Anytime Action Selection Algorithms in Virtual Soccer

Daniil Kivarin[0000−0002−0495−9994] and Michail Panteleyev[0000−0002−6077−760X]

Saint Petersburg Electrotechnical University 'LETI', 197376, Saint-Petersburg, Russia
{dkivarin, mpanteleyev}@gmail.com

Abstract. The article is devoted to the problem of action planning in dynamic multi-agent worlds. Multi-agent worlds or systems are sets of interacting intelligent agents performing targeted actions in a dynamic environment. They are used for many applications including transportation, logistics, graphics, manufacturing etc. One of key objectives of intelligent agents in multi-agent worlds is real time planning in a constantly changing environment, because depth and completeness of the analysis of possible actions are limited and must adapt to current time limitations. The aims of the study are to consider existing approaches to action planning and to experimentally analyze their work in typical situations from multi-agent system representing the counteraction of agents in virtual soccer match. The main approach to action planning considered in this article is the model of advanced iterative action planning for real time intelligent agents. This approach is based on using any-time algorithms to avoid problems with time restrictions. The proposed algorithms and models for action planning are considered taking into account decision-making under real-time constraints. They work is illustrated with an example of a concrete situation from virtual soccer. The operating time of the presented algorithms was experimentally measured and compared with theoretical values. According to analysis, complexity of algorithms used in assessment was estimated. These results allow us to develop an action planning algorithm as an any-time algorithm so that the agent can effectively use all available time in the situation.

Keywords: intelligent agent · multi-agent systems · action planning · utility assessment · virtual soccer · robocup

1 Introduction

Autonomous intelligent agents (IAs) capable of goal-directed acting in complex dynamic multi-agent environments have been of great interest to researchers in recent years [1]. When constructing such a systems one of the key problems is an action selection in a constantly changing multi-agent worlds [2, 3].

Robocup Soccer Server is currently the most popular platform for the study IAs and multi-agent systems [4]. The platform allows to simulate team acting scenarios in real time under limited perception of the world. Therefore, this article discusses the task of evaluating the actions utility in virtual football.

In [5] the action selection algorithm based on forward simulations is presented. The approach made it possible to optimize the time for the player to turn body before kick in order to reduce the risk of the opponent stealing the ball before the kick. However this algorithm belongs to a fairly narrow scenario of the kick selection in RoboCup.

A case-based reasoning approach for cooperative action selection, based on the storage, retrieval, and adaptation of example cases is presented in [6]. A retrieval technique that weighs up the similarity of a situation in terms of the ball positional features, the uncontrollable features, and the cost of moving the robots from the current situation is proposed.

In [7] biologically inspired action control architecture as a combination of hierarchical-vertical and non-hierarchical horizontal mechanisms is considered. Motivational model of action selection for autonomous virtual humans is described in [8]. The model provides overlapping hierarchical classifier systems, working in parallel to generate coherent behavioral plans, which are associated with the functionalities of a free flow hierarchy to give reactivity to the hierarchical system.

The known approaches do not take into account the requirements of real time and do not allow to adapt the decision-making process to changing time constraints. In a dynamic multi-agent environment, the depth and completeness of the analysis of possible options for action are limited and must adapt to current time constraints.

A promising approach to solving the action selection problem in real-time is a model of advanced iterative action planning [9, 10]. In accordance with this model, the refinement of the options for action possible in the situation under consideration and the assessment of their utility are carried out within the framework of dynamically identified time constraints. The approach involves the use of anytime algorithms (ATA) [11], in which the quality of results increases with increasing execution time.

The purpose of this article is to refine the models and algorithms for actions utility evaluating in virtual soccer, as well as their analysis for typical situations that arise during a soccer match. Based on the analysis, dynamically changing (situational) factors affecting the time complexity of the utility assessment are identified. This allow to construct utility evaluating algorithm as ATA, so that agent can effectively use all the time available in a particular situation for action selection.

