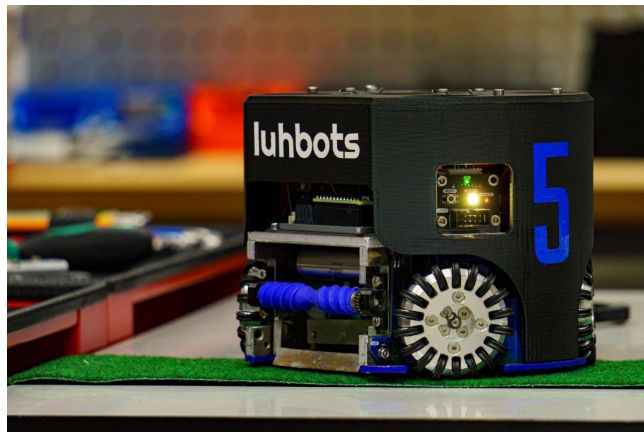


**luhbots soccer**  
**Extended Team Description for**  
**RoboCup 2024**  
(Strategic Positioning: Offensive Robot Placement in Robot Soccer)

Max Westermann, Tobias Pahl, Timo de Vries, Sebastian Knackstedt, and  
Larissa Seegemann

Faculty of Electrical Engineering and Computer Science  
Gottfried Wilhelm Leibniz University Hannover  
Appelstr. 11, 30167 Hanover, Germany  
[soccer@luhbots.de](mailto:soccer@luhbots.de)  
<https://luhbots-hannover.de>



**Fig. 1.** luhbots soccer robot from 2024

**Abstract.** This paper introduces a component of our offensive game-play strategy which is designed for deployment in the Soccer Small Size League of the RoboCup. Emphasizing the positioning of offensive robots, a fundamental component of our attack strategy, we present three different approaches: *Force Driven Positioning (FDP)*, *Restricted Radial Positioning (RRP)*, and the *Octopus Algorithm*. These methods will be elucidated by providing an explanation of their function as well as a comparison of their respective advantages and disadvantages.

## 1 Introduction

The luhbots soccer team made its debut in the RoboCup 2022. In Division B the team secured 3rd place at RoboCup 2022 and 2nd place at RoboCup 2023. Being a novel participant, luhbots soccer builds on its past successes and aspires to move up to Division A in the upcoming season.

This year, in pursuit of this goal, the TDP is written in an extended form for the first time. In the previous years, the focus lied on the introduction and overview of the system as a whole [3,4]. This paper primarily delves into the offensive game play strategy. Over the past two years various approaches have been developed and employed. The experiences and insights gained from these endeavors, along with the functionality of the algorithms, will be elucidated in this paper.

The preceding TDP [4] outlines the foundational architecture of our strategy, featuring the Observer, Role Manager, and Task Manager components. The Observer extracts high level data from the vision’s low-level data, such as the current goal probability, making it accessible to other modules. This information is used to assign a role to each robot by determining whether it should engage in offensive or defensive behavior. In turn the resulting action is controlled by the Task Manager. Robots sharing the same role operate collectively as a squad, which facilitates cooperation between them. This approach has proven to be extremely effective over the past year, due to the decision-making process being divided into several stages which can be processed and modified independently. This not only simplifies the development process, but also increases adaptability. In this paper, we will look at the behavior of the *Secondary Attackers*. Their main task is to support the *Primary Attacker* by strategically positioning themselves for a potential pass. The selection of this position is crucial. The aim is to create a scenario where intercepting the pass becomes challenging for the opponent. Simultaneously, the chosen pass should enhance the overall team’s prospects of scoring. To find such a position, three different approaches will be presented in this paper: *Force Driven Positioning*, *Radial Restricted Positioning*, and the *Octopus Algorithm*.

