Proceedings of Grammar Systems Week 2004, edited by E. Csuhaj-Varjú and Gy. Vaszil, MTA SZTAKI, Budapest, pages 300 - 312, 2004.

Multiple Robots in Space: An Adaptive Eco-Grammar Model

Peter Sebestyén

Mathematical Institute, Silesian University,
Opava, Czech Republic,
Peter.Sebestyen@math.slu.cz

Petr Sosík

Institute of Computer Science, Silesian University,
Opava, Czech Republic,
and
Department of Computer Science, The University of Western Ontario,
London, ON, Canada, N6A 5B7
sosik@csd.uwo.ca

Abstract

We discuss aspects of multi-robot models suitable for study of interactions and emergence of rational behavior. To demonstrate advantages of the grammatical approach, we design an eco-grammatical model of adaptive multi-robot community. We demonstrate that this grammatical model can naturally involve a reinforcement collective learning. We test two learning algorithms in an environment with minimal communication of (almost) reactive robots. Experimental results show that using the eco-grammatical model, the robot community can be successfully trained to find a close-to-optimal solution to a given NP-complete task of a truss construction.

1 Models of Social Robot Behavior

The design and understanding of adaptive multi-agent systems is one of the most important challenges for theoretical robotics and artificial intelligence as well. There has been a movement in the last decade from the traditional idea of monolithic fabricated intelligence to the paradigm of *social intelligence*, see [1, 3] and others. An intelligent behavior and learning tends to be interpreted as a result of interaction of multiple autonomous agents. Examples of successful multi-agent robotic design include the walking six-legged robot Genghis [2], robotic soccer [10] or rescue robot team simulation [18], to name a few. In this paper we demonstrate that *eco-grammar* systems can be particularly useful for modelling and understanding emergence of social behavior of such simple agents.

There have been already reported some symbol- and grammar-based robotic models. For instance, Flann et al. [11] used a grammar-based approach to generate

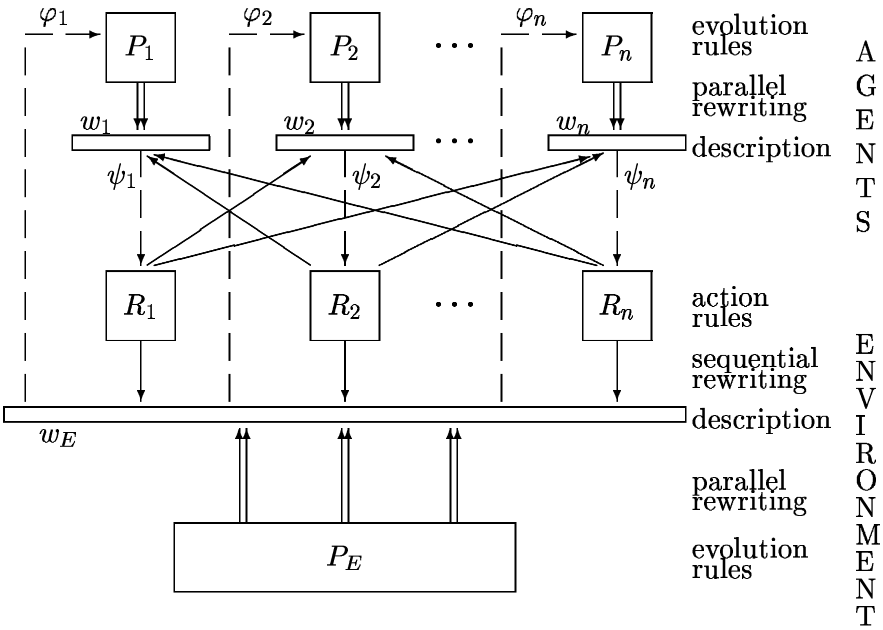

Figure 1: A basic scheme of eco-grammar system.

strategies for autonomous vehicles. Takadama et al. [17] studied a grammar-based robot community learning to perform a collective task. Dassow [9] used an eco-grammar system for modelling the behavior of the MIT robot Herbert. Due to [17], advantages of grammatical approach include (i) guarantee of a necessary level of rational behavior of each robot (ii) easy adaptation of robots to possible defective or inoperative functions (iii) autonomous behavior and emergence of rationality with minimal communication.

We add that (iv) clear interpretation of a robot's behavior and decision is obtained, (v) a successful implementation of collective learning algorithms is possible, and (vi) an analytical solution to the problem of deadlocks or cyclic behavior of the community is available under some restrictions.