2 Algorithms for actions utility assessment

Let us consider the main concepts of approach according to [9, 10]:

1. Current situation S^* – the world state, at the time IA chooses next actions;
2. Set of possible agent actions $\{Act_i\}$ – actions the considered IA can perform in the current situation;
3. Set of possible situations $\{S_n^{**}\}$ – possible world states resulting from the performing of selected action by the considered IA;
4. Set of actual actors $\{AA_k\}$ – all possible IAs which can significantly affect the world state after performing of selected action by the considered IA;
5. Set of partners $\{T_m\}$ – the subset of actual actors achieving the same goals as considered IA;
6. Set of possible partners' actions $\{TAct_i^m\}$ – actions the m-th partner from set of partners can perform in the current situation;
7. Set of opponents $\{O_k\}$ – the subset of actual actors opposing the goals of considered IA;
8. Set of possible opponents' actions $\{OAct_i^k\}$ – actions the k-th opponent from set of opponents can perform in the current situation;
9. Action utility $U(Act_i)$ – the assessment shows how profitable the performing of this action in the current situation to achieve the goals of considered IA;
10. Situation utility $U(S^{**})$ – the assessment which shows how profitable this situation to achieve the goals of considered IA.

The generalized algorithm for assessing the utility of action Act_i in current situation according to [9, 10] contains the following steps:

1. Identification of sets $\{T_m\}$ and $\{O_k\}$ of actual actors in current situation;
2. Identification of sets $\{TAct_i^m\}$ and $\{OAct_i^k\}$ of possible actions for each agent from $\{T_m\}$ and $\{O_k\}$;
3. Identification of set $\{S_n^{**}\}$ of possible situations. Possible situations are predicted as world states at the time then action Act_i and all possible combinations of actions $\{TAct_i^m\}$ and $\{OAct_i^k\}$ are completed;
4. Assessing the utility $U(S^{**})$ of all possible situations from $\{S_n^{**}\}$;
5. Assessing the utility $U(Act_i)$ of action Act_i as minimum value of utilities $\{U_n(S_n^{**})\}$ of possible situations.

The generalized algorithm for determining the most useful action in the current situation according to [9, 10] contains the following steps:

1. Generating the basic set of possible agent actions $\{Act_i\}$;
2. Assessing the utility $U(Act_i)$ of all actions from the basic set $\{Act_i\}$;
3. The most useful agent action is defined as action from $\{Act_i\}$ with maximum value of utility $U(Act_i)$.

Each action has an a priory utility assessment dependent on action type and its main parameters. For example, shot for the goal is more preferred than pass, dribble forward is more preferred than dribble diagonally etc. A priory utility assessment practically doesn't require time for computation (since it is retrieved from IA's memory) which corresponds to the ATA-based approach.

After the considered IA has selected a specific action, in determining the set of actual actors, the possibility of their influence on the situation development is assessed. For example, if the IA is considering passing the ball to the left, then players who are to his left and are near to the probably pass trajectory will be included in the set of actual actors for this action. At the same time, the players who are to the right of the player who has the ball in this case cannot directly influence the situation development. So they will not be included in the set of actual actors for this action (pass to the left).

3 Implementation of actions utility assessment by any-time algorithms

To illustrate the proposed approach, let's consider the situation when the player who owns the ball is close to the opponent's goal (see Fig. 1). Designations of objects in the figure:

1. A – player, who owns the ball and assesses the utility of his next actions;
2. B – ball;
3. $T1$ – teammate of player A;
4. $O1, O2$ – players of opponent team;
5. G – goalie of opponent team.

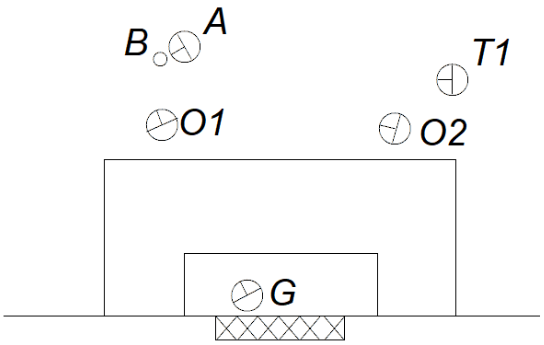

Fig. 1. Players arrangement in action utility assessment.

The basic set of considered player actions with their a priory utility assessments is presented in Table 1.

Table 1. Basic set of actions of player with ball.

Player action	A priory utility
Act_1 = Shot for the goal	90%
Act_2 = Pass	60%
Act_3 = Dribble	50%

The conditions for including players in the set of actual actors for teammates and opponents are presented in Table 2.

Table 2. Conditions for selection of actual actors.

