Balancing Gains, Risks, Costs, and Real-Time Constraints in the Ball Passing Algorithm for the Robotic Soccer

Vadim Kyrylov

Simon Fraser University – Surrey
Surrey, British Columbia V3T 2W1 Canada
vkyrylov@sfu.ca

Abstract. We are looking for a generic solution for the optimized ball passing problem in the robotic soccer which is applicable to many digital simulated sports games with ball. In doing so, we show that previously published ball passing methods do not properly address the necessary balance between the anticipated rewards, costs, and risks. The multi-criteria nature of this optimization problem requires using the Pareto optimality approach. We propose a scalable and robust solution for decision making, as its quality degrades in a graceful way once the real time constrains are kicking in.

1 Introduction

1.1 Ball Passing Algorithms: State of the Art

Passing the ball to a teammate is a critically important player skill in many sports games with ball. Early RoboCup scholars have developed two reasonably good algorithms for the simulated soccer [1, 2]. In both the soccer agent chooses values of the direction of the kick and its force. In [1] the anticipated outcome of passing ball is evaluated with two heuristic indicators: (1) the tactical value of the end point and (2) the likelihood of that the receiving teammate will intercept the ball. This algorithm is searching for both direct and leading passes including passes to self. The tactical value is the only criterion for selecting the best option; the likelihood of success is used as a constraint. Although this method has proved to be rather good, it neglects risks such as the possible proximity of other opponents to the anticipated interception point. One more shortcoming is the requirement that the ball should be always intercepted by the receiver in the minimal time. Indeed, this may result in lost opportunities in executing leading passes when the ball is sent to the point of the field still reachable by the teammate and having better tactical value.

The algorithm implemented in [2] appears to be more sophisticated, as it is taking into account the opponent player congestion in the vicinity of the ball destination. Also are considered ball travel distance, opponent goal scoring opportunity if the pass is successful, and the possible outcomes if the ball would not be intercepted as intended. The decision is made by deliberating on 5 options for each receiving teammate: direct pass, leading pass, pass to the expected location of the teammate,

G. Lakemeyer et al. (Eds.): RoboCup 2006, LNAI 4434, pp. 304–313, 2007.

pass to a point near teammate having low congestion, and pass along a low congestion line. Each alternative is evaluated using 9 performance indicators. With the purpose of making a choice, these indicators are analyzed using a decision tree.

Even more advanced ball passing algorithm with player collaborating using aural messages was recently reported in a short paper [3]. As this algorithm is all based on a decision tree, it is possible that some indeed good ball passing options could be overlooked. This is a general shortcoming of decision trees; in what follows, we discuss this in more detail.

1.2 Unresolved Issues and Research Objectives

In this paper, we address three issues.

1. No benchmark. The existing algorithms are collections of sophisticated heuristics; it is still unknown to what extent they could be improved and what the benchmark solution is.
2. Smoothly balancing rewards and risk. We believe that implementing a continuous spectrum of risk-taking vs. risk aversive strategies by the soccer agent is highly desirable. However, in the existing methods this balancing does not render itself as a controlled feature.
3. Avoiding possible conflicts with the real-time constraints. Reduction of required computations in existing algorithms can normally be done by removing some branches in the decision tree. That may result in abrupt loss in the quality of decisions.

We resolve these issues using the multi-criteria decision analysis (MCDA). In doing so, we are pursuing the following objectives.

- Developing a theoretical framework for a totally optimal ball passing algorithm that could serve as a benchmark. We want this solution to be generic and thus reusable. This intention is standing in a concert with other RoboCup scholars looking for generic solutions [4].
- Fully identifying rewards, risks, and costs involved in passing the ball and demonstrating how they could be balanced in the proposed framework. We wish to offer a way to implementing a continuous spectrum of risk-taking and risk-aversive attitudes by the soccer player.
- Addressing the real-time constraints. We want to propose a truly scalable solution with just one parameter which determines the amount of the required computations. We also want to design a robust ball passing algorithm that would be resulting only in a gradual loss of the decision quality if we are forced to bypass some computations.

2 Rewards, Risks, and Costs in Ball Passing

Prior to developing the optimal decision making algorithm, we identify the presumably complete list performance criteria that govern the decision to pass the ball.

In doing so, we slightly modify the ball passing problem formulation as compared to [1, 2]. In our case, player with the ball considers all possible points (x, y) in the field and must decide to which point he should send the ball now and determine the ball speed in the end point. This end speed affects the probability of the successful interception by the receiving teammate; it also determines the ball travel time and thus the incurred risk. The decision is made by comparing performance indicators calculated for different ball passing options.