2 Eco-Grammar Systems

The *eco-grammar* system was originally created as a grammatical model of interactions between an environment and organisms living in it. For motivation, examples and basic results of eco-grammar systems we refer to [5], [6].

Various models of eco-grammar systems have been studied, taking into the account direct mutual interactions between agents (predator – prey relations), prescribed teams of agents, prescribed scenarios (tables) of agent behaviors, multiplication and dying of agents etc. For these and further variants of eco-grammar systems we refer the reader to [7, 13, 14].

Prior to formal definitions, we fix some basic notation. For an alphabet V, we denote by V^+ the set of all nonempty strings over V. If the empty string, λ, is included, then we use notation V^*. The length of a string x is denoted by $|x|$; $\text{alph}(x) \subseteq V$ denotes the set of all letters contained in x. An *0L scheme* (an interactionless Lindenmayer scheme) is a pair $G = (V, P)$, where V is an alphabet and P is a complete set of context-free rewriting rules applied in parallel. For a set of context-free rules P we denote by $\text{dom}(P)$ the set of left-hand sides of the rules in P, i.e. $\text{dom}(P) = \{a \mid (a \to v) \in P\}$. We refer to [15] for further elements of formal language theory.

Definition 2.1 An *eco-grammar system* of degree $n, n \geq 1$, is an $(n+1)$-tuple $\Sigma = (E, A_1, \dots, A_n)$, where

- $E = (V_E, P_E)$,
  - V_E is a finite alphabet and
  - P_E is a finite set of 0L rewriting rules over V_E, and
- $A_i = (V_i, P_i, R_i, \varphi_i, \psi_i)$ for i, $1 \leq i \leq n$, where
  - V_i is a finite alphabet,
  - P_i is a finite set of 0L rewriting rules over V_i,
  - R_i is a finite set of rewriting rules of the form $x \to y$ with $x \in V_E^+$, $y \in V_E^*$,
  - $\varphi_i : V_E^* \longrightarrow 2^{P_i}$,
  - $\psi_i : V_i^+ \longrightarrow 2^{R_i}$.

(φ_i and ψ_i, $1 \leq i \leq n$, are always supposed to be computable functions but $\varphi_i(x), \psi_i(y)$ are not necessarily complete; 2^X denotes the power set of X.)

The above items are interpreted as follows:

- E represents the environment: V_E is the alphabet and P_E is the set of evolution rules.
- $A_i = (V_i, P_i, R_i, \varphi_i, \psi_i)$ corresponds to an agent (the i-th agent), $1 \leq i \leq n$: V_i is the agent's alphabet, P_i is the set of evolution rules, R_i is the set of action rules. The mapping φ_i, depending on the state of the environment, selects the actual evolution rules. The mapping ψ_i selects the rules for the actual action depending on the current state of the agent.

Thus the 0L schemes (V_E, P_E) and (V_i, P_i), $1 \leq i \leq n$, describe the evolution of the environment and the agents, respectively. The production sets $R_1, \dots, R_n$ define possible actions of the agents $A_1, \dots, A_n$ within the environment. Figure 1 gives a schematic description of the system.

Definition 2.2 A *configuration* of an eco-grammar system Σ is a $(n+1)$-tuple

$$\sigma = (w_E; w_1, w_2, \dots, w_n),$$

where $w_E \in V_E^*$ and $w_i \in V_i^*$, $1 \leq i \leq n$; w_E is called the *state* (evolution stage) of the environment, and w_i is the *state* of the i-th agent, $1 \leq i \leq n$.

In this paper we deal with the *weak* variant of eco-grammar system [4]. An agent A_i is said to be *active* in a configuration $(w_E; w_1, w_2, \ldots, w_n)$ if the set of its actual action rules $\psi_i(w_i)$ is nonempty. Agents can be active only if they are in a nonempty state; formally, $\psi_i(\lambda) = \emptyset$, $1 \leq i \leq n$.

A derivation step of an eco-grammar system from a configuration $(w_E; w_1, \ldots, w_n)$ to $(w'_E; w'_1, \ldots, w'_n)$ means:

(i) an application of a randomly chosen rule $r \in \psi_i(w_i)$ of each active agent A_i to a randomly chosen environment al symbol;

(ii) 0L rewriting of the environment w_E to w'_E by the productions of P_E in every place except the symbols already rewritten in phase (i);

(iii) a simultaneous 0L rewriting of every w_i to w'_i by productions of $\varphi_i(w_E)$.

