30

Generate Parallel Behavior Structure for Automatic Workshop Problem with GP

Wenwei YU, Yukinori KAKAZU
Department of Precision Engineering
Hokkaido University
West-8, North-13, Sapporo, Hokkaido, Japan
Tel: +81-11-706-6443
Fax: +81-11-700-5070
Email: kakazu@hupe.hokudai.ac.jp

Dan JIN
Mechanical Engineering Department
Shanghai Jiaotong University
1954 Huashan Road, Shanghai, China

Abstract

Existing researches that apply hierarchical behavior for AI problem, but in a few instance, hierarchical behavior is abstracted for multi-agent problems solving. In this paper we introduce a scheme that creates a series of behavior structures which pursue parallel mechanism in acquiring the action plan of multi-agent in the case each agent should autonomously make decision, i.e., make decision independently according to the dynamic environment. GP(Genetic Programming) is used for evolution of the behavior structure. By solving the Automatic Workshop Problem wherein the working and moving schedule should be made by each robot considering obstacle avoidance, process order and working ability of itself, we show that the scheme is viable and promising.

Keywords

Multi-agent system, Parallel, Behavior, Genetic Algorithm, Genetic Programming, Job Process Problem, Automatic Workshop Problem, AGV Plan, Control Problem

1 INTRODUCTION

1.1 Background

Developing schedule for automatic workshop is a task more complex than ordinary Job Process Schedule Problem, because of the combination of variety of robots and moving trajectory conflict besides of the job process order. The space of feasible schedules grows exponentially as there are increases in the number of different jobs that must be processed, number of operations required by each job, number of kinds of robot and number of robot of each kind. This multi-variable changeability makes the traditional scheduling, i.e., collective and supervisory scheduling method lack of flexibility and even impossible, while finishing

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

the works by several agents with different level of hierarchical behavior rules shows feasibility and effectiveness to be high level learning and searching scheme.

On the other hand, the searching for an optimal or even sub optimal behavior structure for each kind of robot also suffers from unrealistic search time and computational complexity while the condition and action grows. Recently, there has been wide research interest in applying model-less methods such as GA(Genetic Algorithm), GP(Genetic Programming), NN(Neuron Network) and RL(Reinforcement Learning) to the searching and learning problem[1][2]. These kind of methods can solve the problem with less dependence on the prior knowledge and model in less time in comparing with conventional algorithms. Among these model-less methods, GP is a promising one, which is an adaptive optimization algorithms based on principles of natural evolution[2]. In this paper, a series of hierarchical behavior rule sets represented by behavior tree are created using GP for each agent, it is the agents that autonomously search the path and coordinate to complete the tasks.

1.2 The Problem Description

The automatic working factory problem can be described like followings:

Definition of F, C and A

F= {R, M, B}

where

R= $\{S_i , i = 1 \sim SPECIE\_NUM\}$

S= $\{r_j , j = 1 \sim robot\_number(i)\}$

M= {W, N}

W= $\{w_k^{\,l} , k = 1 \sim WORK\_NUM, l = work\_level\}$

N= $\{n_p , p = 1 \sim NODE\_NUM\}$

B= {W', E}

W'= $\{w_q^{\,l} , q = 1 \sim REMAIN\_WORK, l = work\_level\}$

E= $\{e_q^{\,t} , t = 1 \sim APPLIED\_ROB\_NUM\}$

here, F stands for Factory configuration, consisting robot set R and map set M;
the robot set R includes species S_i, each species S_i contains several robots r_j;
the map set M includes node set N and work set W, see Figure 1 for illustration;
B is the bulletin for communication and management, involving remain work set W' and evaluation set E which is send by the robot applied for the work from bulletin, the robot with minimum evaluation of a work will get the work, i.e.,
$w_{q'} \Rightarrow r_{t'}$, $e_{q'}^{\,t'} = \min\{e_{q'}^{\,t}, t = 1 \sim APPLIED\_ROBOT\_NUM\}$, see Figure 2 for illustration.

C= $\{C(r) \mid r \in S\}$

the condition set C of robot behavior.

A= $\{A(r) \mid r \in S\}$

the action set A of robot behavior.

• Problem:

Generate the rule set as follows under the given condition of F, C and A;

$Rule_i\{C,A\}(S_i) \Rightarrow \min(W'), \quad i = 1 \sim SPECIE\_NUM$

in which $Rule_i$ is the hierarchical combination Tree of C and A, for robot species i.