Once the passing player has made his choice of the point and of the ball end speed, he is able to determine the kicking force and direction, which are the actual decision variables. If the required kicking force exceeds the available limit, the point is just removed from the consideration. Likewise, points are eliminated if the perceived risks are prohibitively high.

Each remaining potential destination point for pass is assigned a vector criterion having continuous values of its m components, which are the performance indicators. So there is a two-dimensional decision space (*kicking_force, direction*) and an m-dimensional criterion space. For the analysis, we make two modifications. First, we replace the decision space by a three-dimensional one (*x, y, end_speed*) with only two coordinates being independent; this space is much easier to visualize. Second, in order to make our algorithm scalable when the real-time constraints are present, we replace the continuous decision space by a discrete one.

We split the decision criteria in three categories: (1) gains, (2) risks, and (3) costs.

Gains. We see two gains, or rewards, from passing the ball; we wish to maximize both.

Both are similar to the indicators used in [1] and [3], which served their purpose very well. The first is the tactical value of the point (x, y) where the ball will be sent to. This function encourages sending the ball close to the opponent goal and discourages destinations near own goal. The second reflects the chance to score the opponent goal from the ball destination. Its value depends on the anticipated number of opponent players between the opponent goal and the destination point (x, y) of the pass.

Risks. As proposed in [1], the risks involved in ball passing all are defined as soft constraints. We further improve this idea by dropping the requirement that the receiving player is intercepting the ball in minimal time. Rather, we assume that he must be chasing the ball if necessary. Hence we have more risk factors than our predecessors.

1. Opponent may reach (x,y) before the teammate. The risk function $r_1(x,y)$ is the time difference between the arrivals of the fastest teammate and the fastest opponent to this point.
2. Ball can be intercepted by the opponent on its way to (x,y). The risk function $r_2(x,y)$ is the time difference between the intended arrival time of the ball in (x,y) and the earliest time when it can be stolen by the opponent.
3. Teammate may be too late in point (x,y) after the ball rolls by. So the risk function $r_3(x,y)$ is the time difference between the arrivals of the teammate and the ball. However, this risk increases if the ball is moving in (x,y) too fast which is making it difficult to intercept.
4. Too many opponents may get close by. The risk function $r_4(x,y)$ is the time difference between the arrivals of the ball and the second fastest opponent in (x,y).
5. If the teammate fails to intercept the ball, it may cross the field boundary. The risk function $r_5(x,y)$ is minus the time remained until the ball crosses this line after bypassing (x, y).

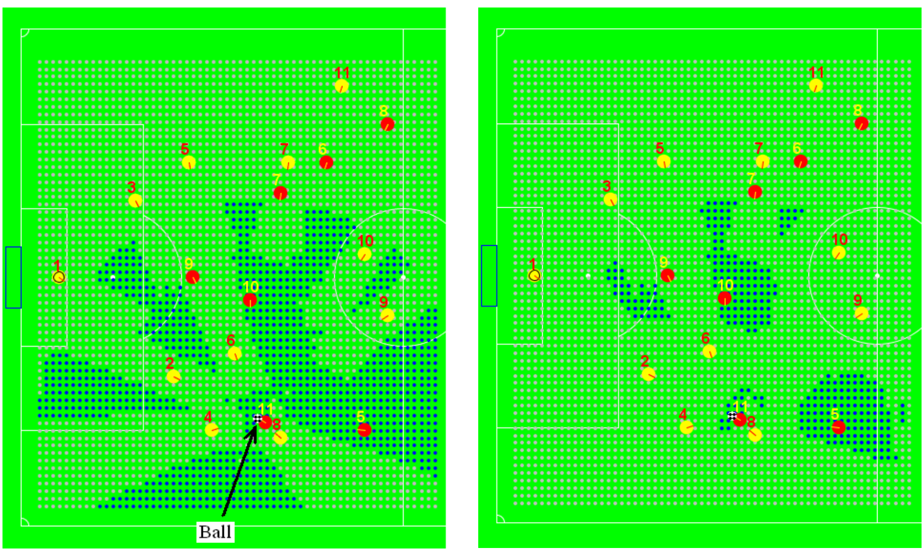
6. The receiving player may have too low stamina to chase the ball. The risk function $r_6(x,y)$ is time when the receiving teammate reported low stamina less current time.
7. Ball may not reach the destination point at all, as (x, y) is too far away for given initial ball speed. As the ball movement is distorted by noise, the actual maximal ball traveling distance may differ from the calculated theoretical one, $D_{\max}$. A soft constraint $r_7(x,y)$ is used to reflect this risk.