If an agent A_i enters the empty state $w_i = \lambda$, then it remains inactive during the further functioning of the EG system. On the contrary, if $w_i \neq \lambda$ and no action rule of the agent is selected, i.e. $\psi_i(w_i) = \emptyset$, then the agent is only temporarily inactive. The same holds also if none of the agent's selected rules can be recently applied to the environment. For an example of functioning of an EG-system we refer to Section 3.1.

A uniform formal description of eco-grammar system allows one to solve analytically some important problems of behavior of a multi-robot community. For instance, consider the problem of detection of a *deadlock* (*blocking*): is there a situation during performing a common task when one or more robots block mutually their resources and cannot further act? Another problem is the one of *stagnating* behavior: are there situations when a team is trapped in a cycle? Under certain restrictions, answers to these questions can be effectively found in an eco-grammar system. The following result has been shown in [6].

Theorem 2.3 *There exists a polynomial-time algorithm (w.r.t. to the size of the system) deciding whether the development of a given unary deterministic simple EG system is blocking or stagnating.*

Also for *tabled* EG systems, in which agents switch between prescribed tables of behavior, a similar result has been shown [16]:

Theorem 2.4 *There exists an algorithm deciding for a conditional tabled $EG_1(i, 0; p)$ system, $i \geq 0$, whether there exists an infinite acyclic sequence of configurations (i.e. the system is non-stagnating).*

We refer the reader to [6, 8, 16] for further information on unary, deterministic and tabled eco-grammar systems.

3 A Case Study: Multiple Robots in Space

It follows by the previous description that the eco-grammar system's architecture corresponds to a team of behavior-based robots. There is a common environment

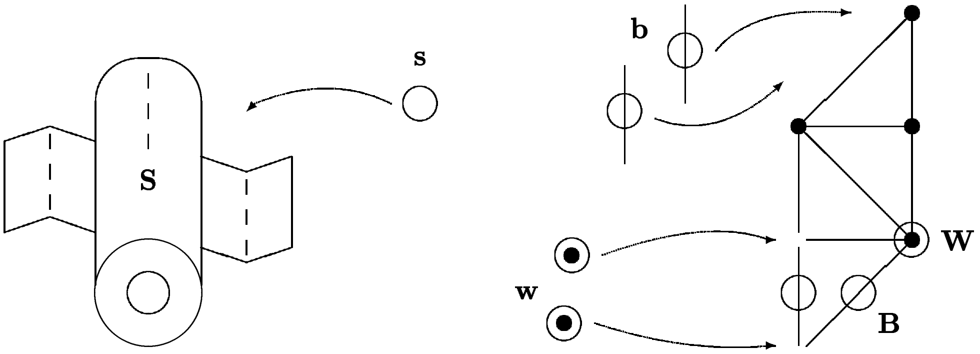

Figure 2: Multiple robots in space. Capital letters denote positions: **S** – space station, **W** – welding position, **B** – beam-holding position. Small letters denote actions: **b** – go to beam-holding position, **w** – go to welding position, **s** – go to station.

w_E, subject to an independent evolution rules P_E. A *perception* function for an agent A_i is defined by φ_i, an *action selection* function is ψ_i and action rules are R_i. The learning task for multiple robots described bellow has been originally presented in [17], where a hybrid model with grammatical aspects was presented.

Consider that there are m beams in a space station which have to be transported and weld together to construct a truss. There are n universal robots to participate at a solution. Each robot can perform one of the following actions:

- *transport a beam* from the station to the truss;
- *hold a beam* in a suitable position within the truss;
- *weld a beam* held by another robot;
- *transfer* between two locations.

These elementary actions can assemble to patterns of a robot behavior — *roles*, e.g. to transport a beam and then hold it until it is weld by another robot. The situation when the number of robots ready to weld is greater than the number of the other robots is considered as a *deadlock*. (The ready-to-weld robots block the access of the others to the truss.)

In our eco-grammar model the environment string w_E contains a global description of a current state of the task. The 0L scheme (V_E, P_E) defines the behavior and properties of the environment. Each robot is represented by an agent A_i, $1 \leq i \leq n$. Its internal state w_i represents its current position and a sequence of its previous actions for the purpose of training. Selection of a current role of the robot is driven by rules in the set P_i. Similarly as in [17], some rules in P_i have assigned individual weights that can be applied in a deterministic or probabilistic way.

Formally, consider a truss construction task with n robots and m beams. Let $\Sigma = (E, A_1, \ldots, A_n)$ be a weak EG system describing fully the task. Denote $d(X, Y)$ the number of steps to move from the position X to Y, for $X, Y \in \{S, B, W\}$.

