Spontaneous speciation by GA for division of labor in two-agent systems

Naoki Oshiro, Koji Kurata and Tetsuhiko Yamamoto
Faculty of Engineering, University of the Ryukyus,
1 Senbaru, Nishihara, Okinawa. 903–0213
oshiro@mibai.tec.u-ryukyu.ac.jp

Abstract

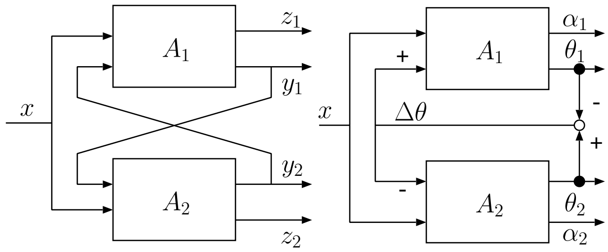
In this paper, we propose a framework of genetic algorithm in which division of labor among the agents emerges. The agents in the system spontaneously develop different charactaristics to cooperate effectively to achive a task and speciate to different species. We assume the agents recieve input to specify the task and can observe each other. We apply the framework to a mechanical system. The task is designed simple and essential to elucidate the effectiveness of the algorithm.

Key Words: spontaneous speciation, coevolution, multi-agent system, genetic algorithm

1 Introduction

There are many researchs about multi-agent system to solve complex problem like n-Traveling Salesman Problem (nTSP)[1], and Genetic Algorithm (GA)[3] with speciation is an important and effective idea for organiging cooperative sharing of a task by agents. Speciation is also an important aspect in evolutional biology. There are also researchs to build a virtual ecosystem with GA-like framework[2].

In this paper, we apply this framework to a mechanical system. We assume two agents cooperating for a task (Figure 1). Each agent A_i receives external input x and generates its outputs y_i and z_i, generally. z_i is the action of the agent, and y_i is the signal to be observed by the other agent. y_i can be identical to z_i or a part of z_i. We assume rather a poor ability for the agents, so that the task is too difficult and two agents must cooperate. For better performance each of the cooperative pair must play a different role. This formulation will give rise to the division of labor. We want to make this division emerge spontaneously. For this purpose, we assume every agents have a same structure, learning mechanism and live in a same environment.



(a) General (b) Customized

Figure 1: Information flow between two agents. (a) General framework. Each agent receives external input x and part of the other agent's output y_i. To achive given tasks, the agents use these information appropriately. (b) Customized framework for our task. Each agent can obtain the difference $\Delta\theta$ between its angle and its coleage's. Agents change their location θ_1, θ_2 only once using this information.

The task adopted in this paper is "disk moving problem", which is simple and essential to test the framework.

2 Disk moving problem

We assume two agents set on a disk in two dimensional space (Figure 2). Initially, the agents are located on the edge of the disk randomly and recieve input x to specify one of two target motions to be caused, "rotation" or "translation". Then they observe the relative angle between them. Based on the observation each of them moves along the circumference to the final position, where the two agents apply force simultaneously in the directions determined by x to cause rotation or translation.

Each agent has two state variables the angle representing the position of the agent on the edge of the disk θ_i and force angle α_i measured counterclockwise from the direction of center of the disk. $\bar{\boldsymbol{f}}_1$, $\bar{\boldsymbol{f}}_2$ are the impulsive force applied by the agents. The forces are assumed to have a fixed absolute value f ($|\bar{\boldsymbol{f}}_1| = |\bar{\boldsymbol{f}}_2| = f$). The forces give two kind of velocities to the disk. One is the angular velocity $\omega = \frac{2f}{MR}(\sin\theta_1 + \sin\theta_2)$, where R, M are radius and mass of the disk. The other is the translation velocity $\boldsymbol{v} = \frac{1}{M}(\bar{\boldsymbol{f}}_1 + \bar{\boldsymbol{f}}_2)$.