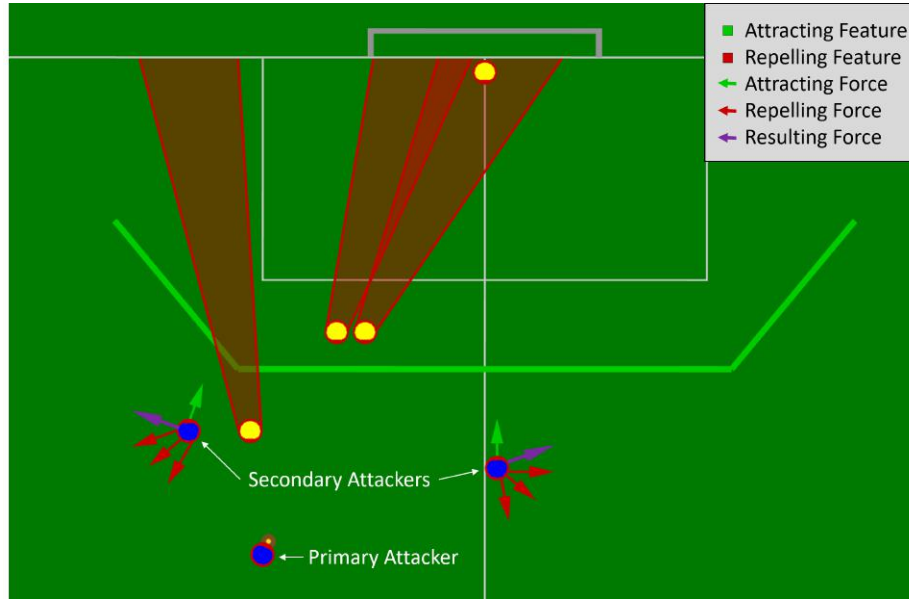
## 2 Force-Driven Positioning (FDP)

In the past the lubbots soccer team utilized *Force-Driven Positioning* (FDP). Features and anti-features are defined to attract or repel robots [3]. The resulting combined forces determine both the direction and speed of the robot’s movement. The selection of features is based on the following assumptions:

1. The passing line to the robot should remain unobstructed, preventing opponents from intercepting the ball.
2. The robot should position itself as far away from opponents as possible allowing time for ball manipulation after receiving the ball.
3. The robot should position itself close to the opponent’s penalty area increasing its chances for a successful goal shot after receiving the ball.
4. The robots should keep their distance from their teammates avoiding obstruction and forcing the opponent to split up their defenders.

The forces that act on the robots at all times can be derived from these assumptions. In order to position the robots close to the opponent’s penalty area, an attractive force should emanate from it. It makes sense not to define the penalty area itself as a feature but rather lines in its vicinity as this provides robots with more space and freedom for positioning. Requirement 2 can be implemented by interpreting the ball as a light source. The shadows consequently thrown by the opposing robots are viewed as anti-features from which our robots are repelled. This prevents robots from positioning themselves behind opponents which would make it easier for the opponent to intercept the ball. Requirements 3 and 4 also indicate that both ally robots and the opponent’s robots should be regarded as repelling anti-targets. Tests in the simulations have shown that the force exerted by enemy robots should be stronger than that of the team members. The total force resulting from the individual forces acting on the robots causes them to drive away from opponents and team members while positioning them close to the opponent’s penalty area, as illustrated in Fig. 2.

**Discussion** A major challenge in employing this method is adjusting the weightings of the individual forces. If individual weightings are too strong, they dominate so that the other factors are no longer sufficiently considered, which in turn leads to sub optimal positioning. Similarly, weightings that are too weak also lead to poor positioning, as factors are neglected and therefore not all the requirements can be met. Adjusting the weights is a time-consuming process in which values are changed manually and assessed in a simulated environment. As this is done without standardized test cases, the results are also subject to a strong human bias. Future applications of this approach should involve the creation of standardized test scenarios for a more objective evaluation. A particular challenge is the weighting of the opposing shadows. The weightings should ideally be as high as possible to minimize the risk of an enemy’s interception. However, the dynamic nature of the robots and ball as well as the associated rapid changes of the shadows result in uncontrolled movements and the robots



**Fig. 2.** Positioning of offensive robots using the FDP algorithm. Positioning is performed by forces acting on the robot from the shown features. The force vectors depicted in this image are not true to scale.

stray far away from the opponent’s penalty area. Additionally, robots frequently become stuck in local optima between two shadows. Another issue is the disregard of the ball’s distance to the goal. Given that robots generally position themselves near the opponent’s penalty area, the distances that the ball has to cover during the pass can be quite extensive. This complicates ball reception and increases the risk of a successful interception. While defining the ball itself as an attracting feature may help to resolve this issue, it also introduces the challenge of additional parameters. This would therefore further complicate the adjustments and optimization of the weightings. Despite these problems, the approach has proven to be effective in achieving good positioning of the robots. The approach may be particularly suitable for newer teams due to its relatively simple implementation. However, to further increase the effectiveness of the robots, other approaches should be used instead, as presented below.