- $V_E = \{F, G, \#\} \cup \{D_i, H_i, I_i, \#_i \mid 1 \le i \le n\} \cup \{D_{i[XY]} \mid 1 \le i \le n;\ X, Y \in \{B, S, W\}\}$;

- $P_E = \{D_{i[XY]} \to D_i^{d(X,Y)} \mid 1 \le i \le n,\ X, Y \in \{B, S, W\}\} \ \cup\ \{\#_i \to \# \mid 1 \le i \le n\}$ (recall that D^n denotes the string consisting of n symbols D) plus completing rules of the form $a \to a$ for the other symbols $a \in V_E$ not present on the left-hand side of the above rules.

Meaning of symbols in V_E is the following:

D_i	a travel distance element for the robot A_i;
G	a beam located in the station;
H_i	a beam transported by the robot A_i;
I_i	a beam held by the robot A_i in a beam holding position;
$\#$	a beam weld with the truss;
F	an auxiliary symbol – port for one robot.

Let $A_i = (V_i, P_i, R_i, \varphi_i, \psi_i)$, $1 \le i \le n$, be a complete description of a robot.

- $V_i = \{S, B, W, W_x, P_S, P_B, P_W, s, b, w, R_{SB}, R_{BS}, R_{BW}, R_{WS}, R_{WW}, R_{SW}\}$;

- The mappings φ_i and ψ_i are defined as follows. Let for each rule $x \to y$ in P_i there be a set $C \subseteq V_E$ of *permitting* symbols and a set $D \subseteq V_E$ of *forbidding* symbols. Then

$$x \to y \in \varphi_i(w_E) \text{ iff } C \subseteq \text{alph}(w_E) \text{ and } D \cap \text{alph}(w_E) = \emptyset.$$

  The mapping ψ_i is defined in the same way w.r.t. the rules in R_i and the string w_i. Due to the above convention each rule in P_i or R_i will be denoted as $(C, D : x \to y)$. This type of regulation can be found in random context grammars [15] or conditional tabled eco-grammar systems [8].

- P_i contains the following rules:

$$\begin{array}{lll}
(\{G\}, \emptyset : S \to R_{SB})^\dagger & (\emptyset, \emptyset : R_{SB} \to P_B) & (\{D_i\}, \emptyset : P_W \to P_W) \\
(\{G\}, \emptyset : S \to R_{SW})^\dagger & (\emptyset, \emptyset : R_{BS} \to P_S) & (\{D_i\}, \emptyset : P_S \to P_S) \\
(\emptyset, \{H_i, I_i\} : B \to R_{BW})^\dagger & (\emptyset, \emptyset : R_{BW} \to P_W) & (\{D_i\}, \emptyset : P_B \to P_B) \\
(\emptyset, \{H_i, I_i\} : B \to R_{BS})^\dagger & (\emptyset, \emptyset : R_{WS} \to P_S) & (\emptyset, \{D_i\} : P_S \to sS) \\
(\{H_i\}, \emptyset : B \to B) & (\emptyset, \emptyset : R_{WW} \to P_W) & (\emptyset, \{H_i\} : P_B \to S) \\
(\{I_i\}, \emptyset : B \to B) & (\emptyset, \emptyset : R_{SW} \to P_W) & (\{H_i\}, \{D_i\} : P_B \to bB) \\
(\{\#_i\}, \emptyset : W_x \to wR_{WW})^\dagger & (\emptyset, \{\#_i\} : W_x \to W) & (\emptyset, \{D_i\} : P_W \to W) \\
(\{\#_i\}, \emptyset : W_x \to wR_{WS})^\dagger & (\{G\}, \emptyset : W \to W_x) &
\end{array}$$

$$(\emptyset, \{G\} \cup \{H_j, I_j \mid 1 \le j \le n\} : W \to R_{WS})$$
$$(\emptyset, \{G\} \cup \{H_j, I_j \mid 1 \le j \le n\} : S \to \lambda)$$

$$\left.\begin{array}{ll}
(\{H_j\}, \emptyset : S \to R_{SW})^\dagger & (\{H_j\}, \emptyset : W \to W_x) \\
(\{I_j\}, \emptyset : S \to R_{SW})^\dagger & (\{I_j\}, \emptyset : W \to W_x)
\end{array}\right\} 1 \le j \le n,$$

  plus completing rules of the form $a \to a$. Rules denoted by $\dagger$ can have assigned weights described in the next section.