Each agent has four parameters $\{a_i, b_i, c_i, d_i\}$ to determine its final position and force angle. These parameters will be coded binary and used as chromosomes for GA. Movement from the initial position θ_i^{init} to the final position θ_i^{fin} is determined by parameters a_i and b_i:

$$\begin{cases} \theta_1^{\mathrm{fin}} := \theta_1^{\mathrm{init}} + a_1(\theta_2^{\mathrm{init}} - \theta_1^{\mathrm{init}}) + b_1, \\ \theta_2^{\mathrm{fin}} := \theta_2^{\mathrm{init}} + a_2(\theta_1^{\mathrm{init}} - \theta_2^{\mathrm{init}}) + b_2. \end{cases} \tag{1}$$

Here we assumed the agents can observe only the relative angle between them, and the positional change is determined by the observation (Figure 1 (b)). After this, each agent apply impulsive force in direction α_i, which is determined as

$$\alpha_i = xc_i + d_i, \tag{2}$$

where x is motion specification ($x = 1$ is for "rotation" and $x = -1$ for "translation" of the disk). Then the rotational and translational velocities of the disk are obtained as described before and are used for evaluating the fitness function of the agents.

It will be easily understood that the agents cannot achive the task with a same behavior, so they must emerge their different characteristics to play different roles. It means division of labor between the agents.

3 Learning method

Let us consider a population of agents with a chromosome $\{a_i, b_i, c_i, d_i\}$ which are coded binary. These parameters range over the following intervals: $a_i \in [-1, 1]$, $b_i, c_i, d_i \in [-\pi, \pi)$. We initialize all agents by assigning random values to all the parameters. We evaluate every two combination of different agents A_i, A_j, $(i \neq j)$. These two agents are set on the edge of the disk randomly and input $x = \pm 1$ is given. Then rotational velocity ω and translational velocity $\boldsymbol{v}$ is calculated as described before. The quality $F_{i,j}$ of the resulting motion is evaluated as:

$$F_{i,j} = x(\omega^2 - |\boldsymbol{v}|^2). \tag{3}$$

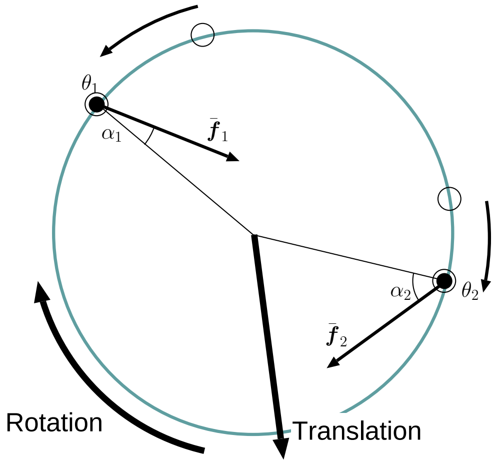


Figure 2: Action of two agents on a disk. Agents have two state variables, location angle θ_i, force angle α_i. To generate rotational or translational velocities effectively.

For this evaluation function $|\omega|$ must be maximized (minimized) and $|\boldsymbol{v}|$ minimized (maximized) when $x = 1$ ($x = -1$). This evaluation process is repeated several times with $x = 1$ and with $x = -1$ same number of times to obtain the averaged evaluation $\bar{F}_{i,j}$, which is used for evaluation of the pair of agents.

After testing all combination of agents, we define the fitness function of agent i as follows:

$$F_i = \max_j\{\bar{F}_{i,j}\}, \tag{4}$$

which is the average quality of the motion given by agent i with its best partner.

After all combination are tested, we apply the standard genetic algorithm to the population of agents. The agents, namely, is sorted according to the fitness values F_i and inferior half of the population are deleted. From the superior half of the population a pair of agents are randomly chosen for mating and two new agents are generated from the pair with crossover operation and mutation (see Figure 3). New agents are generated until the deleted population is recovered.

4 Simulation results

We set the absolute value of impulsive force $f = 1$ and moment of inertia $I = 1$ $(M = 1, R = 2)$. Thus, the maximum value of two kinds of velocities are $|\omega| = 2$ and $|\boldsymbol{v}| = 2$, and the maximum evaluation value is $F_i = 4$ $(= \omega^2_{\max} = |\boldsymbol{v}|^2_{\max})$.