### 3 Restricted Radial Positioning (RRP)

In order to tackle the problems of the FDP approach, an alternative was developed and assessed at the 2023 Schubert Open Tournament in Crailsheim. The basic principles remain the same: The robots should avoid opponents and move into open space. To prevent robots from getting stuck in local minima, FDP is to be replaced by a global search. Depending on the resolution of the playing field, this can require an immense amount of computing power as a large number of positions have to be evaluated for their suitability. Many of these evaluated positions can be discarded, for example if they are too close to or too far away from the ball. In the former case, a pass would not make sense as it would not result in any significant progress. The second case can be ruled out, because as explained in the previous section, long-distance passes are difficult to execute and increase the risk of an interception by the opponent. Therefore, the possible positions of the secondary attackers are limited to a circle around the ball. Various positions on this circle are evaluated in terms of suitability and then selected. The exact procedure is explained in the following section.

**Algorithm** As the permitted positions are limited to a circular shape around the ball, it is advantageous to display them in polar coordinates. The radius is set to approximately two to three meters, a distance that is considered optimal for advancing toward the opponent's goal and allowing the ball to be received reliably.

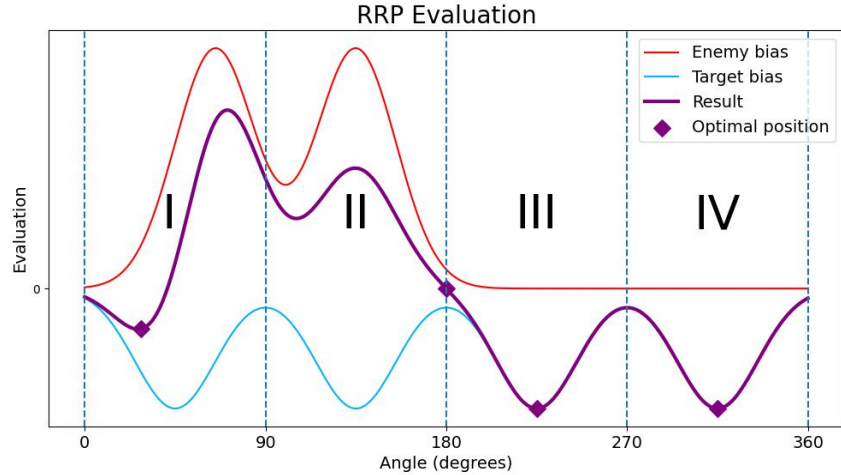
These individual positions can be displayed as an array, with the index corresponding to the angular position in relation to the ball. Notably, the radius is irrelevant in this context as it remains consistent for all positions and does not provide any additional information. Subsequently, values signifying the suitability of the respective position should be assigned to each element of the array. This array can then be used to identify local and global optima and thus influence the positioning strategy.

As in the previous section, it should also be possible with this approach to define features which robots are attracted to. More than one attracting feature needs to be chosen, so that the optimal position does not converge into a single point. This is accomplished by assigning negative values at the corresponding positions in the array. One possible choice of such points is in the middle of the individual quadrants, specifically at angles of  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$  and  $315^\circ$ . These angles indicate the preferred position of the robots. The center of the respective quadrant is selected to give the robots as much freedom of movement as possible in both directions if, for example, an enemy robot needs to be avoided. These features should also affect their neighbors. This is achieved by smoothing the resulting array using a Gaussian filter kernel [5]. Smoothing means that positions close to the defined features are also considered to be favorable, while those situated further away are being avoided.