We want to minimize each of these seven risk factors. For convenience, they could be scaled so that they all are taking values in, say (0, 10).

Costs. The cost factor is the time required for obtaining the anticipated rewards, which we want to minimize. Taking this in consideration makes sense because the precision of the situation prediction substantially decreases with the forecast time. This criterion would be discouraging too long passes if, given all the rest conditions equal, shorter ones exist.

Concept demonstration. With the sole purpose of the concept demonstration used throughout this paper, we have designed an example with three simplifications. (1) Decision space is further reduced to determining the pass direction only; end speed in the destination point is a fixed parameter of the algorithm. (2) Only the tactical value of the end point is used as the reward. (3) Risk and costs merged in just one parameter by applying heuristic rules.

This allowed using two-dimensional displays for the visualization. The full-scale algorithm is treating all criteria separately.



(a) applied risk factors 2 and 7 only (b) all seven risk factors applied

Fig. 1. Screenshots of the software tool for analyzing the soccer player tactics. Of the original 3600 points, most have been eliminated because the anticipated risk is inappropriately high.

Fig.1 shows a grid of 3600 points considered as candidates for passing the ball by player 11 from the right-hand team. Most are eliminated because the risk is too high. These points are shown in light gray; the darker points are the remaining alternatives. Player 11 must select the best one based on the vector of performance indicators available for each point.

We wish this decision to be optimal in some sense. This sense is the Pareto optimality.

3 Applying the Pareto Optimality Principle to Ball Passing

Pareto optimality, first originated in economics, is now a standard principle for solving vector optimization problems with conflicting criteria [5]. In what follows, we will replace the reward function with the negation thereof; thus we want all our criteria to be minimized simultaneously. In the general case, though, simultaneous minimization cannot be achieved. The Pareto optimality principle only offers a method for substantially reducing the set of decision alternatives by identifying among them the set of so-called non-dominated alternatives; altogether they are making the Pareto set, or the Pareto frontier.

By definition, the criteria vector $\mathbf{v}_i = \{v_{i1},\ldots,v_{im}\}$ is **dominated** by vector $\mathbf{v}_j = \{v_{j1},\ldots,v_{jm}\}$ if the following condition holds:

$$\forall k \{v_{ik} > v_{jk}\}\,. \tag{1}$$

This means that $\mathbf{v}_j$ is located inside the cone in $\mathbb{R}^n$ with the vertex $\mathbf{v}_i$, the sides of this cone being parallel to the coordinate subspaces $\mathbb{R}^{n-1}$. By definition, the Pareto set is the subset of non-dominated alternatives, i.e. whose cones do not contain other alternatives. The Pareto set is not necessarily convex, nor is it in the general case even connected. The computational complexity of determining the Pareto subset in the finite set with N elements is $O(N^2)$.

The meaning of a non-dominated alternative $\mathbf{v}_j$ is that outside the Pareto set there is no another alternative that outperforms $\mathbf{v}_j$ simultaneously by all criteria; at least one criterion value is worse, anyway. From this follows that the optimal decision should be sought within the Pareto set; all the rest alternatives could be eliminated as they are all inferior.

In Section 4, we will be also using a weakened version of the dominance relation, which is called ε-domination [5, 6]. The set of non-ε-dominated points is referred to as ε-Pareto set. Elements lying outside this set are having at least one criterion that is worse by more than ε.

Noteworthy that, eliminating ball passing alternatives before identifying the Pareto set, as it has been done in the existing algorithms, may result that some of the Pareto optimal points would be apparently removed without even evaluating thereof. This is exactly what may happen in decision trees. Unless the decision conditions are designed so carefully that any eliminations do not affect the Pareto set, there is no guarantee that the decision tree yields optimal solution to the problem in all cases. However, the trouble is in that such a decision tree is difficult to design, and for each new applied problem this must be done over and over again. On the other hand, the Pareto optimality principle offers a general solution.

