MULTICRITERIAL DECISION-MAKING IN ROBOT SOCCER STRATEGIES

Petr Tučník
University of Hradec Králové
Náměstí svobody 331, 500 01, Hradec Králové
petr.tucnik@uhk.cz

Jan Kožaný and Vilém Srovnal
VŠB - Technical University of Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba
{jan.kozany,vilem.srovnal}@vsb.cz

Keywords: Multicriterial decision-making, robot soccer strategy.

Abstract: The principle of multicriterial decision-making is used for the purpose of autonomous control of both individual agent and the multiagent team as a whole. This approach to the realization of control mechanism is non-standard and experimental and the robot soccer game was chosen as a testing ground for this control method. It provides an area for further study and research and some of the details of its design will be presented in this paper.

1 INTRODUCTION

A great deal of scientific effort is aimed at the multiagent systems today. Also, the area of multicriterial decision-making (or decision-support) is well developed. Nevertheless, the combination of both principles mentioned above still stands aside of the scientific interest focus and provides a space for further study.

The need to follow the restrictions as well as semantic meanings of general definitions provided by both principles is the reason why it is necessary to re-define some notions from the point of view of multicriterial decision-making (MDM) in multiagent systems (MAS). Also, the principle changes of both little or big importance have to be made accordingly. It is important to remember the fact, that there are restrictions and constraints, i.e. above all, the need of numerical representation of all the facts relevant for the decision-making (DM) process.

The MDM in MAS shows many features which make such approach at least very interesting. It is a method that is capable of providing an autonomous control to the MAS as a whole or to the individual agent. We have focused our research of this DM system on the problem of the robot soccer game. The game is fast, quickly evolving and represents a dynamically changing environment, where it is impossible to follow long-term plans. We have to make decisions based on the limited set of attributes (most of them have to be computed from the processed images), while there is a theoretically infinite amount of solutions and a minimal number of possible actuator actions.

2 THE THEORETICAL PART

We introduced the basics of MDM terms in (Tučník et al., 2006). In (Ramík, 1999), (Fiala et al., 1997), the wide scale of methods of multi-criteria decision-making may befound. The Fig. 1 shows the steps of the MDM process.

After initialization and goal acquisition in the phase 0, the agent tries to refresh its environmental data by its sensors – this is the phase 1. The agent's environment is described by attributies. Each attribute is expressed numerically and represents measure of presence of given characteristic in the environment. The attribute value fits into bordered interval the endpoints (*upper limit* and *lower limit*) of which

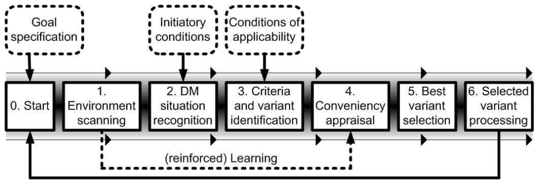

Figure 1: The MDM process.

are specified by the sensor sensitivity and range. During the signal processing, the standard normalization formula

$$\text{normalized\_value} = \frac{\text{attribute\_value} - k}{l - k} \quad (1)$$

is used. The attribute value is normalized to the interval of $\langle 0; 1 \rangle$ and variables k, l are expressing the lower and upper limit of the sensor scanning range.

The set of initiatory conditions is used for the decision-making situation recognition in the phase 2. It has to provide the agent with the ability to react to unexpected changes in the environment during its target pursue. Also, it must allow the agent to change its goal through the decision-making process pertinently, e.g. if the prior goal is accomplished or unattainable. In the phase 3, the conditions of applicability characterize the boundary that expresses the relevance of using the variant in the decision-making process. The most important step is the phase 4, where the convenience appraisal of applicable variants is performed. This step plays the key part in the decision-making procedure. The following formula is used:

$$\text{conv} = \sum_i w(\text{inv}(\text{norm}(a_i^v))), i = 1, \ldots, z, \quad (2)$$

where v stands for the total number of variants and z is total number of attribute values that are needed for computation of convenience value.

The convenience value for the each of assorted variants is obtained. Attributes a_i^v stand for presumptive values of universum-configuration and the *norm* function normalizes the value of the attribute and is mentioned above (1). The function *inv* is important, as it represents reversed value of difference between real attribute value and ideal attribute value:

$$\text{inv}(\text{current\_val}) = 1 - |\text{ideal\_val} - \text{current\_val}|. \quad (3)$$

The optimal variant remains constantly defined by m-nary vector of attributes, where $m \leq z$, for the each of the decision-making situations and attributes a_i^v differ for each variant other than the optimal variant. There is a final number of activities that the agent is able to perform. As the inverse values of difference between the real and ideal variant are used, in the most ideal case, the convenience value will be equal to 1, and in the worst case it will be close to 0. $\text{Im}(\text{inv}) = \langle 0; 1 \rangle$. The lower open boundary of the interval is useful, because troubles related to computation with zero (dividing operations) may be avoided.