Building upon this methodology, we can employ the same principle to define anti-features which represent positions that should be avoided. Primarily,

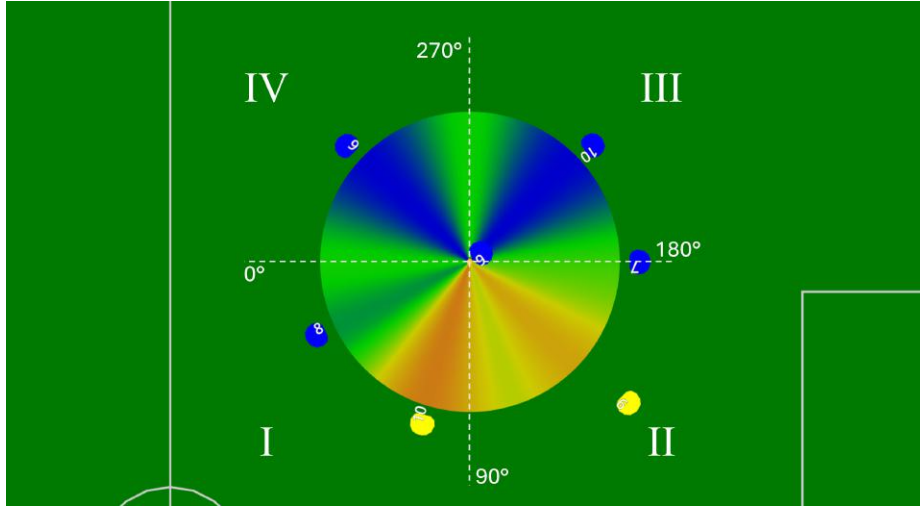
this category encompasses the enemy robots as previously described in the assumptions. Only robots that are close to the ball have to be regarded. Those at greater distance have negligible impact on passing success. As a general rule, robots positioned beyond 1.3 times the defined circle radius are disregarded in subsequent considerations, since it is unlikely that those robots will be able to intercept the pass in time. Similar to the process outlined for target positions, the opponent positions are first converted into polar coordinates. Subsequently, positive values are assigned in a separate array at these opponent positions, signifying locations that should be avoided by our robots. As before the array must then be smoothed using a Gaussian filter kernel to optimize avoidance strategies and minimize potential interference from opponents.

This procedure yields two arrays: one specifies the preferred positions, while the other identifies positions to be avoided. The element wise sum of these arrays results in a function that characterizes the suitability of each position. Lower values in this function indicate a higher suitability for the positioning of an offensive player.



**Fig. 3.** Evaluation function generated by the RRP algorithm is based on the scenario illustrated in Fig. 4. The optimal position is determined by locating the minimum of the result function in each quadrant. The result function is calculated by adding the enemy bias function and the target bias function.

To prevent the congregation of all robots at a global optimum, a restriction of one robot per quadrant is imposed. Additionally, a preference is given to quadrants that are closer towards the enemy’s defense area. Other quadrants are considered only if enough offensive players are available. When the ball is near the opponent’s penalty area, quadrants in the direction of one’s own half may be



**Fig. 4.** Resultant positions identified by the RRP algorithm through the utilization of minima within each quadrant of the evaluation function depicted in Fig. 3. This function is visualized as a radial heat map.

preferred instead. This adjustment facilitates the execution of back passes and subsequent reflex shots on the opponent’s goal.

Tests have shown that due to the global search in the quadrant, the optimum position may be in an unstable state if two or more positions have a similar score. Introducing the current robot position as an additional attracting feature resolves the issue by inducing a hysteresis-effect. This ensures that a new position must be a significant improvement over the robots’ current position. This prevents the position from being changed too frequently. Otherwise, the robot could temporarily be unavailable for a pass during the transition between positions. To apply such a hysteresis effect an additional array must be defined in which the robot’s current position is given a negative value. Analogous to the previous features a Gaussian filter must be applied.

**Hyperparameter Tuning** It follows from the algorithm described that a total of three features are being used: the target position of the robot, the current position of the robot and the opponent positions. Each of these features is represented by a Gaussian curve, which in turn contains two essential hyperparameters: the height of the peak and the variance. The height of the peak signifies the magnitude of the feature’s impact, while the variance characterizes its influence on neighboring points. The values for each parameter must be chosen appropriately in order to achieve good positioning. Analogous to FDP, manual adjustments were made during the optimization process and manually validated in a simulated environment. Moving forward, it is recommended to define standardized test scenarios for optimization purposes. This optimization approach resulted

in the following parameter set used at the 2023 Schubert Open Tournament in Crailsheim:

Parameter	Peak Height	Variance
Target Position	0.5	20°
Current Position	0.1	20°
Enemy Position	1.0	15°