From the description of the problem, we can see that:

• more than one set of rule are necessary for more than one species of robot.

• each set of rule need evaluation.

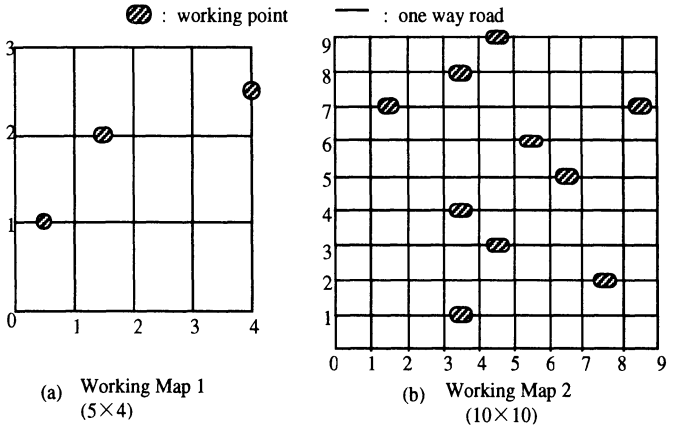


Figure 1 Examples of working maps of 5×4 and 10×10 problem.

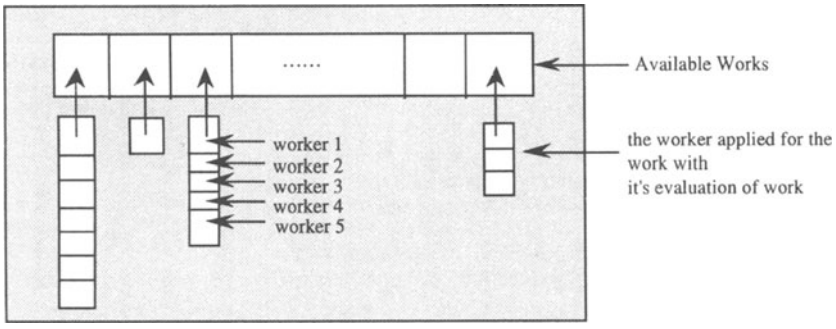


Figure 2 The manage bulletin of the working factory problem.

2 APPROACHING TO THE PROBLEM BY GP

2.1 Introduction of Problem Oriented GP

According to the before working and papers on GP, GP's weak points, the large population and generation comparing to small problem and damage of good sub tree can be understood.

Trying to reduce the calculation and keep the good sub tree of behavior, we introduce the island model[5] to the method. Showing in Figure 3, for s species robot problem, we have s population pools, each has the pool size p. Then, for the evolution of s set of rule:

- According to the island model, we divide the each population pool by island number n, then we have each pool has n islands with island population of p/n.
- After each certain generations (migration interval), certain number individuals (migration rate), will migrate to other island. The migration happens like ring, i.e., the individuals from island i migrate to i+1, while those from island n go to island 1.
- The genetic operations like crossover, mutation happen just in the range of island.

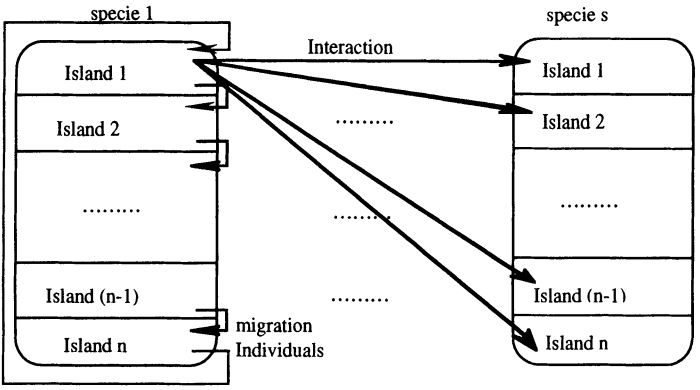


Figure 3 The island model introduced to the problem oriented GP.

2.2 The Condition Set C and Action Set A

After analyzing the problem presented in segment 1.2 and characteristics of GP, the domain dependent condition set C and action set A are decided and shown in Figure 4 in which each terms has the meaning as follows.