The function w assigns the importance value (weight) to the each attribute. The machine learning is realized by proper modifications of the weight function. Importance of attributes differs in accordance with the actual state of the agent. E.g. energetically economical solution would be preferred when the battery is low, fast solution is preferred when there is a little time left, etc. Precise definitions of weight functions are presented in (Ramík, 1999), (Fiala et al., 1997). In the phase 5, the variant with the highest convenience value is selected and its realization is carried out in the phase 6. During processing of the selected solution, the agent is scanning the environment and if the decision-making situation is recognized, the whole sequence is repeated. The evaluation function (e.g. reinforced learning functions examples in (Kubík, 2000), (Pfeifer and Scheier, 1999), (Weiss, 1999)) provides the feedback and supports the best variant selection, as it helps the agent in its goal pursue. Based on the scanned environmental data, modifications of the function w are made during the learning process.

3 THE PRACTICAL PART

As it was said above the robot soccer game is a strongly dynamical environment (Kim et al., 2004). Any attempt to follow a long-term plan will very probably result in a failure. This is a reason why there are many solutions of this problem founded on the agent-reactive basis. But such reactive approach lacks potential to develop or follow any strategy apart from the one implemented in its reactive behavior.

On the other hand, the MDM principle, in general, provides a large variety of solutions and the strategy may be formed by the modifications of the weight coefficients value. Such modifications change behavior of the team as a whole and/or its members individually.

Simple reactions do not provide sufficient effort potential, or, in other words, the efficient goal pursue. There is a need of a quick, yet more complicated, actions, that would allow us to build up a strategy. Also, there is an important fact that the efficient strategy must not omit the agent cooperation.

3.1 Centralized vs. Autonomous Control

As there are two possible points of view – individual (agent) and global (team) - there are also two corresponding goals for the each agent. The global goals represent target state of the game for the team as a whole and pursue of global goal is superior to the individual goal of each agent. Therefore, the aim of individual goal has to correspond with the global goal. However, the emphasis on the autonomous behavior allows the agent to temporarily damage the team's effort of the global goal pursue in order to improve its position significantly in the next iteration(s). The autonomous aspect of the multiagent approach is therefore used to avoid local maximum lock-down. These ideas underline our approach to the solution of this problem.

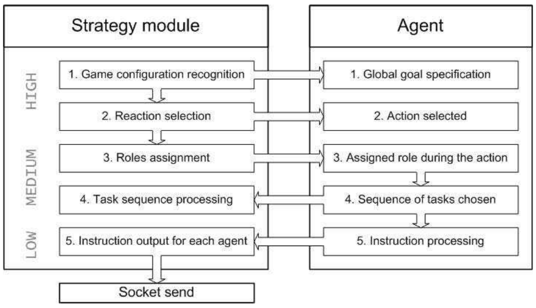

Figure 2: The scheme of communication between the strategy module and agents (the agent on the right stands for any agent of our team). At the beginning the game configuration is recognized and goal specified for each agent. In the second phase is chosen the most appropriate reaction in the form of action the agent has to perform. It is also assigned a role in the scope of this action. Then the agent itself forms a sequence of small tasks, which leads it towards advance in its goal pursue. These tasks are also processed in the strategy module, because of possible conflict check. As the last step, the set of simple instructions (i.e. practically HW instructions) is generated accordingly. These instructions represent the final output of the strategy module.

In order to gain maximum profit from both global (centralized) and individual (autonomous) approaches, the incremental degradation of centralization towards the autonomy was chosen during the strategy selection process. This means assignment of more individual instructions at the end of the strategy forming process and, on the contrary, the global team role assignment at its beginning. Fig. 2 shows the sequence of the decision output assigned to the agent.

There are three levels of the strategy module corresponding with three levels of decisions made. The high level strategy is responsible for the game configuration assessment and selection of the most appropriate actions. It is also a level, where the most centralized evaluations and global decisions are made.

The second layer of the medium level strategy is where the role and task assignment operations are performed. The notion of a "task" refers to the quite simple sequence of orders, for example: *GO TO POSITION X, Y, ANGLE 60°*.

Such simple tasks are transformed into instructions in the third (and final) low level strategy layer. These instructions are hardware-level commands and are used to set the velocity of the left and right wheel of the robot, separately. Because these instructions represents formatted output of the strategy module, they require no more of our further attention, as it is no aim of this paper to discuss such issues.

