Heuristic Approach to Conflict Problem Solving
in an Intelligent Multiagent System

Witold Jacak and Karin Pröll

Dept. of Software Engineering at Hagenberg
Upper Austrian University of Applied Sciences
Softwarepark 11, A 4232 Hagenberg, Austria
jacak@fh-hagenberg.at proell@fh-hagenberg.at

Abstract. The paper presents a method that allows an intelligent multiagent system to coordinate and negotiate their actions in order to achieve a common goal. Each individual agent consists of several autonomous components that allow the agent to perceive and react to its environment, to plan and execute an action, and to negotiate with other agents in an intelligent manner. A heuristic approach for conflict solution is presented, which is used for coordination of a society of independently acting agents in common environment. The application of the method is shown on an intelligent multiagent robotic system realizing complex transfer operations simultaneously.
Key words: multi agent systems, intelligent robotic agent, heuristic conflict problem solution

1 Introduction

Intelligent agents are a new paradigm for developing complex system applications. The most powerful tools for handling complexity in system development are modularity and abstraction. Industrial application of agent technology were among the first to be developed, in such domain as process control, manufacturing and robotic systems. The agent systems base on autonomous software-hardware components (technical agents) that cooperate within an environment to perform some task. This paper presents a heuristic approach for conflict management in an intelligent multiagent system, which is used to coordination of a society of independently acting agents in common environment [1, 2, 6, 7, 9, 10, 12, 13].

1.1 Multiagent system

A multiagent system is a group of cooperating agents $A_i$ acting in common environment. Cooperation between agents has a different focus. The main task is to solve complex problems by using cooperation and communication mechanism with other agents or external resources. For the communication among agents the communications platform have to be established and communication protocols have to be defined in order to ensure that agents can exchange information. The
behavior of the overall multiagent system can be observed as the set of actions performed by each individual agent. The special contract agent determines the goal for each agent. Each agent $A_i$ follows its own specific goal $g_i$. The set of all goals $G$ is quasi ordered by a priority relation (reflexive and transitive), and can change in the time. Each agent has its own goal $g_i$ and an autonomous behavior, which is a result of its observations, its knowledge and its interactions with other agents [1-4]. A multiagent system is presented in Figure 1.

1.2 An intelligent agent

Each intelligent agent consists of several cooperating units such as: $KB$ - knowledge base, $Neg$ - negotiator, $Int$ - relay ports (the interface) the set of agent input and output and $E$ - state engine that describes the behavior of an agent. The $i$-th technical agent can be represent by a tuple

$$A_i = (KB_i, Int_i, Neg_i, E_i)$$

(1)

The behavior of an agent is specified by state engine $E$. State engine can be represent by a tuple

$$E_i = (Ac_i, S_i, M_i, \tau_i, \delta_i, \lambda_i, \kappa_i)$$

(2)

where:
- $Ac_i$ is the action set of agent. The action represents the possible activities of the agent,
- $M_i$ is the set of incoming and outgoing messages,
$S_i$ is the set of states of the agent. The state can be passive or active. The active state has finite advance time defined by $\pi_i$ (time advance function),

- the trigger function $\delta_i : S_i \times Ac_i \rightarrow S_i$ determines the state transition,
- the response (output) function $\lambda_i : S_i \rightarrow M^{send}$ generates the possible output messages connected with the state,
- the conflict function $\kappa_i : M^{received}_i \times S_i \rightarrow \{T, F\}$ determines the feasibility (conflict freeness) of state based on incoming messages. When $\kappa_i(m^{received}_i, s_i) = T$ it means that state $s_i$ is in direct conflict with another agent state $[8, 9, 13]$.

2 Goal Reachability Problem

The goal $g_i$ of an agent determines the set of final states $S_i^{final} \subseteq S_i$ of $A_i$ in current moment of the time. The main goal of a multiagent system is to realize all of the goals from set $G$, which are distributed to the individual agents. The problem of goals realization for all active agents appears to be a problem of reachability of the set of final states from the start state for each agent. In order to solve such reachability problem we are going to apply the graph searching procedure to the state transition graph of state set $S_i$. The way of expanding the graph will depend on the form of the cost function using to evaluate each node. As the evaluation function we can use the sum of the cost function $c(s_i^{start}, s_i^c)$ and a cost estimate heuristic function $h(s_i^c, s_i^{final})$, the estimated cost between agent current state and terminal state. Using the standard $A^*$ procedure we can find the state path

$$s_i^* = (s_i^{start}, s_i(1), \ldots, s_i(K), \ldots, s_i^{final})$$

from current state to goal state which includes only feasible states.

The problem of optimal state path $s_i^*$ planning for i-th agent amounts to finding a sequence of actions

$$ac_i^* = (a_i(1), \ldots, a_i(K))$$

such that: the terminal state is achieved, every state $s_i(k)$ for $k = 1, \ldots, K$ is conflict free and the sequence $ac_i^*$ minimizes the cost function $c_i = c_i + h_i$.

**Fact 1** If for all agent states, no conflict occurs then the global trajectory $s^*$ is composed from the $s_i^*$ state trajectories of each active agent.