Player action	Partners $\{T_m\}$	Opponents $\{O_k\}$
Shot for the goal	-	Opponents who can stop the shoot for the goal
Pass	Partners who can take the pass	Opponents who can stop the pass
Dribble	Partners who can lead defenders away	Opponents who can block the player, take away the ball

Possible actions of players included in the set of actual actors are presented in Table 3.

Table 3. Set of actual actors actions.

Player action	Partner action	Opponent action
Shot for the goal	-	$OAct_1$ = Try to take away the ball
		$OAct_2$ = Stay on the defensive
Pass	$TAct_1$ = open to pass	$OAct_1$ = Try to take away the ball
		$OAct_2$ = Stay on the defensive
Dribble	$TAct_1$ = lead defender away	$OAct_1$ = Block player
		$OAct_2$ = Try to take away the ball

The transition from the current world state to the set of possible states after agents perform certain actions can be represented in the form of bipartite graph (see Fig. 2) [10].

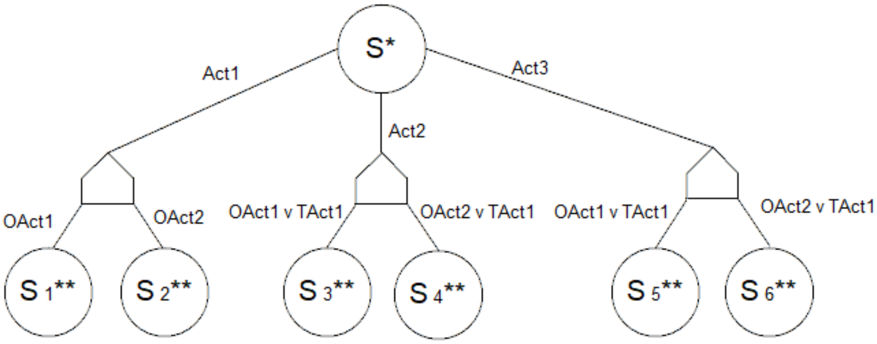

Fig. 2. Tree of predicted situations (of possible world states).

The root node of the graph corresponds to the current world state S*. Outgoing edges show the possible actions $\{Act_i\}$ of player who owns the ball. The second stage of the tree contains splitter-nodes, from which originate edges that denote combinations of actual actors possible actions when player performed action Act_i corresponding to the edge entering the node.

Action utility is calculated on the basis of the utilities of possible situations according to the minimax principle.

Utility of situations achieved as a result of certain action performed by players is evaluated on the basis of a set of many factors. The set of these factors depends on the specific situation and can vary widely. This allows us to construct utility assessment algorithms as ATAs. For example, in the simplest case the situation utility can be assessment based on the distance between the ball and the opponent's goal. In more complex cases the following factors can be taken into account: the numerical advantage of the players of our team in a certain area of the field, their relative position, individual capabilities of the players and other factors. In this way, the computational complexity of the algorithm for actions utility evaluating depends on the number of relevant actors, as well as on the number of actions that they can take.

Since the actions of different agents can be performed in various combinations, the number of possible situations (leaves in a tree) reached from one splitter-node in the worst case is m^n, where m is the maximum number of actions of one actor, and n is the number of actual actors.

The computational complexity of algorithm for determining the most useful action in the current situation depends on two factors:
- Number of actions for the agent possible in current situation;
- Complexity of utility value calculating for one action.

In the worst case, it can be estimated as $O(m^n * k)$, where k is the number of possible actions of the agent.

Taking into account such high computational complexity and hard real-time limitations for action selection, the utility calculation algorithm is constructed as ATA. The construction of a possible worlds tree is performed in descending order of the a priory (and then the current) utility of actions within the available time.

4 Approach experimental assessment

The operating time of action utility assessment algorithm is determined by the following formula:

$$T = t_{aAA} * k_A + t_{aU} * k_s \tag{1}$$

where t_{aAA} - upper estimate of the time to determine whether the agent in the considered situation is an actual actor; k_A - number of potential actual actors; t_{aU} - time for predicting and assessing of one situation; k_S - number of predicted situations.

The values t_{aAA} and t_{aU} should be calculated for certain computing platform by experimental measurement of the execution time of the corresponding program fragments.

The value k_A depends on the specific considered situation and is known before the algorithm starts. The value k_S is calculated after determining of actual actors set.

In this way, estimated time to action utility assessment can be calculated at the beginning of the algorithm work. This will allow agents to rationally distribute time.

