# Announcement Based Policy Shaping in Multiagent Systems

Arnis Stasko[1], Janis Grundspenkis[1]

[1] Riga Technical University, Riga, Latvia
{arnis.stasko,janis.grundspenkis}@rtu.lv

**Abstract.** European Organization for Nuclear Research (CERN) offered challenging Grid Wars competition during CERN Spring Campus in Riga. To manage complexity, the authors treated the game as a multiagent system where a team of agents tries to grow their army and fight against the opponent. The group of agents collaborated to follow the optimal collective strategy and maximize cooperative goal. Due to the growing size of the team after each step, possible cooperative action combinations dramatically increased. As a solution, this paper introduces an announcement based policy shaping method for decentralized multiagent system coordination. The method is validated by demonstrating experimental results and by winning the Grid Wars competition.

**Keywords:** decentralized multiagent system, coordination, planning, policy.

## 1 Introduction

Agents are independent computer systems that work to achieve goals located in a particular environment. Essential agents' capabilities are autonomy to decide for actions to satisfy their design objectives and capability of interacting with other agents (Wooldridge, 2009). In multiagent systems, multiple intelligent agents interact between themselves and the environment. They can cooperate to achieve a collective goal or compete to attain individual ones.

In multiagent systems where unified coordination is impossible due to too many agents and limited time for communication, another solution should be found. Looking in nature how a swarm of birds collectively make an organized flight we see a great example of coordination. Inspired by nature, authors introduce announcement based policy shaping method for decentralized multiagent system coordination.

The paper is structured as follows. Next chapter presents the European Organization for Nuclear Research (CERN) organized Grid Wars competition rules and problem domain. Related work is discussed in Chapter III. Announcement based policy shaping method for multiagent systems is introduced in Chapter IV. Method validation is given in Chapter V. Finally, conclusions and future work are discussed in Chapter VI.

## 2 Problem Domain

Hence having learned the best policy for a single agent in a given environment, it's not guaranteed for an agent to be optimal while working in a team of multiple agents. For each step, the action shall be selected with respect to expected actions of agent's teammates. Given that there is a limited time for making decisions and the fact that an agent would choose different action knowing planned actions of other agents in the team the problem becomes sorely challenging. To discuss the problem domain and illustrate results authors will use a grid world example derived from Grid Wars competition offered by European Organization for Nuclear Research (CERN) to CERN Spring Campus participants.

The grid world will consist of 50x50 cell board. There are two teams or coalitions which fight for the victory. Initially, there is a 100-unit army in a random cell for each of the teams. In this paper, each cell with an army on it is considered as an agent. The agent can decide how many units to keep in the cell and how many of them to send to the left, to the right, up or down. As the board is a torus the edges wrap around, that means, e.g. by taking left move on the very left cell the selected units of the army will be moved to the very right cell on the same line of the board. During a single move, every agent (a cell with an army on it) has to make a decision and take action.

After each move, there will happen population grows by 10% per cell rounded mathematically. So having 0-4 units in the cell will not get an increase. Having 5 – 10 units the cell will get increase by 1. Every cell can handle at maximum 100 units. All units over 100 per cell will be lost. An illustrative example of the first move population's is given in the Fig. 1.

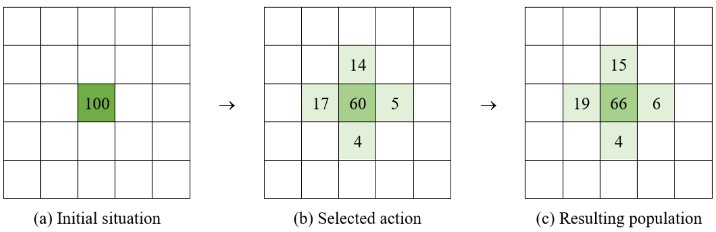


**Fig. 1.** Example of the first move

In the original game, when the armies of two teams meet in a single cell, they lose an equal amount of units. For example, team A sends 10 units to a cell while team B sends 15 units to the same cell. As a result, team A loses all 10 units and also team B loses 10 units, so 5 units of team B stays in the cell. The game is won by the surviving team or by the highest population after 2000 rounds. For the purposes of this paper, authors will analyze only single team without an opponent and set the target to reach maximum population as soon as possible (during a minimum count of moves).