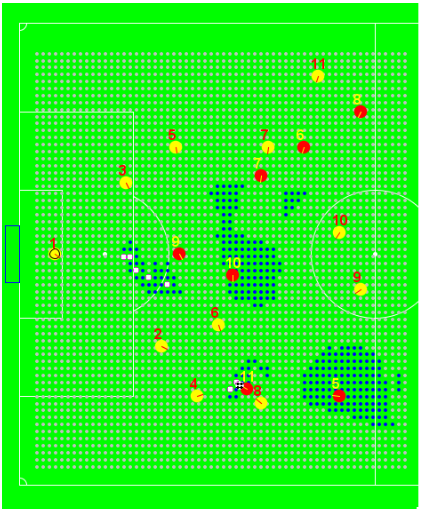


Fig. 2. Situation in Fig.1(b) with the points making the Pareto set shown in white

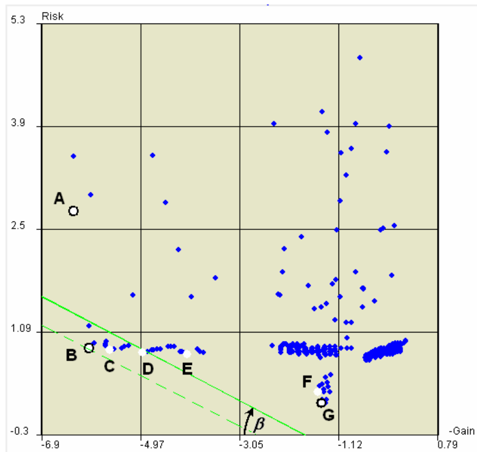


Fig. 3. The ball passing alternatives in the criterion space. Points in the Pareto set shown in white.

The Pareto set of the alternatives that player 11 in Fig.1(b) should be indeed choosing from is shown in Fig 2. This example suggests that either the leading pass to teammate 9 should be executed (five slightly different options), or player 11 should leave the ball for himself, i.e. execute so-called fast dribbling (two options). Passes to teammate 10 are not in the Pareto set. Note that some points near player 9 cannot be reached in minimal time; yet they are better for scoring the goal. Fig. 3 shows the situation as it occurs in the criterion space.

The MCDA theory leaves the final choice of the single alternative from the Pareto set up to the decision maker. In our case, however, it is the algorithm developer who must formalize the player preferences which could be used for searching the Pareto set. This search is exactly about *balancing* the rewards, risks, and costs; in what follows, we explain this idea.

A naïve approach suggests merging all criteria in just one and applying commonly known single-criterion optimization techniques. For example, one can use the utility function U of the decision variables (x,y), which is the weighted sum of risk *Risk* and gain *Gain*:

$$U(x,y) = -wGain(x,y) + (1-w)Risk(x,y), \tag{2}$$

where w is the positive weight, $0 \le w \le 1$; it reflects the importance of *Gain* for the decision maker, as compared to *Risk* whose weight is thus $1\text{-}w$. (Note the minus sign before *Gain*).

To find the solution, function (2) must be minimized. Equation $U(x,y) = c$, where c is some constant, in the criterion space represents the slant strait line shown in Fig.3. Search for the optimal solution in this case would be moving this line towards the origin by decrementing *c* until the line (shown in the dashed style) intersects with just one decision alternative **B**. Presumably, this would be the optimal, balanced solution sought.

Unfortunately, this simple approach does work only when the Pareto set is convex [6]. If non-convexity is in place, some elements of the Pareto set would be never rendered as the solutions to the optimization problem, no matter what values the decision maker assigns to *w*. However, this is counter intuitive, as each point in the Pareto set is the best option for some combination of the decision maker preferences. In our example we can scan all possible preferences by varying the weight *w* in the range $0 \leq w \leq 1$. Note that, as $\beta = \tan(w)$, this parameter determines the angle β of the line $U(x,y) = c$ (in Fig.3, w=0.335). For all possible weights, this would render only three points, **A**, **B**, and **G** of total seven available in the Pareto set (marked with black circles). The rest four would be never returned as solutions. This just illustrates the fact that with the multi-modal criteria functions which we are dealing with in the robotic soccer, a different way to finding the balanced optimal solution should be taken.

4 Searching the Optimal Ball Passing Decision in the Pareto Set

The different way is applying more sophisticated methods for searching the Pareto set that can work with non-convex problems. As there is a plethora of such methods, we will demonstrate just one, developed by the author of this paper. The method is called '*the randomized sequential elimination of the poorest alternatives*'. Because it does not rely on any information about the criteria functions, it is applicable to any MCDA problem with a finite Pareto set. This nicety, comes at rather low cost: with the total of *K* elements in the Pareto set, the computational complexity of this algorithm is $O(K^2)$. (Note that $K \ll N$, where *N* is the number of points in the set of the alternatives before any eliminations.)