**Discussion** This algorithm underwent testing during a competitive event at the Schubert Open Tournament 2023, followed by a comprehensive evaluation. As stated earlier, the primary motivation behind the described algorithm is to confine the robot’s positions, ensuring their proximity to the ball. The objective is to enhance reliable ball reception, in order to gain progress towards the opponent’s goal through passing, while minimizing the likelihood of an interception. As the games have shown, this approach is effective in maintaining the desired distance from the ball. However, one notable observation was that the strict limitation on robot positioning unintentionally led to sub optimal placement on the field. While the robots optimize their chances of successfully receiving the ball, the strategic aspect of posing a threat to the opponent from their positions is not considered. Similar to FDP, no assessment is made to whether the chosen positions can lead to potential shots at the goal. Although advancing close to the opponent’s goal generally improves the chances of scoring, this factor alone is not sufficient to evaluate the position. It fails to account for other critical factors such as a clear line of sight towards the goal. To resolve the conflict between optimizing the chance of receiving the ball and posing a threat to the opponent, additional freedoms in positioning are necessary, which are not feasible with the described approach. One solution would be to define multiple circular paths and incorporate the goal probability as an additional factor into the positions. However, this idea was discarded in favor of a new concept, which is presented in the following chapter.



## 4 Octopus Algorithm

In the previous sections, FDP and RRP were presented as two approaches that can be used to position offensive robots. Practical use has shown that FDP gives the robots too much freedom, resulting in poor positioning. Although RRP solves this problem, the restrictions on the robots are too large, which also result in poor positioning. In order to combine the advantages of both concepts the Octopus Algorithm was developed.

**Motivation** A significant change in the lhubots soccer game strategy involves the introduction of decision trees, which have not been utilized in previous approaches. With the aid of decision trees, it becomes possible to simulate and evaluate a variety of passes and pass combinations, determining which options offer the highest probabilities of a successful goal shot while minimizing the risk of interception by the opponent. This evaluation not only considers multiple pass combinations but also short distance passes. This approach aims to replace the purely reactive playing style of the previous two years with a proactive planning approach [3,4].

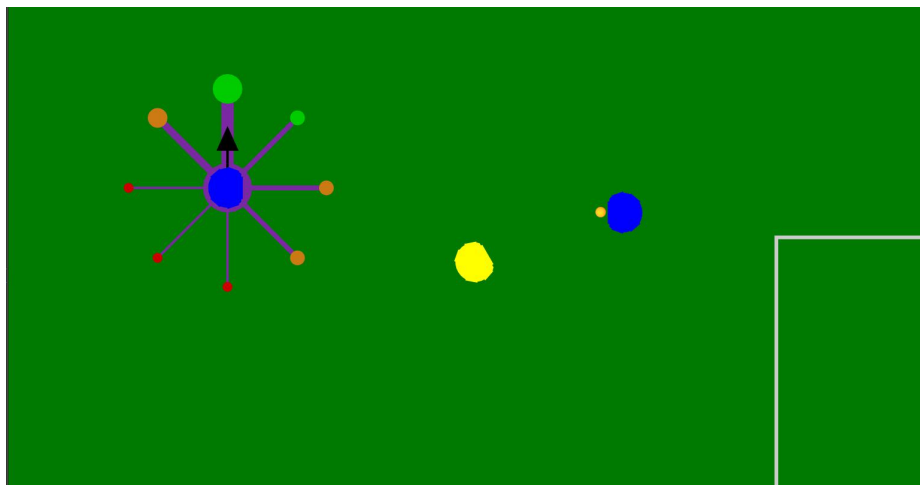
This shift in the overall game strategy also necessitates the development of a novel concept for player positioning. Notably, not all offensive players need to be controlled, as the decision tree already evaluates potential pass sequences and, if necessary, defines the location at which a robot should receive the ball. Robots situated in a sub-optimal position will not be part of the pass sequence and should consequently move to enhance their position. For this purpose, algorithms such as FDP or RRP as described above, could be employed. However, due to the issues associated with these algorithms as well as the emerging requirement for continuous movement, a novel approach has been developed, which will be introduced in this section.

**The Basic Octopus Algorithm** The *Octopus Algorithm* is a local optimization search. Unlike RRP, it does not seek an optimum in the vicinity of the ball but rather in the region surrounding the robot. To achieve this, points are generated at 45° intervals with approximately 0.5m distance from the robot. These points form the so-called *Tentacles* of the *Octopus*, that are used to continuously scan its surroundings, as demonstrated in Fig. 5. During the scanning process, the algorithm estimates the success probability of a pass and goal shot at each position, which is discussed in more detail in the following paragraph. These two factors are multiplied to obtain the overall evaluation of that position.