- R_i contains the following action rules:

$$\begin{array}{ll} (\{R_{BS}\}, \emptyset : F \to FD_iD_{i[BS]}) & (\{R_{WS}\}, \emptyset : F \to FD_iD_{i[WS]}) \\ (\{R_{BW}\}, \emptyset : F \to FD_iD_{i[BW]}) & (\{R_{WW}\}, \emptyset : F \to FD_iD_{i[WW]}) \\ (\{R_{SW}\}, \emptyset : F \to FD_iD_{i[SW]}) & (\{R_{SB}\}, \emptyset : G \to H_iD_iD_{i[SB]}) \\ (\emptyset, \emptyset : D_i \to \lambda) & (\{B\}, \emptyset : H_i \to I_i) \end{array}$$

plus the set $\{(\{W\}, \emptyset : I_j \to \#_i) \mid 1 \le j \le n,\ j \ne i\}$.

Meaning of symbols in V_i, $1 \le i \le n$, is the following:

S	a robot is in the station;
B	a robot is in a beam holding location;
W, W_x	a robot is in a beam welding location;
P_Y	for $Y \in \{S, B, W\}$: a robot is in space on the way to location Y;
b	a robot transported a beam to a beam holding location;
s	a robot returned to the station;
w	a robot weld beams at a welding location;

3.1 An example of a simple task

As mentioned above, the rules in P_i denoted by † can have assigned weights. These rules determine the choice of current roles for robots. In our experiment the robots have two different main roles: *transport and hold a beam* and *(wait and) weld a beam*. For simplicity, we fix the distance between the space station and the truss to $d(S, B) = d(S, W) = 1$ time step and the distance between welding and beam-holding position within the truss to $d(B, W) = d(W, W) = 0$.

Example 3.1 Consider $n = 3$, let two robots A_1, A_2 transport and hold beams an let the third one A_3 weld beams. The weights will be set as follows:

	A_1	A_2	A_3
$S \to R_{SB}$	0.9	0.9	0.1
$S \to R_{SW}$	0.1	0.1	0.9
$B \to R_{BS}$	0.9	0.9	0.1
$B \to R_{BW}$	0.1	0.1	0.9
$W_x \to wR_{WS}$	0.9	0.9	0.1
$W_x \to wR_{WW}$	0.1	0.1	0.9

The sequence of configurations of the whole community follows, beginning in an initial configuration with all the beams and robots located in the station.

Step	Environment w_E	A_1	A_2	A_3
0.	F^3G^2	S	S	S
1.	F^3G^2	R_{SB}	R_{SB}	R_{SW}
2.	$F^3D_3D_{3[SW]}H_1D_1D_{1[SB]}H_2D_2D_{2[SB]}$	P_B	P_B	P_W
3.	$F^3D_3H_1D_1H_2D_2$	P_B	P_B	P_W
4.	$F^3H_1H_2$	P_B	P_B	P_W

5. $F^3H_1H_2$	bB	bB	W
6. $F^3I_1I_2$	bB	bB	W_x
7. $F^3I_1I_2$	bB	bB	W

Now robots 1 and 2 are holding beams and robot 3 is ready to weld. After welding the robot 1's beam, robot 3 continues to another welding position, while robot 1 returns to the station. Then the same is done with the robot 2.

8. $F^3\#_3I_2$	bB	bB	W_x
9. $F^3\#I_2$	bR_{BS}	bB	wR_{WW}
10 $FFD_1D_{1[BS]}FD_3D_{3[WW]}\#I_2$	bP_s	bB	wP_W
11. $FFD_1F\#I_2$	bP_s	bB	wP_W
12. $F^3\#I_2$	bP_s	bB	wW
13. $F^3\#\#_3$	bsS	bB	wW_x
14. $F^3\#\#$	bs	bR_{BS}	wwR_{WW}
$\vdots$ $\vdots$	$\vdots$	$\vdots$	$\vdots$
23. $F^3\#\#$	bs	bs	wws

In the final configuration all the beams are weld, all the robots are back at the station, state of each robot contains the history of its actions.

4 The Learning Algorithms

The task of coordination of multiple robots to fit into a given time limit is generally NP-complete. It is a case of so-called Precedence Constrained Multiprocessor Scheduling [12]. Hence, in larger robot communities, the problem of central control strategy is computationally unfeasible. The character of the task suggests as a solution the *Profit Sharing* (PS) reinforcement collective learning.

4.1 PS reinforcement learning

In the learning process the rules with assigned weights are selected by a roulette selection. After each task completion, positive or negative rewards are distributed to robots due to the result. Through the rewards distribution robots modify their own roles by changing the weight values of their individual rules. In accordance with [17], we modify the distribution of rewards to a *decreasing* (instead of original *increasing*) geometrical progression.