In our GA, we set the parameters as follows: number of the units is 40, bit length of binary expression of

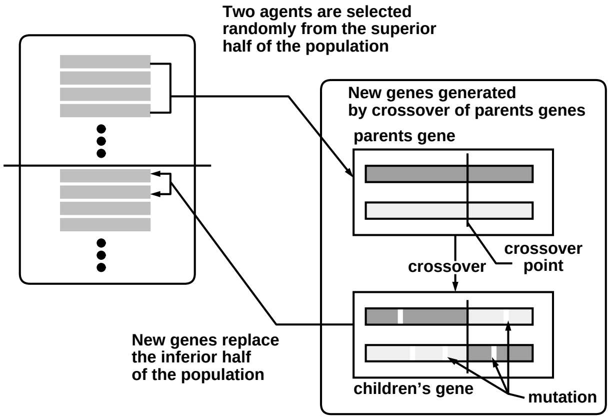


Figure 3: Genetic algorithm of parameters of the agents

each parameter is 28, mutation rate is 5%, and maximum generation is 100. Simulation results are depicted in Figures 4, 5 showing agents' behavior on the disk. An arrow in the figure corresponds to an agent. Root of the arrow represents the value (angle) of parameter b_i and the direction of the arrow represents the force angle α_i. Figure 6 shows the distribution change of (a) b_i and (b) α_i. Values of b_i clearly splits to two values with difference π corresponding to two different types necessary for effective operation.

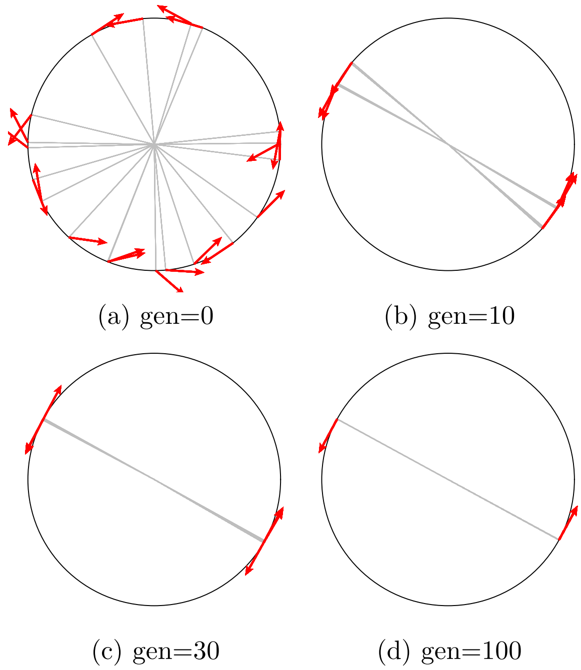
5 Discussion

In optimal condition for disk rotation ($x = 1$), two agents are located at opposite positions on the disk edge ($\theta_1 - \theta_2 = (2n + 1)\pi$). If the final positions of the agents satisfy this condition, we can obtain the following equation from Equation (1)

$$\begin{aligned} \theta_1^{\text{fin}} - \theta_2^{\text{fin}} &= (\theta_1^{\text{init}} - \theta_2^{\text{init}})(1 - a_1 - a_2) + b_1 - b_2 \\ &= (2n - 1)\pi. \end{aligned} \tag{5}$$

To make this hold regardless of the random initial positions θ_1^{init} and θ_2^{init} the following condition is necessary and sufficient: $a_1 + a_2 = 1$ and $b_1 - b_2 = (2n + 1)\pi$. There are infinitely many solutions for these conditions also. Different values are obtained in each GA simulations. The optimal force angles for rotation ($x = 1$) are $\alpha_1 = \alpha_2 = \pm\pi/2$ (Figure 4 (d)), and the optimal force angles for translation ($x = -1$) must satisfy $\alpha_1 - \alpha_2 = (2n+1)\pi$ (Figure 5 (d)). There are infinitely many solutions for these conditions also. Different values are obtained in each GA simulations.