In the grid world example mentioned above, the problem is to find an optimal policy. The policy is a mapping from every possible state to the best move in that state. Optimal one is that yields the highest expected utility for an agent (Russell & Norvig, 2010). Having found an optimal policy, the agent can construct a plan in which sequence to take which actions. If an agent can't detect optimal policy, it is difficult to plan future actions.

## 3 Related Work

Addressed problem corresponds to cooperative decision making or cooperative action. Sandholm et al. identify it as solving the optimization problem of a coalition (Sandholm et al. 1999) where the coalition's objective is to maximize joint monetary value. In our case, we don't have to solve an issue about which agents are in which coalitions. Coalitions are known.

To make coalition member actions maximize joint utility distributed planning problem "where each agent must formulate plans for what it will do that take into account (sufficiently well) the plans of other agents" can be identified (Durfee, 1999). Durfee classifies three main categories of multiagent planning:

1. Centralized planning for distributed plans – unlike the work mentioned above in our case a master agent creates a plan and distributes it to other agents which then executes it;
2. Distributed planning – a group of specialized agents cooperatively form a centralized plan for other agents to execute;
3. Distributed planning for distributed plans – agents dynamically cooperate to form individual plans and execute plans by themselves.

In our case due to changing environment and limited time for action selection for each step third category "distributed planning for distributed plans" is applicable where each agent forms its plan by taking into account expected activities of other agents. Moreover, due to unknown next actions of the opposite team when it's impossible to exactly know the next state after the move the multiagent plan can be drawn up only for a single step. Thus to the best of our knowledge, the exact solution is not proposed in the literature.

## 4 Proposed Solution

Firstly, in this section, an individual strategy for one agent is explained. Secondly, announcement based policy shaping method for the multiagent system is proposed.

### 4.1 Individual strategy

Getting to know the grid world rules the reader can notice that it is optimal to have agents/cells with the unit amount which is divided into five without remainder. That's because 1, 2, 3 and 4 units don't produce any units in the next step while 5 units produce the same amount as 6, 7, 8, 9 or 10. Furthermore, having three agents with 5 units per agent is far better than having 15 units in one cell. To continue the agents must be aware of concentrating too many units in a single spot due to the risk of over-population and thus lost units per cell.

Taking into account mentioned above let's build a simple strategy for an agent to select a roughly optimal action:

Step 1. Reserve 5 units as a minimum to be kept in the cell [transfer] = [units] – 5

Step 2. Detect empty neighbour cells [empty cells]

Step 3. IF there are empty neighbour cells [empty cells] > 0:

Step 3.1. Calculate the transfer portion of units [portion] = [transfer] / [empty cells]

Step 3.2. Reduce the transfer portion to amount which divides into five [opt portion] = (7 → 5; 10 → 5; 17 → 15 and so on)

Step 3.3. Transfer the reduced portion [opt portion] to each empty neighbour cell

Step 3.4. Transfer extra units to the one of previously empty neighbour cells [extra portion] = [transfer] – [opt portion] * [empty cells]

Step 4. IF all neighbour cells have units [empty cells] == 0:

Step 4.1. Split the army equally to all directions [opt portion] = [transfer] / 4

Step 4.2. Transfer [opt portion] to each neighbour cell

Step 4.3. Leave surplus in the cell

According to the described policy suppose that we have a situation depicted in Fig. 2a. Every agent tries to keep 5 units in the cell (bolded number aligned left) and move the rest units according to the policy to the neighbour cells (regular number aligned right) as shown in Fig. 2b. During this move, the population has grown from 65 units to 74 units (Fig. 2c).