Positive rewards are distributed according to the formula:

$$W(r_i) := W(r_i) + R^+ \cdot (1/2)^{i-1}, \quad i = 1, ..., p. \tag{1}$$

Negative reward is distributed just to the last selected rule:

$$W(r_i) := W(r_i) - R^-, \text{ where } \; i = p. \tag{2}$$

In the above formulas $W(r_i)$ represent the weight of the rule r_i, R^+ and R^- are nonnegative reward values, p is the length of the current role-selection sequence of the robot. As an assumption, the weights of rules are set to the same value at the beginning of training.

4.2 Advanced reward assignment

As one can observe in Section 5, the robot community using the PS learning was able to learn the task successfully. If the ratio of beams/robots was high, however, the experimental results were worse than the estimated optimum. Furthermore, a large number of training cycles was necessary and a robustness of training was rather weak. For instance, in the case of 3 robots and 24 beams only one of 200.000 training cycles leaded to performing the task successfully.

We deduce that the PS learning algorithm did not motivate robots to effectivity, it just leaded to any non-deadlock solution. A improvement can be gained by scaling the positive reward value in dependence on the number of steps needed to complete the task:

$$W(r_i) := W(r_i) + (\ell/k)^{\alpha} \cdot R^{+} \cdot (1/2)^{i-1}, \quad i = 1, ..., p \ , \tag{3}$$

where ℓ is the estimated optimal length of the task, k is the number of steps in the success case and α is a positive integer. With growing value of α the reinforcement of weight values depends stronger on number of steps k. Moreover, unlike the previous section, the robots were trained to solve the task with subsequently increasing number of beams.

5 Simulation Results

We compare the number of steps needed for completing a given task in three experiments. In the first experiment used as a benchmark we perform the truss construction task described in section 3 with pre-set values of weights. This experiment called *Central Control.* In the second experiment named *PS Learning* we used the modified PS reinforcement learning from Section 4.1. In the third experiment denoted by *Advanced Learning* we applied the learning strategy from Section 4.2.

We performed all the experiments for the case of 13 and 24 beams. Figures 3 and 4 allows one to compare the results. When considering experimental results, we must remember that there is also certain number of "delay" steps due to the transfer of information in eco-grammar system between environment and the robots. An original Lisp implementation of eco-grammar system was used for the experiments.

5.1 Experiment with central control

In this experiment we set the weight values and hence divided the roles manually. Two main roles are denoted as follows:

- *HB* for transport and hold beam;
- *WB* for weld beam.

As the problem is NP-complete, we did not calculate an exact analytical minimum $s_{\min}(m, n)$ of steps depending on n (the number of robots) and m (the number of beams), for a given distances $d(X, Y)$, $X, Y \in \{B, S, W\}$. We can nevertheless give a certain upper and lower bound of this minimum.

For a lower bound, consider the case when the welding procedure and the distance $d(W, W)$ is very short, majority of the robots bring and hold beams and only a small fraction of robots weld beams. Then clearly at least $\lfloor m/n \rfloor$ beam-holding cycles are necessary and hence

$$\left\lfloor \frac{m}{n} \right\rfloor \cdot c_1 + c_2 \leq s_{\min}(m,n) \tag{4}$$

for some constants $c_1 \geq 1$, and $c_2 \geq 0$ depending on $d(X,Y)$.

For an upper bound, assume on the contrary that the welding procedure and/or the transfer between welding locations takes more time than the beam bringing and holding. If we divide the robots to two teams of roughly the same size, the first one with the roles *HB* and the second one with the roles *WB*, we can certainly perform the task in $\lceil m/(2n) \rceil$ welding cycles, and hence

$$s_{\min}(m,n) \leq \left\lceil \frac{m}{2n} \right\rceil \cdot c_3 + c_4 \tag{5}$$

for positive constants c_3 and c_4. It follows from (4) and (5) that

$$s_{\min}(m,n) = \Theta\left(\frac{m}{n}\right). \tag{6}$$

In the experiment, robots performing *HB* had the weight values set as the robot A_1 in example 3.1. The *WB*-robots had the weight values as the robot A_3. We studied the cases from 3 to 15 robots and distributed the roles in a following way to estimate the optimal solution:

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
HB	*HB*	*WB*	*HB*	*WB*	*HB*	*HB*	*WB*	*HB*	*HB*	*WB*	*HB*	*HB*	*WB*	*HB*

5.2 Experiments with PS learning

The experiments with PS learning were performed for 3 to 15 robots. We used from 2,000 to 200,000 training cycles, reward values were $R^+ = 1$ and $R^- = 0.01$. The initial value of rule weights was set to 50. Since results of the experiments were nondeterministic due to the probabilistic learning algorithm, we have chosen for each value of n (the number of robots) the *best* result among 10 randomly chosen experiments.

