Multi-Agent Programming Contest
Scenario Description
2009 Edition

Revised 18.06.2009

http://www.multiagentcontest.org/2009

Tristan Behrens Mehdi Dastani Jürgen Dix
Michael Köster Peter Novák

An unknown species of cattle was recently discovered in the unexplored flatlands of Lemuria. The cows have some nice features: their carbondioxyde- and methane-output is extremely low compared to the usual cattle and their beef and milk are of supreme quality and taste.

These facts definitely caught the attention of the beef- and milk-industries. The government decided to allow the cows to be captured and bred by everyone who is interested and has the capabilities. Several well-known companies decided to send in their personnel to the fields to catch as many of them as possible. This led to an unprecedented rush for cows. To maximise their success the companies replaced their traditional cowboys by *artificial herders*.

1 Scenario: Cows and Herders

In this year's agent contest the participants have to compete in an environment for cows. Each team controls a group of herders in order to direct the cows into their own corral. The team with the highest number of cows in the corral at the end wins the match.

1.1 General Description

Each team will compete against all other teams in a series of matches. A single match between two competing teams will consist of several simulations. A simulation between two teams is a competition between them with respect to a certain configuration of the environment.

Winning a simulation yields 3 points for the team, a draw is worth 1 point and a loss 0 points. The winner of the whole tournament is evaluated on the basis of the overall number of collected points in all the matches during the tournament. In the case of an equal number of points, the winner will be decided on the basis of the absolute number of captured cows.

Details on the number of simulations per match and the exact structure of the competition will depend on the number of participating teams and will be specified later.

In the contest, the agents from each participating team will be executed locally (on the participant's hardware) while the simulated environment, in which all agents from competing teams perform actions, is run on the remote contest simulation server run by the contest organizers.

The interaction/communication between agents from one team should be managed locally, but the interaction between individual agents and their environment (run on the simulation server) will be via Internet. Participating agents connect to the simulation server that provides the information about the environment.

Each agent from each team should connect to and communicate with the simulation server using one TCP connection. After the initial phase, during which agents from all competing teams connect to the simulation server, identify and authenticate themselves and get a general match information, the competition will start. The simulation server controls the competition by selecting the competing teams and managing the matches and simulations. In each simulation, the simulation server, in a cyclic fashion, provides sensory information about the environment to the participating agents and expects their reactions within a given time limit.

After a finite number of steps the simulation server stops the cycle and the participating agents receive a notification about the end of a simulation. Then the server starts a new simulation possibly involving the same teams.

1.2 Preparation Stage and Communication Protocol

Before the start of the competition, the contest organizers will contact participants via e-mail with details of time and internet coordinates (IP addresses/ports) of the simulation server. The participants will also receive agent IDs and passwords necessary for authentication of their agents for the tournament. Agents communicate with the simulation server using the TCP protocol and by means of messages in an XML format. Details about the communication protocol and message format will be specified later.

1.3 Initial Phase

At the announced starting time of the tournament, the simulation server will go on-line, so that the agents from the participating teams will be able to connect. After a successful initial handshake during which agents will identify themselves by their IDs and receiving acknowledgment from the server, they should wait

for the simulation start. The initial connecting phase will take a reasonable amount of time in order to allow agents to be initialised and connected and will not be less than 5 minutes. Details will be announced later.

1.4 Final Phase

In the final phase, the simulation server sends a message to each agent. By this, the tournament is over.

1.5 Network Miscellanea

Our simulation server does not provide a facility for inter-agent communication. Agents from a team are allowed to communicate and coordinate their actions locally. Based on the number of participants, organisers will decide whether to run the competition in just one or more rounds.

The continuous connection of agents from the first match to the last one cannot be guaranteed. In the case of agent-to-server connection disruption, agents are allowed to reconnect by connecting and performing the initial tournament phase message exchange again.

Generally, participants are responsible for maintaining connections of their agents to the simulation server. In the case of connection disruption during the running simulation, server will proceed with the tournament simulation, however the action of a disconnected agent will be considered as the *skip* action. In the case of a serious connection disruption, organizers reserve the right to consider each case separately.

The agents should inform the simulation server which action they want to perform within a timeout specified at the beginning of the simulation. The contest organisers do not take any responsibility for the speed of the Internet connection between the server and the participating agents. Timeouts will be set reasonable high, so that even participants with a slow network connection will be able to communicate with the server in an efficient way. Simulation timeouts will not be lower than 2 and higher than 10 seconds per one simulation step.

2 The Environment

The environment is a rectangular grid consisting of cells. The size of the grid is specified at the start of each simulation and is variable. However, it cannot be more than 150×150 cells. The $[0, 0]$ coordinate of the grid is in the top-left corner (north-west). The simulated environment contains two corrals—one for each team—which serve as a location where cows should be directed to. Furthermore there can be fences that can be opened using switches.

Each cell of the grid can be occupied by exactly one of the following objects:

- *Agents* are steered by the participants and can move from one cell to an adjacent cell.