To calculate the success probability of a pass, the first step is to determine the action radii of the opponents. These indicate the range within which the opponent's robot could maneuver up to the latest possible interception time. The action radii are calculated with a model priorly trained using machine learning. Throughout the training phase, a regression model is fine-tuned based on factors such as friction parameters of the carpet, shot execution times based on the relative positions between the ball, ball carrier and target, and the measured

maximum speeds of opponents. Subsequently, a line is drawn from the ball to the target position. The intersection of the line with the opponents' action radii is utilized to estimate the probability of the opponent successfully intercepting the shot. If there are no intersections between the line and the action radii, it can be assumed that the opponent is unable to reach the shot trajectory in time. Conversely, if significant overlaps occur, the likelihood of success is considered lower. This method can also be extended to calculate the goal probability by evaluating shots to various points on the goal line. The goal probability and the optimal target point are derived from the point with the best evaluation. As previously explained, for each Tentacle, the pass and goal probabilities are determined and combined into an overall score through multiplication. Subsequently, these scores are compared, with the highest score determining the direction for the robot to move in. This aims to optimize both the robot's chance of successfully receiving the ball and its potential threat to the opponent. Since the robot is only aware of the evaluations of points in its vicinity, but not of the evaluation of its current location, it is compelled to move away from its current position. This behavior is undesirable in RRP, as the robot might become unavailable for a pass during this movement. However, this implication does not need to be considered for the Octopus Algorithm, as the decision tree has already assessed that the robot should not be targeted for a pass at that moment. Consequently, temporary unavailability poses no major concern, allowing the robot to improve its position for potential future passes. However, a significant challenge of this approach lies in the fact that, much like FDP, there is no guarantee that the robot will remain in a specific region. The resulting challenges have already been elucidated in section 2. To address this issue, so-called *Aquariums* are defined, which represent regions that the robot is not allowed to leave. These are discussed in the following section.

**Defining Aquariums** As the experiences with the FDP have demonstrated, it is crucial to be able to restrict the position of the robots. Simultaneously, as evidenced by RRP, overly stringent limitations can be counterproductive, as robots may no longer have the necessary freedom to position themselves effectively. Therefore, a compromise is sought, allowing robots to move freely within a box-shaped zone while prohibiting them from leaving that region. In the following, these zones are referred to as *Aquariums*. To confine the movements of the robots to the Aquarium, it suffices to set the Tentacle value to a negative value once it extends beyond the Aquarium boundaries. It can generally be assumed that at least one Tentacle is always within the Aquarium, allowing for the selection of a direction at any time that keeps the robot within the permitted zone. An exception arises when there is a change in assignment to the corresponding Aquarium, potentially causing all Tentacles to be outside the Aquarium. In such instances, a movement command is issued to reposition the robot at the center of the Aquarium.



**Fig. 5.** Octopus Algorithm Visualization: Tentacles are depicted by purple lines, with their widths indicating the chance of pass success. Colored circles represent goal probabilities at various positions, where green circles denote higher probabilities and red circles denote lower probabilities. Additionally, a black arrow indicates the optimal direction of movement, determined by the highest scoring Tentacle.

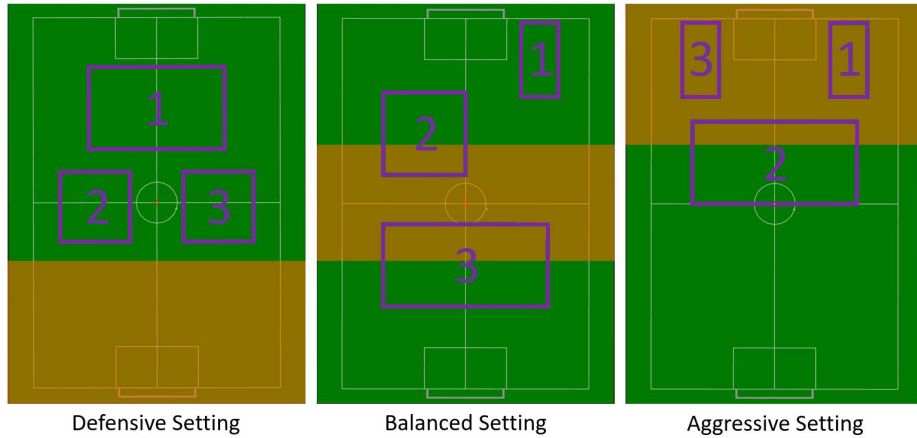
The positions of the Aquariums should be determined according to the current state of the game. As the decisive criterion the current ball position is used. Depending on the ball position, three cases can be classified:

1. Defensive setting: The ball is close to our goal
2. Balanced setting: The ball is in the middle of the field
3. Aggressive setting: The ball is close to the opponent's goal

For each of these cases, three Aquariums must be defined, as there can be up to three offensive players at any given time. This will change with the transition from Division B to Division A, since the higher total number of robots also affects the number of offensive robots and therefore the number of Aquariums needed. Each Aquarium is also given a priority, indicating the order in which the Aquariums should be occupied. When there are fewer than three offensive players, only Aquariums with the highest priority are occupied, leaving the rest vacant. The prioritization of Aquariums is typically based on their proximity to the opponent's goal, recognizing the strategic advantage of positioning offensive players closer to the scoring area. While this approach aligns with a potential concern in FDP regarding the riskiness of long passes, it is noteworthy that the decision tree overrides the option of making such passes to the corresponding robot. Instead, the recommendation is for the robot to position itself strategically, anticipating inclusion in a passing sequence at a later point when the ball is expected to be closer to the opponent's goal. Despite the emphasis on offensive positioning, it is also crucial to define Aquariums in proximity of the ball. This

ensures that robots are readily available to initiate a passing sequence. The task of these robots is to make themselves available to receive a pass and extend it to more aggressive robots near the opponent’s penalty area. This is particularly crucial in defensive and balanced situations, where direct passes may be challenging due to the distance involved. In the defensive scenario, the primary objective is to swiftly advance the ball into the opponent’s half to mitigate the risk of conceding a goal. However, in situations classified as balanced or aggressive, the consideration of back passes becomes integral to the strategy. Back passes serve dual purposes: Firstly, they frequently present the opportunity for a reflex shot, allowing the ball to be quickly shot at the opponent’s goal without the need for ball reception. These shots pose a significant challenge for defenders due to their requirement for rapid reactions. Secondly, back passes offer an effective method for maneuvering around an opponent’s defensive wall or redirecting the ball to the opposite side of the opponent’s penalty area, where defensive coverage may be less concentrated.

The schematic depiction in Fig. 6 outlines the strategic placement of the Aquariums. Once the size and position of the Aquariums have been determined as shown in the illustration, the robots can be assigned to the Aquariums. This assignment is described in the following section.

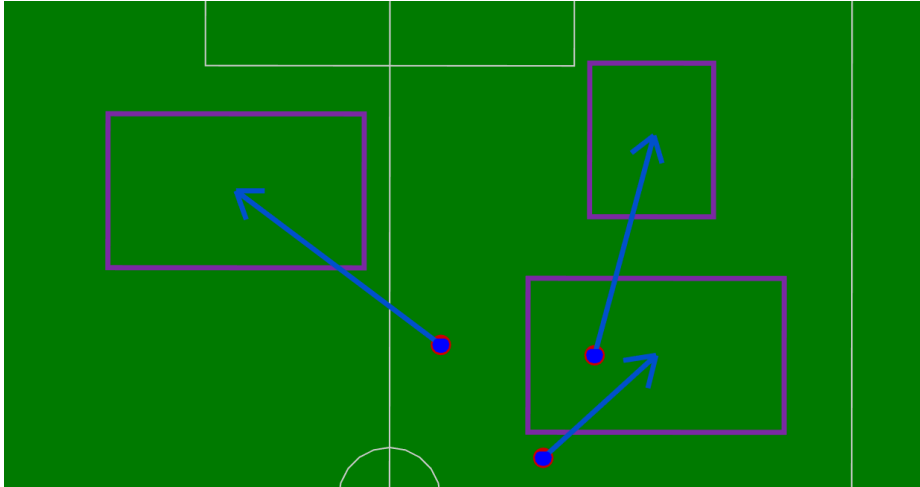


**Fig. 6.** Aquarium Assignment: Aquariums are represented by purple boxes, each labeled with a number indicating its priority, where lower numbers indicate higher priority. The selection is made when the ball position falls within the orange zone. It is assumed that the ball is located on the right side of the field; otherwise, the Aquarium positions must be mirrored accordingly. The opponent’s goal is on top.

