: *Progress in Artificial Intelligence*. Ed. by Luís Seabra Lopes et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 239–250. ISBN: 978-3-642-04686-5.

[11] Davi M. Vasconcelos et al. "ITAndroids 2D Soccer Simulation Team Description Paper 2022" (2022).

[12] Paweł Szynkiewicz. "A Comparative Study of PSO and CMA-ES Algorithms on Black-box Optimization Benchmarks". *Journal of Telecommunications and Information Technology* 8 (Jan. 2019). DOI: 10.26636/jtit.2018.127418.

[13] Guilherme N. de Oliveira, Marcos R. O. A Maximo, and Victor V. Curtis. *Distributed Optimization Tool*. https://gitlab.com/itandroids/open-projects/distributed-optimization-tool. 2022.

[14] Nikolaus Hansen et al. *CMA-ES/pycma: r3.3.0*. Version r3.3.0. Jan. 2023. DOI: 10.5281/zenodo.7573532. URL: https://doi.org/10.5281/zenodo.7573532.

[15] Nikolaus Hansen. "The CMA Evolution Strategy: A Tutorial". *CoRR* abs/1604.00772 (2016). arXiv: 1604.00772. URL: http://arxiv.org/abs/1604.00772.