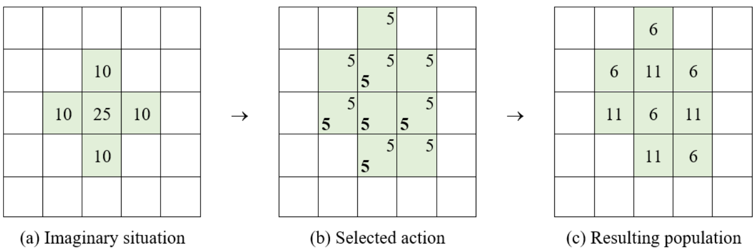


**Fig. 2.** Individual action selection regarding individual strategy

The individual strategy proposed above checks nearby neighbour cells to select direction where to send superfluous army units. This rule could be improved in case if all neighbour cells are occupied to check further distance neighbour cells iteratively:

Step 1. Reserve 5 units as a minimum to be kept in the cell [transfer] = [units] – 5

Step 2. Start with [neighbour radius] = 0

Step 3. Detect empty neighbour cell (within [neighbour radius]) count [empty cells]

Step 4. IF there are empty neighbour cells (within [neighbour radius]) [empty cells] > 0:

Step 4.1. Calculate the transfer portion of units [portion] = [transfer] / [empty cells]

Step 4.2. Reduce the transfer portion to amount which divides into five [opt portion] = (7 → 5; 10 → 5; 17 → 15 and so on)

Step 4.3. Transfer the reduced portion [opt portion] to each empty neighbour cell

Step 4.4. Transfer extra units to the one of previously empty neighbour cells [extra portion] = [transfer] – [opt portion] * [empty cells]

Step 5. IF all neighbour cells (within [neighbour radius]) have units [empty cells] == 0:

Step 5.1. If maximum neighbour radius is not reached:

Step 5.1.1. Increase [neighbour radius] by one

Step 5.1.2. Continue with Step 3.

Step 5.2. If maximum neighbour radius is reached:

Step 5.2.1. Split the army equally to all directions [opt portion] = [transfer] / 4

Step 5.2.1. Transfer [opt portion] to each neighbour cell

Step 5.2.2. Leave surplus in the cell

On the 50x50 cell board, the agent count can reach 2500. Every agents' decision for the next action can impact the utility of selected actions by other agents. Further, in this article, the grid will be treated as a multiagent system where each cell is an agent and each agent tries to select the best possible action by taking into account the possible actions of other agents.

### 4.2 Announcement based policy shaping

In Fig. 1 there are 4 agents/cells which tries to keep 5 units and at the same time receives 5 units from the neighbour. As the reader can notice if those 4 agents had known the arrival of additional 5 units they could send an extra 5 units to empty neighbours as shown in Fig. 3. From the same initial population 65, the population could have grown to 78 units (+4 units than before).

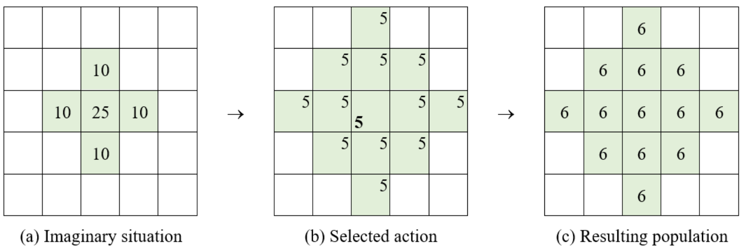


**Fig. 3.** Optimal collective strategy

To achieve in Fig. 3 demonstrated optimal collective strategy the policy for each agent shall be shaped by taking into account expected actions by other agents. In a centralized solution, such communication and computation are possible in small teams. But having teams rather big it becomes expensive. Possible change in the decision of one agent requires recalculation of policy for all other agents. For example, if an agent is having ten units of the army, it can send them left, right, up, down and keep (5 options). For each of five options, they can choose 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 or 10 units (11 choices). Mathematically an agent has (5 – 1) ´ 11 = 44 different actions. If there are ten such agents, they all together make up 4410 = 27 197 360 938 418 176 different cooperative action combinations. Illustrative cooperative action combinations are given in Table 1.