5.3 Experiment with advanced learning

The robots were trained by a learning procedure described in Section 4.2. Denoting n the number of robots, we increased the number of beams by $2 * \lceil n/2 \rceil$ in each learning segment. This leaded to approx. ten times higher effectivity of learning — no more than 20.000 training cycles were necessary. Similarly as in the experiment with PS learning, the *best* result among 5 randomly chosen experiments was chosen. Unlike PS learning, this time more than 50% of the experiments resulted in the same value shown in the graph and only in a few cases the result was worse. Hence the advanced learning algorithm is much more robust. Moreover, the results were better for a higher ration m/n.

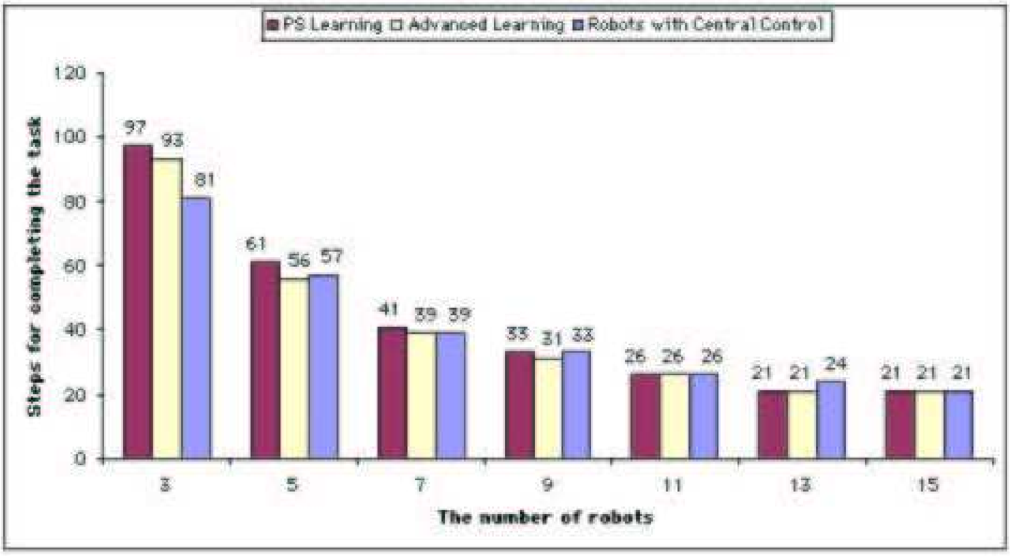

Figure 3: The experimental results for 13 beams

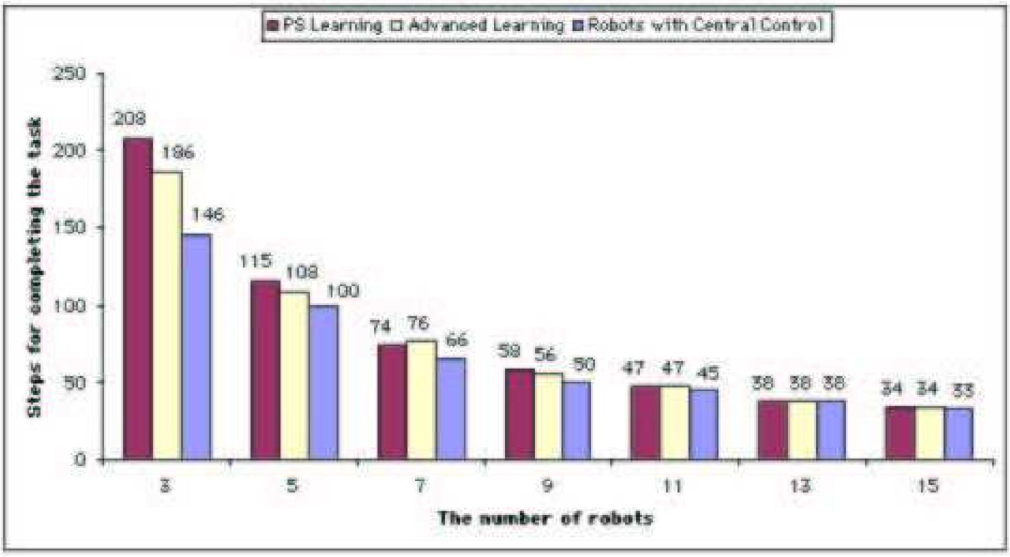

Figure 4: The experimental results for 24 beams

6 Discussion

We considered aspects of multi-agent and multi-robot modelling and discussed advantages of the grammatical approach. Clearly, this approach is appropriate in situations when details of physical interactions of robots and their environment are involved (e.g. robot vision, image recognition, control of drives etc.). But it provides a favorable tool on a certain level of abstraction when the environment, perception and actions of robots can be described by discrete means. We argue that substantial aspects of emergence of rational behavior and social intelligence could (and should) be studied on such a level.