The key assumption is that each criterion has its relative non-negative weight $w_1,\dots,w_n$ whose sum is 1. They are reflecting the preferences of the developer of the decision making algorithm. In what follows, the set of weights is regarded as a probability distribution.

The algorithm has *K*-1 iterations, eliminating at a time one element from the Pareto having the worst value of *j*-th criterion. The criterion index *j* is randomly selected with probability w_j. Therefore, more important criteria tend to be chosen more frequently than the less important ones. The process ends when only one element remains in the working copy of the Pareto set. This is the approximation of the balanced, optimal solution to the problem. With *K* increasing, this approximation converges to the precise optimum.

The scarce discrete subset of the real infinite Pareto set like shown in Fig.3 yields too rough approximation. (Note this is what has remained from the original 3600 points.) Because further increasing the total number of points is not an option, we are using the ε-dominance relation instead of the strict one. This concession can be justified by that the criteria values are calculated with some errors, anyway. As we can guesstimate the standard deviation of these errors, we can choose ε of the same order of magnitude. As the ε-Pareto set will include near-optimal alternatives, it will be much denser. The application of the random elimination in this case would result in much smaller volatility of the solution.

Fig.4 and 5 give the idea of what happens to the situation in Fig.2 and 3 once ε-dominance is applied; the player indeed gets much more options to chose from. The cost for this is a slight deviation from the strict Pareto optimality and a longer, yet not prohibitive, computation time. The benefit is the better robustness of the solution search algorithm.

Assigning weights to the criteria in the proposed framework has a transparent meaning. Unlike using weights to sum up criteria similar to (2), in our method there is no way for that a higher value of one criterion apparently compensates for the insufficient value of the other. So the proposed technique allows easily modeling the continuous spectrum of risk taking and risk aversive attitudes of the decision makers. This is made possible by changing weights.

So far we have been using the example with the weight of *Gain* 0.335, i.e. *Risk* had about twice as much higher weight. This results in the risk-aversive decision shown in Fig. 4 and 5. Player 11 prefers to pass the ball to teammate 10 rather than taking the chance of sending the ball to teammate 9 whose position is much better. By changing the weight in favor of risk taking, it is indeed possible to persuade the player to pass the ball to player 9 (see Fig. 6, 7).

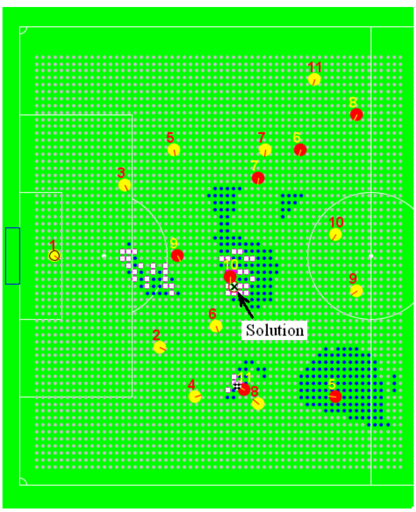


Fig. 4. Situation with the ε-Pareto set. Risky passes are avoided.

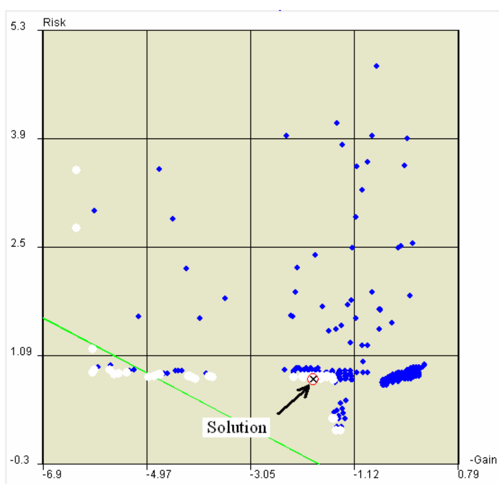


Fig. 5. The ball passing alternatives in the criterion space. The *Gain* weight is 0.335.

5 Addressing the Real-Time Constraints

As described so far, the optimal ball passing decision making algorithm in terms of computations appears to be even more demanding than the algorithms proposed in[1, 2, and 3]. In the first experiments in 2003 with our simulated soccer team *SFUnleashed* we have indeed found that the quality of decisions made by players while passing the ball non-monotonically depended on the total number of points *N*. Starting with small number of point, quality was noticeably increasing with N. Then, with greater N, we observed significantly decreased performance. Indeed, with large N the player process could not complete all required computations during one simulation cycle.