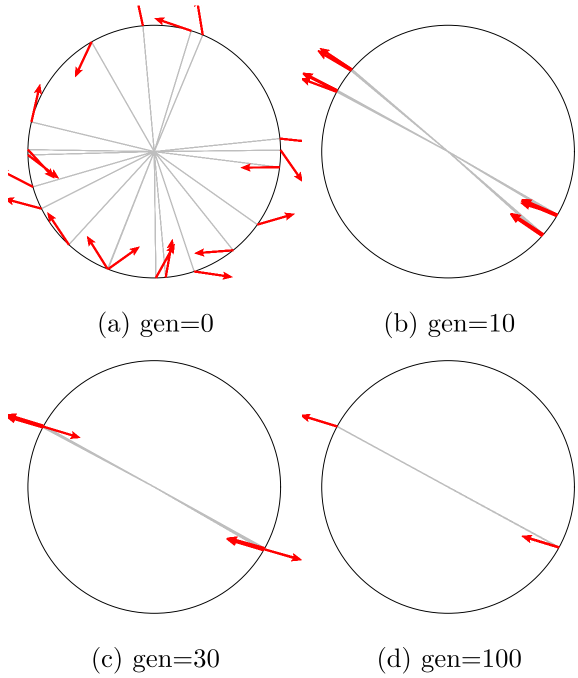
For disk translation ($x = -1$) only there is another solution in which two agents are located at a same



(a) gen=0 (b) gen=10

(c) gen=30 (d) gen=100

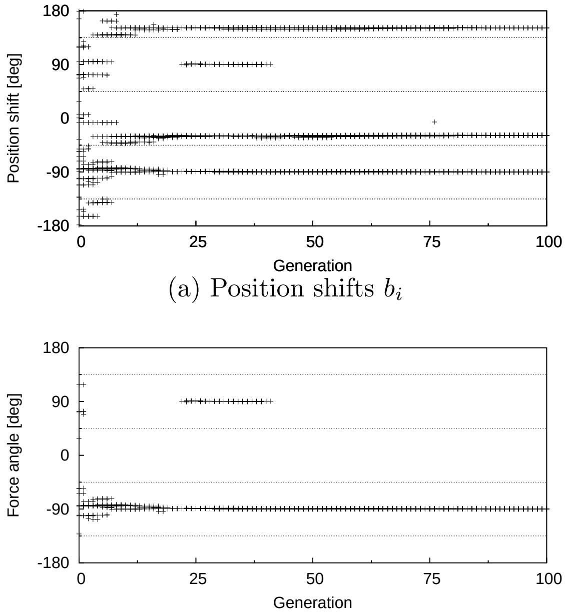
Figure 4: Position shifts b_i and force directions α_i of the agents generated in the simulation ($x = 1$, generation 0 ~100).



(a) gen=0 (b) gen=10

(c) gen=30 (d) gen=100

Figure 5: Position shifts b_i and force directions α_i of the agents generated in the simulation ($x = -1$, generation 0 ~100).



(a) Position shifts b_i

(b) Force angles $\alpha_i (= c_i + d_i)$

Figure 6: Evolution of parameters.

position and a force direction angle for both is π, but this placement is not effective for rotation. Therefore, the placement did not appear in our simulation.

Figure 4 (c) shows that in the early stage of evolution the force angles almost converged to $-\pi/2$, and then the other optimal force angle $\pi/2$ appeared and the two optimal force angles coexisted for a while, but $\pi/2$ disappeared before the 50th generation. This shows the excellent ability of this scheme to search for other optimal solutions. The ability partially comes from the definition of fitness function as the maximum of pairwise evaluations (Equation (4)). An alternative for it is the average of all the pairwise evaluations. This definition, however, might not give the scheme such ability because of the following reason. Suppose a new optimal pair of agents appear, they will not obtain higher fitness value because they cannot cooperate properly with the majority of the agents and average performance will be poor. Thus, they will be eliminated soon.

6 Conclusion

In this paper, we proposed the framework of genetic algorithm to give rise to spontaneous cooperation by division of labor. We apply this method to a simple mechanical problem of disk motion by two agents on the disk. Simulation results show emergence of optimal division of labor by the two agents.

References

[1] Toma, N., Endo, S., Yamada, K., (1998). Immune algorithms for nTSP, Bulltin of the faculty of engineering university of the ryukyus. (in Japanese)

[2] Clement, R., (2003). Multi-Agent Simulations of Evolution and Speciation in Cichlid Fish, *15th European Simulation Symposium and Exhibition ESS2003.* Delft, Holland.

[3] Edited by Davis, L., (1991). Handbook of Genetic Algorithms, *Van Nostrand Reinhold*