**Table 1.** Cooperative action combinations.

| Agent count | Units per agent | Actions per agent | Cooperative action combinations |
|---|---|---|---|
| 1 | 1 | 5 | 5 |
| 2 | 1 | 5 | 25 |
| … | … | … | … |
| 10 | 1 | 5 | 9 765 625 |
| 1 | 10 | 44 | 44 |
| 2 | 10 | 44 | 1936 |
| … | … | … | … |
| 10 | 10 | 44 | 27 197 360 938 418 176 |

Assuming that the ideal solution for each step cannot be calculated due to the growing cooperative action combination count after each step, a good enough compromise solution was searched. What if we introduce announcement based communication between agents? Every agent could calculate their optimal policy for a given step and announce planned actions to their neighbours. After an announcement, every agent could improve their policy and select a more suitable action. Announcement based policy shaping method for multiagent systems is introduced. The offered solution gives collaboration model where agents are better informed and can improve their plans at low computation costs. Step by step algorithm is given below.

Step 1. Every agent calculates individual policy.

Step 2. Every agent announces to their neighbours how much army units it is planning to send during the next move.

Step 3. Every agent recalculates individual policy by taking into account the planned arrival of new army units.

Step 4. Agents perform actions.

In the next section validation of the introduced method is given.

## 5 Validation of Proposed Method

Based on the individual strategy and announcement based policy shaping method introduced in Section 4 a multiagent solution was developed for the CERN Grid Wars competition. To illustrate outcomes, the multiagent solution was run only 10 steps with different settings, and the results were recorded. Firstly, the multiagent solution was run with an only individual strategy for each agent in two different settings one with neighbour collaboration radius=1 another with neighbour collaboration radius=5 (Fig. 4).

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 4 | 6 | 7 | 6 | 2 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 8 | 10 | 10 | 9 | 8 | 7 | 6 | 3 | 0 | 0 |
| 0 | 3 | 12 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 2 | 0 | 0 |
| 0 | 7 | 12 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 1 | 0 | 0 |
| 0 | 0 | 4 | 11 | 10 | 9 | 8 | 7 | 6 | 3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 9 | 8 | 7 | 6 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(a) neighbour collaboration radius = 1

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 7 | 11 | 11 | 8 | 7 | 9 | 2 | 0 | 0 |
| 0 | 0 | 1 | 6 | 7 | 7 | 7 | 6 | 6 | 6 | 6 | 1 | 0 |
| 0 | 0 | 2 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 8 | 8 | 0 |
| 0 | 3 | 11 | 7 | 7 | 7 | 6 | 6 | 7 | 8 | 8 | 4 | 0 |
| 0 | 0 | 0 | 3 | 9 | 6 | 7 | 6 | 6 | 7 | 2 | 0 | 0 |
| 0 | 0 | 0 | 0 | 6 | 6 | 8 | 7 | 6 | 7 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 4 | 8 | 9 | 6 | 4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 3 | 7 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b) neighbour collaboration radius = 5

**Fig. 4.** Individual strategy demonstration results

To remember the target is to get maximum population and maximum occupied cells. With the neighbour collaboration radius=1, the result after 10 steps is 54 occupied cells and total army 346 units (Fig. 4a). While with the neighbour collaboration radius=5 the result after 10 steps is 64 occupied cells and total army 388 units (Fig. 4b). The increase in the neighbour collaboration radius improves the results. The radius should not exceed ½ of the grid world diameter as by exceeding ½ of the diameter cause agents to check the same neighbours double.

Secondly, the multiagent solution was run with individual strategy plus announcement based policy shaping method. Again with neighbour collaboration radius 1 and neighbour collaboration radius 5 (Fig. 5).