As the individual behavior of all agents involved in own goal achievement cannot be predicted in advance, the states of two or more agents can be in direct conflict with one another and the achievement of the goal is endangered. In order to resolve a conflict situation, a negotiation process among all conflict parties has to take place. The result of the negotiation should be a solution, which results in goal achievement for all involved agents $[2, 3]$.
2.1 Sequential synchronized reachability solution

The conflict between the goal states from the set $S_{i}^{\text{final}}$ and states of another agent causes that for example a solution for the reachability problem does not exist. To avoid this problem we propose the sequential decomposition of the global search problem into coordinated sequential search for each agent separately. It means that the not existing simultaneous solution will be substituted by a sequence of partial solutions distributed in time.

Fact 2 If for each agent there exists an optimal trajectory, which achieves the goal state separately conflictless in start state of each agents, and an additional trajectory, which states are not in conflict with each state from optimal trajectory of other agents then the global reachability problem can be decomposed into n sequential synchronized problems such that each agent realizes part of it's optimal trajectory, when all other agents move to the conflicts free states sequentially [13].

The global goal will be reached by sequence of realization of local goals for each agent (Figure 2).

3 Heuristic conflict problem solving

The state machine model of the agent behavior can be applied to solve the conflict avoidance problem by substitution of the concurrent performed planning and realization of actions of agents group by the sequentially performed one or more steps planning and realization of the action of several agent from agent group. Such round robin procedure for time synchronization of agent action leads to concurrent realization of actions with k-step delay and to solve the conflict avoidance by using the sequence time synchronized partial solution of goal achievement. Let $AGENT(t)$ be a list of agents ordered by values of priorities of their goals in time moment $t$. We will look for the best action of each agent in moment $t$, which does not cause the conflict and leads to final states set. For this purpose we shall exploit the state transition graph of the each agent generated implicitly by applying the transition function as production rules sequentially for each agent from list $AGENT(t)$.

3.1 Sequential coordination and planning algorithm

The first agent from the list generates the new part of state's path in following steps:

Step 1 Agent requests the message about current states of other agents in common environment, and establishes the own conflict states.

Step 2 Based on the conflict states and its own status, agent recognizes its own current situation and establishes the parameters for graph search algorithm, such as type of evaluation function and type of state feasibility testing function.
Fig. 2. Optimal trajectory, but conflict situations above and coordinated sequential search for conflict avoidance below

**Step 3** Action planner starts the state-graph searching algorithm from its current state with previously established evaluation and conflict freeness testing functions. The searching stops if the goal state is reached or if the OPEN set is empty or if the OPEN set has more than N elements.

**Step 4** The temporary path of states is calculated i.e. $\text{path}^{b,m} = s_i^c \rightarrow s_i^b$ where $s_i^b$ is the best state in CLOSE set. Depending on the length and conflict freeness of this path the new state chosen and actions sequence is realized.

**Step 5** The new current state is analyzed and the adequate message is send to other agents. The message includes the information if agent has achieved its goal and/or if the conflict occur. Negotiator Neg tries to perform change the priority level of agent goal, based on current distance to final state. This leads to change the order of AGENT list for the next step. The new list is
sent to each agent.

With sending the message agent ends his activity and the next agent starts the path planning. In each interval of time only one agent is working. The other agents are waiting for message Action done to come [11-13]. The overall negotiation process is shown in Figure 3.

![Diagram of communication platform and agent actions](image)

**Fig. 3.** Action planning and conflict solving cycle

4 **Application in Intelligent Robotic Agents System**

This section presents an intelligent multiagent robotic system as an application of the general multiagent system approach, which is used to control and coordinate a community of independently acting robotic agents in partially known environment.

Each robotic agents contains two main units: the software unit (intelligent controller) and the hardware unit (robot). The general goal of a multiagent system
is to solve all transfer tasks, obtained from other parts of the manufacturing system. Our considerations will focus on groups of service robots realizing their goals with different motions. Goal distribution and supervising of goal execution is done by the management level responsible for cooperation and negotiation between robotic agents on the operational level.

4.1 Intelligent robotic agent

Robot kinematics with n DOF can be described as FSM Model in Joint Space Q. State of the robotic agent is \( q = (q_1, \ldots, q_n) \) and the action can be modelled as \( \delta q = (\delta q_1, \ldots, \delta q_n) \). If the distance between two agents is less than a safety zone it will be denoted as conflict between these agents. The feasibility of the agent current state \( q(i) \) can be expressed as the geometrical distance between states \( q(j) \) of different agents \( \rho(q(i), q(j)) \). If \( \rho(q(i), q(j)) < \epsilon \) then \( \kappa_i(m_{received}^{from j}, q(i)) = T \) [1, 2, 5].

4.2 Example

Figure 4 presents a full conflict situation between Agent 1 and Agent 2 in various and final positions. Concurrent planning and realization of actions of agents group without negotiation leads to full conflict in final positions. The agents do not achieve their own final states.

![Image](image.png)

Fig. 4. In the left picture a full conflict between Agent 1 and Agent 2 in various positions, including the final state is shown. In the picture on the right concurrent planning and realization leads to not achieving final states.

Solving the conflict avoidance problem by substitution of the concurrent planning and realization of actions of robots group with the sequentially performed one step planning and realization of the action of several robot. The round robin procedure for time synchronization of agent action leads to concurrent realization of actions with \( k \)-step delay and to solve the conflict problem by using the sequence time synchronized partial solution of goal achievement. (Figure 5)
Fig. 5. Conflict situation resolved

References