3.2 High Level Strategy

The purpose of this part of the strategy module is to provide a game configuration analysis and to perform a multicriterial selection of an appropriate reaction to the given state (of the game). There is an infinite number of configurations. Therefore, limiting, simplifying supportive elements have been implemented.

We will mention just one of these simplifying tools. The first and most important supportive element of the DM process is a ball-possession attribute. As the ball is the most important item in the game, it also plays a crucial role in the strategy selection process. Whoever controls the ball controls the game. Accordingly to the ball-possession, the set of all possible strategies is divided into three subsets. These subsets are:

- offensive strategies (our team have the ball),
- defensive strategies (opponent has the ball) and
- conflict strategies (no one has the ball or the possession is controversial (both teams have players very close tothe ball)).

Each item of these subsets has attached a certain number of actions to it. These actions represent different ways of superior (global) goal pursue.

In the MDM system, the convenience value must be obtained (from the formula (2)) for each of the possible solutions. This procedure is called multicriterial decision making process (MDMP) in the following text.

Because we need enough information to make a qualified decision (see (Fiala et al., 1997), (Ramík, 1999)), the sufficient amount of attributes has to be present. But there is a limited amount of data from the video recognition module. This is the reason, why it is necessary to perform additional attribute computing and gain more information needed.

The notion of the game configuration was used before. It refers, in the first place, to the position and angle of orientation of the each agent (robot) and to the position of a ball in the game. These are basics we can use to form a prediction of state of the game in next iterations. The prediction is based on the game configuration data (GCD) of the actual iteration and a few iterations back. However, it is vain to try to predict too far into the future, as the game is developing and the environment changing quickly.

Such approach to information extraction ensures enough data for GCD processing. The ACTION selected afterwards represents reaction of the multiagent system to the present situation. Such ”ACTION” is the basic concept for the behavior scheme and few steps in the medium layer of the strategy module have to follow until the final output (command) is generated.

3.3 Medium Level Strategy

The function of this layer of the strategy module is to assign the role to the agent and to ensure, that the possible collisions of plans between team-mates will be avoided. The high level of the strategy module has selected an appropriate action to the given state of the game. Further steps during the DM process have to be taken.

One of the most important parts of the ACTION structure is the ROLE. The ROLE stands for function the agent has to guarantee during the subsequent performance of chosen activity. The ROLE may be either leader or support. For the leader role, the most convenient agent is assigned. The convenience value is calculated using the formula (2), and this calculation is preferentially based on its position and angle of orientation and additional supportive attributes, such as prediction of positions and movement trajectories. The support roles are assigned in a similar manner.

The most important are actions of the leading agent with the ball. All other activities are inferior to it and all agents required have to cooperate. However, in the opposite case, there may be a spare potential of non-employed agents and its utilization should be arranged.

As the back-up solution, there is also present a standard behavior algorithm. Such behavior is applied when there are no other actions possible or suitable for given situation, or not enough agents available for execution.

As it is shown in the Fig. 2, the selection of the sequence of tasks is the next step. Every action may be divided into atomic tasks. This decomposition forms desired sequence and check operations are performed to avoid collisions and other errors or mistakes. The possible necessary adaptations are made in a centralized manner and all agents proceed to the final low-level layer of the strategy module.

4 CONCLUSION AND FUTURE WORK

The implementation of the MDM-based control mechanism is an experimental matter. The robot soccer game provides the testing and proving ground for this approach. Further tests have to be performed as well as adaptation and optimization work. However, the principle shows a promising potential and it is able to function as a control mechanism and decision-making tool.

ACKNOWLEDGEMENTS

The work and the contribution were supported by the project from Grant Agency of Czech Academy of Science - Strategic control of the systems with multiagents, No. 1ET101940418 (2004-2008) and by the project from the Czech Republic Foundation - AMIMADES, No. 402/06/1325.

REFERENCES

Fiala, P., Jablonský, J., and Manas, J. (1997). *Vícekriteriální rozhodování*. VŠE Praha.

Kim, J., Kim, D., Kim, Y., and Seow, K. (2004). *Soccer Robotics (Springer Tracts in Advanced Robotics)*. Springer-Verlag.

Kubík, A. (2000). *Agenty a multiagentové systémy*. Silesian University, Opava.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. Cambridge, Massechusetts.

Ramík, J. (1999). *Vícekriteriální rozhodování – Analytický hierarchický proces (AHP)*. Silesian University.

Tučník, P., Kožaný, J., and Srovnal, V. (2006). Multicriterial decision-making in multiagent systems. *Lecture Notes in Computer Science*, pages 711 – 718.

Weiss, G. (1999). *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massechusetts.