As an example we presented an eco-grammar model of and adaptive robot community in space. We tested in this framework two algorithms of distributed reinforcement learning: a modified PS learning and our advanced learning algorithm. The community of robots was able to perform the given task of the truss construction with both algorithms. Our advanced learning turned out to be more robust and also about ten times more effective. Rather surprisingly, the learning model has reached in many cases the estimated optimal solution.

Further theoretical and experimental study is necessary to exploit potential of grammatical models of robotic systems. Among many *open problems* we select a few:

- to study systematically models in which an analytical prediction of (un)desired configurations as a deadlock or cycles is possible;
- to accommodate the model with two- or three-dimensional environment, for the cases when the studied experiment would require to involve spatial interactions and topology;
- to study and model possible emergence of rational behavior in collectives of nanomechanisms as molecular motors, bio-arrays etc.

Acknowledgements

Research was partially supported by the Grant Agency of Czech Republic, Grant 201/02/P079 to P.S.

References

[1] R. M. Axelrod: *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration.* Princeton University Press, 1997.

[2] R. A. Brooks: A layered intelligent control system for a mobile robot. *IEEE Journal of Robotics and Automation* **2(1)** (1986), 14-23.

[3] R. A. Brooks: *Cambrian intelligence: the early history of the new AI.* MIT Press, Cambridge, Mass., 1999.

[4] J. Csima, Two remarks on variants of simple eco-grammar systems. *Acta Cybernetica* **14(4)** (2000), 569–582.

[5] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, Gh. Păun: Eco-grammar systems. A preview. In R. Trappl (ed.): *Proc.* 12$^{\text{th}}$ *European Meeting on Cybernetics and System Research,* World Scientific, Singapore, 1994, 941–948.

[6] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, Gh. Păun: Eco-grammar systems. A grammatical approach to life-like interactions. *Artificial Life* **3** (1997), 1–28.

[7] E. Csuhaj-Varjú, A. Kelemenová: Team behaviour in eco-grammar systems. *Theoretical computer science* **209** (1998), 213–224.

[8] E. Csuhaj-Varjú, Gh. Păun, A. Salomaa: Conditional tabled eco-grammar systems versus (E)T0L systems. *JUCS* **1(5)** (1995), 252–268.

[9] J. Dassow, An example of an eco-grammar system: a can collecting robot. In [13], 240–244.

[10] A. Drogoul, A. Collinot: Applying an agent-oriented methodology to the design of artificial organizations: A case study in robotic soccer. *Autonomous Agents and Multi-Agent Systems* **1(1)** (1998), 113–129.

[11] N. S. Flann, M. Davidson, J. Martin, K. L. Moore: Intelligent behavior generation strategy for autonomous vehicles using a grammar-based approach. *International Conference on Field and Service Robotics*, Helsinki, University of Technology, Otaneimi, Espoo, Finland, 2001.

[12] M. R. Garey, D. S. Johnson: *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Freeman and Co., San Francisco, 1979.

[13] G. Păun (ed.): *Artificial Life: Grammatical Models.* The Black Sea University Press, Bucharest, 1995.

[14] G. Păun, A. Salomaa (eds.): *Grammatical models of multi-agent systems*, Gordon and Breach, Yverdon, 1999.

[15] A. Salomaa, G. Rozenberg (eds.): *Handbook of Formal Languages.* 3 volumes. Springer-Verlag, Berlin, 1997.

[16] P. Sosík: On the decidability problems of eco-grammar systems. *Journal of Automata, Languages and Combinatorics* **5** (45–58), 2000.

[17] K. Takadama, K. Hajiri, T. Nomura, K. Shimohara, S. Nakasuka: Grammatical learning model for adaptive collective behaviors in multiple robots. In G. Păun and A. Salomaa (Eds.): *Grammatical Models of Multi-agent Systems,* Gordon and Breach, London, 1999, 343–355.

[18] T. Takahashi, S. Tadokoro, M. Ohta, N. Ito: Agent Based Approach in Disaster Rescue Simulation - From Test-Bed of Multiagent System to Practical Application. RoboCup 2001, 102–111.