**Assignment Optimization** To minimize travel times as much as possible, an optimal assignment of robots to the Aquariums is essential. Algorithms like k-nearest neighbors face the challenge of heavy dependence on the assignment order

[1]. Instead, the assignment is determined using the Munkres algorithm, also known as the Hungarian algorithm, which serves as a general solution method for the assignment problem [2]. The underlying objective of the assignment problem is to find the combination of assignments that minimizes total costs.

To achieve this, a cost matrix is defined, containing the costs for each task and agent. In the Octopus Algorithm, the costs are represented by the squared Euclidean distances between the robots and the center points of the Aquariums. Squaring the costs punishes longer distances, prioritizing the minimization of the longest route. This approach facilitates better parallelization of movements, ensuring the swift occupation of all Aquariums. Once the optimal assignment has been determined, movement tasks are issued to the robots corresponding to these Aquariums, as shown in Fig. 7. As soon as at least one Tentacle is within the Aquarium, the positional optimization is executed, as described in the previous section.



**Fig. 7.** Visualization of the optimal assignment using the Munkres algorithm.

**Discussion and Evaluation** The Octopus Algorithm was first deployed at the RoboCup 2023 in Bordeaux and tested against other teams. The results reveal that the algorithm effectively addresses many of the previously mentioned shortcomings of FDP and RRP. By defining the Aquariums, it becomes possible to incorporate domain knowledge and ensure a strategically advantageous positioning of the attackers. Simultaneously, this approach allows sufficient flexibility, enabling the robots to maximize their chances of a successful goal scoring attempt and a successful pass reception. This results in significantly improved player positioning compared to previous competitions where FDP and RRP have been utilized.

## 5 Conclusion and Future Work

This paper presented three distinct approaches that the luhbots soccer team has developed and deployed over the preceding two years. Among these, the Octopus Algorithm has proven to be the most effective in achieving optimal positioning for offensive players. Through this algorithm, robots actively enhance the probability of successful ball reception and subsequent goal scoring attempts, all while integrating domain knowledge to constrain their positioning. Particularly when combined with decision trees, the Octopus Algorithm allows to swiftly and effectively exploit opposing defenses. This was demonstrated at the RoboCup in Bordeaux, often leading to early game terminations with scores of 10:0, particularly evident in the preliminary rounds.

As the luhbots soccer team aims to participate in Division A for the upcoming season, the primary focus of the current development efforts is on scaling the algorithms to accommodate 11 robots, as opposed to the current 6. A crucial step in this process involves the transition from a polling-based architecture to an event-based strategy to enhance overall computational efficiency.

The core functionality is intended to be retained, indicating the likelihood of the Octopus Algorithm being deployed again at the Schubert Open Tournament 2024 and the RoboCup 2024.

**Acknowledgements** Special thanks to our supporting institutes, the Institute of Automatic Control (irt) and Institute of Mechatronic Systems (imes). We appreciate the financial and educational support. Also, we want to mention all RoboCup 2023 SSL teams' fair play. They were incredibly forthcoming and gave us advantageous insight into their research. Finally, grateful for being part of the Leibniz University Hannover, we also want to thank our university for supporting several different student associations which allow us to exchange interdisciplinary ideas.

## References

1. Cunningham, P., Delany, S.J.: k-nearest neighbour classifiers - a tutorial. *ACM Comput. Surv.* **54**(6) (jul 2021)
2. Munkres, J.R.: Algorithms for the assignment and transportation problems. *Journal of The Society for Industrial and Applied Mathematics* **10**, 196–210 (1957)
3. Seegemann, L., Zeug, F., Ebbighausen, P., Waldhoff, L., de Vries, T., Sukkar, M., Känner, M., Weber, J.O.: luhbots soccer team description for robocup 2022 (2022)
4. Seegemann, L., Zeug, F., Ebbighausen, P., Waldhoff, L., Westermann, M., Knackstedt, S., Känner, M., Fücksel, T., Hart, R., Pahl, T.: luhbots soccer team description for robocup 2023 (2023)
5. Zhang, X.: Gaussian Distribution, pp. 425–428. Springer US, Boston, MA (2010)