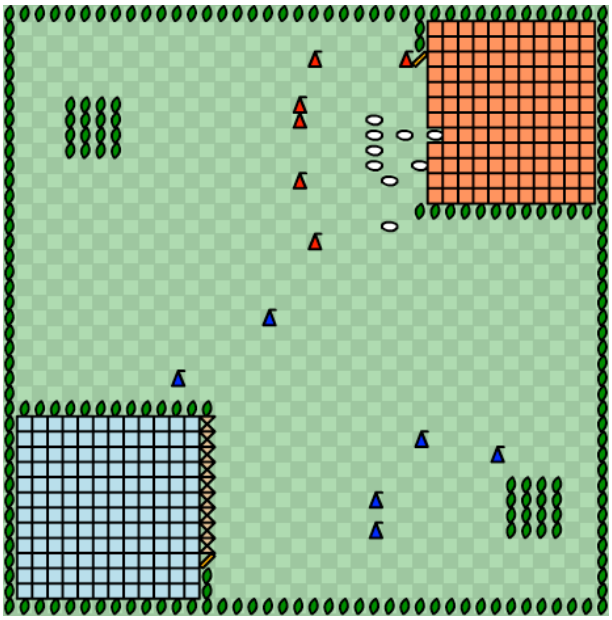

Figure 1: The environment is a grid-like world. Agents (red and blue triangles) are steered by the participants. Obstacles (green ovals) block cells. Cows (white ovals) are steered by a cow-algorithm. Fences (x-shapes) can be opened by letting an agent stand on a reachable cell adjacent to the button (slash-shaped). Cows have to be pushed into the corrals (red and blue rectangles.

- An *obstacle* blocks a cell.
- *Cows* are steered using a flocking algorithm. They can move from one cell to an adjacent cell. Cows tend to form herds on free areas, keeping the distance to obstacles. If and agent approaches, cows get frightened and flee.
- *Fences* can be opened using a button. To open a fence and keep it open an agent has to stand on a cell adjacent to the respective button. Thus, a switch is activated if the agent is next to the switch and:
  1. the current cell (where the agent is in) does not contain an open or closed fence, and
  2. the position of the agent is not diagonal to the switch, i.e., $abs(x0 - x1) + abs(y0 - y1) = 1$.

  Note:

  – An agent cannot open a fence and then definitely go through it. Instead it needs help from an ally.

– When fences closes, agents and cows that stand on a fence cell get pushed into a free cell.

Furthermore, there are two corrals, which are rectangular areas, one for each team. Cows have to be pushed into these corrals. Each teams learns the position and the dimensions of its corral at the beginning of each simulation.

2.1 Agent Perceptions and Actions

Each agent perceives the contents of the cells in a fixed vicinity around it. It can distinguish between empty cells, obstacles, cows (distinguished by unique identifiers), fences, buttons, and other agents. The participants will learn the position and dimensions of their team's corral at the beginning of each simulation. Each agent can move to one of the adjacent cells if the cell is not blocked by an obstacle, cow, fence, button or another agent. Each agent perceives a square of cells of size 17×17 with the agent in the center. Each team has 10 agents.

Each agent reacts to the received sensory information by indicating which action (including the *skip* action) it wants to perform in the environment. If no reaction is received from the agent within the given time limit, the simulation server assumes that the agent performs the *skip* action. Agents have only a *local view* of their environment, their *perceptions can be incomplete*, and their *actions may fail*. The simulation server can omit information about particular environment cells, however, the server never provides incorrect information. Also, agent's action can fail. In such a case the simulation server evaluates the agent's action in the simulation step as the skip action.

2.2 Cow Movement Algorithm

Although we deem the details of the cow movement algorithm not to be important, we will sketch it here. Note that the complete algorithm is available in the source-code of this year's scenario.

For each cow the algorithm considers all the cells that can be reached by it in one step. Then the weight of these cells is computed. The cow moves to that cell whose weight is maximal. If there are several maxima, the cow moves randomly to one of them.

Algorithm 1 Cow movement algorithm.

Require: a cow represented by its position vector $c \in \mathbb{N} \times \mathbb{N}$

1: let N be the set of the 9 cells adjacent to c, including c;
2: remove from N all those cells that are not reachable;
3: calculate the weights of all cells $n \in N$;
4: determine the set $M \subseteq N$, where the weight for each $m \in M$ is maximal;
5: randomly pick a cell $m \in M$;
6: move the cow to m;

Algorithm 2 Calculate the weight of a given cell.

Require: a cell represented by its position vector $n \in \mathbb{N}\times\mathbb{N}$, and a cow-visibility range $r \in \mathbb{N}$

1: determine the set C of all cells that are a in the rectangle $[n_x - r, n_y - r + n_x + r, n_y + r]$ and that are on the map;
2: set ret to 0;
3: **for all** $c \in C$ **do**
4: calculate d the distance between c and n;
5: get the weight w of c in respect to the cell content;
6: add w/d to ret;
7: **end for**
8: **return** ret

The weights for attractive cells – empty space, other cows, and corral cells – are positive. The weights for repellent cells – agents and obstacles (trees, gates) – are negative.

Finally, cows are slower than agents. They move every third step.

2.3 Comparison to Last Year's Contest

The 2009 edition of the Multi-Agent Contest is essentially the same except for three minor differences.

- Cows will not be removed from the environment if they enter the corrals. The number of cows in the corrals after the last step counts.
- The cow movement algorithm has been improved in order to yield a more convincing behavior of the cows.
- The new scenario also introduces fences.
- The team-size has been increased.

3 Submission and Winning criteria

A submission consists of a 5 page description (analysis and design) of your solution and the participation of your agent team in the tournament. The winner of the contest will be the best performing team with the highest number of points from the tournament.

4 Technical Support and Organisational Issues

We are running a mailing list for all the inquiries regarding Multi-Agent Programming Contest 2009. Feel free to subscribe if you are interested in further details on the contest. Subscription for participants is mandatory. The list's address: `agentcontest2009@in.tu-clausthal.de` To subscribe, please send an e-mail to `agentcontest2009-subscribe@in.tu-clausthal.de` The most recent information about the Multi-Agent Programming Contest 2009 can be found on the official web site `http://www.multiagentcontest.org/2009`