An experimental study for the situation discussed above was carried out on a computer with an AMD A4-9120 processor, clock frequency 2.20 GHz. The results of measuring the times t_{aU} and t_{aAA} of the computation of the key fragments of the algorithm are presented in Table 4.

Table 4. The execution time of the key fragments of the algorithm.

Fragment	Execution time, ms
Predicting and assessing of situation, t_{aU}	0,22
Actual actor determining, t_{aAA}	0,06

Comparative estimates of theoretical and experimental time for calculating utility assessment for a different number of actual actors are presented in Table 5.

Table 5. Comparison of theoretical and experimental estimates.

Number of actual actors	T_{exp} ,ms	T_{theor} ,ms
0	0,00	0,00
1	0,25	0,28
2	0,50	0,56
3	1,06	1,06
4	1,16	1,13
5	2,06	2,06
6	4,09	3,88
7	3,91	3,94
8	6,91	7,50
9	12,71	14,56
10	13,44	14,63
11	26,44	28,69
12	52,72	56,75
13	60,69	56,81
14	104,75	112,87
15	215,25	224,94
16	210,71	225,00
17	420,97	449,06
18	810,28	897,13
19	821,19	897,72
20	1675,16	1793,25
21	3466,31	3585,31

The results of action utility assessment (in percent) for the considered example (see Fig. 2) are presented in Table 6.

Table 6. Actions utility assessment.

Player action	Predicted situations	Situation utility	Action utility
Shoot for the goal	S_1^{**}	65%	65%
	S_2^{**}	75%	
Pass	S_3^{**}	56%	56%
	S_4^{**}	63%	
Dribble	S_5^{**}	46%	46%
	S_6^{**}	53%	

5 Conclusions

Utility assessment of possible actions is the key task when an intelligent agent is choosing next its actions. The algorithm for action utility assessment in multi-agent worlds has high computational complexity which increases rapidly depending on the number of actors and their possible actions in the considered situation.

In the course of the research, the model for action utility assessment proposed in the model of advanced iterative action planning for real time intelligent agents was clarified. The work of the proposed model was analyzed on the example of a typical situation from the virtual soccer. Based on this analysis, conclusions were drawn about the complexity of the used algorithms. These results allow us to develop a utility assessment algorithm as an ATA so that intelligent agent can effectively use all available time in the current situation.

The obtained experimental results are focused on the environment of virtual soccer. However, the approach can be extended to other applications of agent systems, which constitutes the direction of further research.

References

1. Panteleyev M. G., Puzankov D. V. Intelligent agents and multi-agent systems: a monograph.: Publishing house SPbGETU ”LETI”, 2015. - 215 p. (in Russian)
2. Tyrrell T. Computational Mechanisms for Action Selection// PhD Thesis, University of Edinburg, 1993, 212 P.
3. Modelling Natural Action Selection// Seth A.K., Prescott T.J., Bryson J.J. (eds.), Cambridge University Press, 2012, 560 P.
4. Official RoboCup site [Electronic resource]: access mode- http://www.robocup.org.
5. Mellmann H., Schlotter B. Advances on Simulation Based Selection of Actions for a Humanoid Soccer-Robot// In Proc. of the 12th Workshop on Humanoid Soccer Robots, 17th IEEE-RAS Int. Conf. on Humanoid Robots, Madrid, Spain.
6. Ros R., Arcos J. L., de Mantaras R.L., Veloso M. A case-based approach for coordinated action selection in robot soccer// Artificial Intelligence, 2009, V. 173, N 9–10, pp. 1014-1039.
7. Öztürk P. Levels and Types of Action Selection: The Action Selection Soup// Adaptive Behavior, 2009, Vol 17(6), pp. 537–554.
8. de Sevin, E. Thalmann, D. A motivational Model of Action Selection for Virtual Humans// Computer Graphics International (CGI), IEEE Computer Society Press, New York 2005.
9. Panteleyev M.G. Advanced Iterative Action Planning for Intelligent Real-Time Agents// Procedia Computer Science, 2019, Vol. 150, pp. 244-252
10. Panteleyev M.G. The Formal Model of Advanced Iterative Real-Time Action Planning for Intelligent Agents// Proc. of the 14th National Conf. on AI KII-2014. Vol. 1. – Kazan: RIC ”School”, 2014, pp. 323-333.
11. Zilberstein, S. Using Anytime Algorithms in Intelligent Systems// AI Magazine, 1996, 17(3), pp. 73-83.