As it should be expected for a real-time system like robotic soccer, attempts to utilize all the player potential by using sophisticated optimization may be counterproductive because of the prohibitive computation time. Still we decided to find a way out so that the real-time constraints were not so restrictive. Our solution comprises two ways for the time reduction.

The first way is further reducing the number of alternatives that wittingly are not in the Pareto set; this can be done by replacing the equidistant grid (Fig.1-5) with randomly scattered points in the vicinity of each teammate (Fig.6, 7).

The second way is automatically adjusting the number of generated points N during run time with respect to the actually available time in the simulation cycle. As we know that the complexity of the whole method is $O(N^2)$, it is always possible to estimate affordable N in advance in the current simulation cycle and thus to prevent real-time constraints from kicking in. Reducing N would result in only gradually increase of the random deviations from the theoretically optimal solutions, without any abrupt losses in the quality of decisions on the average. This behavior is quite different from that of the algorithms based on decision trees whose real-time scalability is very limited. Thus the proposed algorithm is robust by design and is indeed scalable with respect to tightened or relaxed real-time constraints.

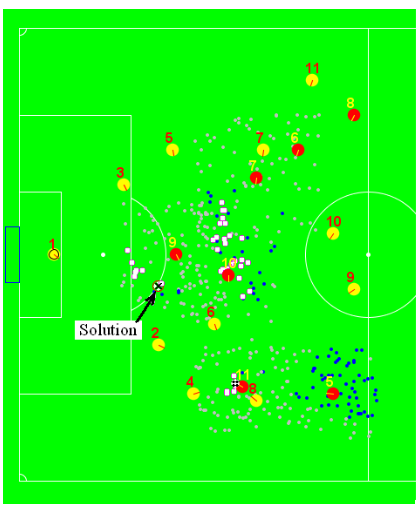


Fig. 6. Situation with 400 points randomly generated about the teammates

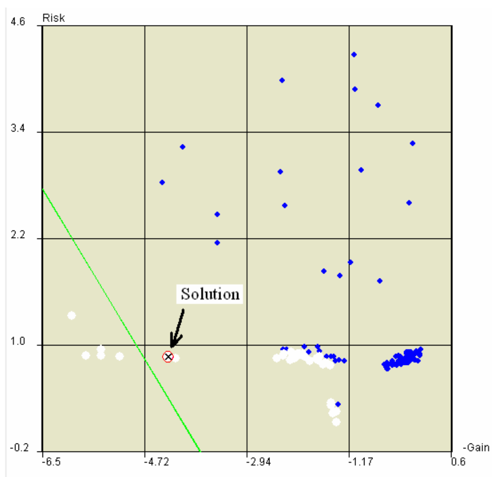


Fig. 7. The ball passing alternatives in the criterion space. The *Gain* weight is 0.614.

The full-blown algorithm is just a straightforward generalization of the simplified method illustrated in the above examples. The only difference is that instead of the two criteria function we are using all ten. The algorithms for computing these criteria have been described in Section 2; some of them are similar to that can be found in the RoboCup literature.

With the exception of particular performance criteria, the proposed optimal decision making framework is general enough to be applicable to a wide range of digital sports games with ball including all RoboCup leagues.

References

1. Stone, P., McAllester, D.: An Architecture for Action Selection in Robotic Soccer. In: Proceedings AGENTS'01, 5th International Conference on Autonomous Agents, May 28-June 1, 2001, Montreal, Quebec, Canada, pp. 316–323 (2001)
2. Reis, L.P., Lau, N.: FC Portugal Team Description: RoboCup 2000 Simulation League Champion. In: Stone, P., Balch, T., Kraetzschmar, G.K. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 29–40. Springer, Heidelberg (2001)
3. Wang, C., Chen, X., Zhao, X., Ju, S.: Design and Implementation of a General Decision-making Model in RoboCup Simulation. International Journal of Advanced Robotic Systems 1(3), 207–212 (2004)
4. Dylla, F., Ferrein, A., Lakemeyer, G., Murray, J., Obst, O., Rofer, T., Stolzenburg, F., Visser, U., Wagner, T.: Towards a League-Independent Qualitative Soccer Theory for RoboCup. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 29–40. Springer, Heidelberg (2005)
5. Ehrgott, M.: Multicriteria Optimization, 2nd edn. Springer, New York (2005)
6. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, New York (1998)