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 6 | 2 | 6 | 6 | 6 | 6 | 2 | 1 | 0 |
| 0 | 0 | 8 | 9 | 8 | 7 | 6 | 6 | 7 | 7 | 6 | 6 | 3 |
| 0 | 3 | 10 | 9 | 8 | 7 | 6 | 6 | 7 | 7 | 6 | 6 | 2 |
| 2 | 6 | 10 | 9 | 8 | 7 | 6 | 6 | 7 | 7 | 6 | 6 | 1 |
| 0 | 8 | 14 | 8 | 7 | 6 | 6 | 6 | 7 | 6 | 6 | 3 | 0 |
| 0 | 0 | 0 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 7 | 1 | 4 | 3 | 2 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(a) neighbour collaboration radius = 1

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 2 | 6 | 6 | 10 | 6 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 6 | 6 | 6 | 6 | 6 | 2 | 0 | 0 |
| 0 | 0 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 2 | 0 |
| 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 0 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 0 |
| 0 | 0 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 12 | 0 |
| 0 | 0 | 0 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 3 | 0 | 0 |
| 0 | 0 | 0 | 0 | 7 | 9 | 6 | 6 | 6 | 3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 3 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(b) neighbour collaboration radius = 5

**Fig. 5.** Announcement based policy shaping demonstration results

For announcement based policy shaping method with the neighbour collaboration radius=1, the result is 73 occupied cells and total army 405 units (Fig. 5a). While with the neighbour collaboration radius=5 the result is 76 occupied cells and total army 429 units (Fig. 5b). Use of announcement based policy shaping method improves occupied cell count and total army amount proving the utility of the method. Summarized results are given in Table 2.

**Table 2.** Summarized results

| No | Method | Neighbour collaboration radius | Occupied cells | Total army |
|---|---|---|---|---|
| 1 | Individual strategy | 1 | 54 | 346 |
| 2 | Individual strategy | 5 | 64 | 388 |
| 3 | Announcement based policy shaping + individual strategy | 1 | 73 | 405 |
| 4 | Announcement based policy shaping + individual strategy | 5 | 76 | 429 |

It is worth mentioning that described individual strategy together with announcement based policy shaping was the winning strategy at CERN Grid Wars 2018 in Riga, Latvia. The developed multiagent solution used neighbour collaboration radius = 10, won 100% battles in the playoffs and earned 1st place during the competition.

## 6 Conclusions and Future Work

The introduced announcement based policy shaping method for decentralized multiagent coordination showed significant overall results. Performance of multiagent system can be improved by increase of neighbour collaboration radius and by the introduction of announcement based policy shaping collaboration method. The method validation results show that even without centralized coordination good enough outcome can be reached.

The method presented in this paper has a high potential for further development. Agents can improve by introducing multi iteration announcement and shaping strategy where multiple announcements of planned actions during a single step could be made and individual plans several times adjusted. Even more, agents could introduce an agreement mechanism when neighbours agree on next local action and through a neighbour chain propagate common agreement to the whole system for the next step. Moreover, analysis of expected actions of opponent team and formation of collective strategy based on announcement based policy shaping method against it is a potential research subject. Finally, multiagent learning methods based on announcement based policy shaping experience can be developed.

## References


1. Durfee E. H., Distributed problem solving and planning. In Weiss G. ed: Multiagent Systems, The MIT Press: Cambridge MA, pp 121-164, 1999.
2. Sandholm T., Larson K., Anderson M., Shehory O. & Tohme F., Coalition structure generation with worst case guarantees. Artificial Intelligence 111, pp 209-238, 1999.
3. Russell S. & Norvig. P., Artificial Intelligence. A Modern Approach, 3rd ed. Prentice Hall, Upper Saddle River, New Jersey, 2010, 1152 p.

Wooldridge M., An Introduction to MultiAgent Systems 2nd ed. John Wiley & Sons Ltd, The Atrium, Sothern Gate, Chichester, West Sussex, United Kingdom, 2009, 492 p.