- *communication* makes the robot register to bulletin, evaluate the each work in bulletin, then bid the work with other registered worker.
- *working* do one work step if in fit work place with fit work ability.
- *waiting* makes the robot wait one time step on the current position.
- *random-step* randomly changes the direction of the robot, then makes the robot move one step in the new direction.
- *get-path* calculates the path between the current position and the object.
- *one-step* moves the robot one step along the gotten path or just randomly if without path.
- *back* moves the robot one step back.
- *exit* makes the robot exit from the working factory.

CONDITION

C0	IF-GET-PATH
C1	IF-REACH-DES
C2	IF-GET-COLLIDE
C3	IF-DIR-RATION
C4	IF-OUT-PATH
C5	IF-GET-WORK
C6	IF-REACH-WORK
C7	IF-WORK-FINISHED
C8	IF-FIT-WORK

ACTION

A0	communication
A1	working
A2	waiting
A3	random-step
A4	get-path
A5	one-step
A6	back
A7	exit

Figure 4 The domain dependent condition set and action set of the approach.

These conditions and actions will be used to generate the behavior rule set for the robot worker of the factory, and the behavior rule set should be evaluated according to the working result of the workers.

2.3 The Evaluation of the Behavior Rule

The evaluation of each behavior set of each species is realized according to the following steps:

- The evaluated rule set will form pair with the representatives of the island of rule set pool for other species of robot, i.e.,

  $Pair_j$:{$Rule_1$,$Rule_2$,...,$Rule_e$, ..., $Rule_i$, ..., $Rule_s$, i=1$\sim s$, e = evaluated rule set number }, j = 1$\sim s^I$, s =SPECIE_NUM, I = ISLAND_NUM.
- The robot set R will enter the factory field M with the $Pair_j$ correspondingly, i.e.,

  $S_i \Leftrightarrow Rule_i$, i = 1$\sim s$.
- Within a preset time limit T, the robots will act according to their behavior rule in the factory.
- The remaining works RW and the used time UT, consumed energy CE will be used to evaluate the $Rule_e$,

$$F'(Rule_e) = w_{all} \times (w_{time} \times UT_{all} + w_{work} \times RW_{all} + w_{energy} \times CE_{all}) + w_{species} \times (w_{time} \times UT_{species} + w_{work} \times RW_{species} + w_{energy} \times CE_{species})$$

  here, e = evaluated rule set number, *all* means the result of all robot set R and *species* mean the species evaluated. And we have

$$w_{all} + w_{species} = 1.0$$

- One Rule set can form s^I pairs with the rule set of pool for the other species, the average value

$$Fitness(Rule_e) = \sum_i F_i'/s^I, \quad i = 1 \sim s^I$$

  the maximum value

$$MaxFit(Rule_e) = \max(F_i'), \quad i = 1 \sim s^I$$

  here, we use the average value as the fitness because we believe that this can prevent the solve from convergence to rule set with too many random elements.

3 NUMERICAL EXPERIMENTS

3.1 Experiment Condition

The experiment parameters and condition are shown in Table 2.

Table 2 Experiment parameters and condition

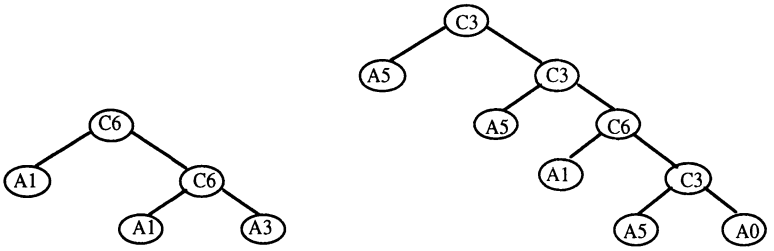
Exp	*Gen*	*Pop*	*Map*	*Time Limit*	*Island num/rate/inter*	*Remain works/steps*	*Weight all/specie*	*Worker specie/num*
1	100	100	Map1	20	10/2/5	1/3	0.2/0.8	2/2
2	100	200	Map2	60	10/2/5	4/16	0.2/0.8	2/2
3	100	200	Map2	30	10/2/5	0/0	0.0/1.0	2/6
4	100	200	Map2	60	no model	4/16	0.2/0.8	2/2

Note:

- the Map1 is shown in Figure 1, (a), the Map2 is shown in Figure 1, (b), each work contains 4 steps;
- the genetic parameters: crossover rate = 0.85, mutation rate = 0.10, copy rate = 0.05 the mutation threshold = 0.10;
- maximum tree depth is set to 10.

3.2 Experiment Results and Discussion

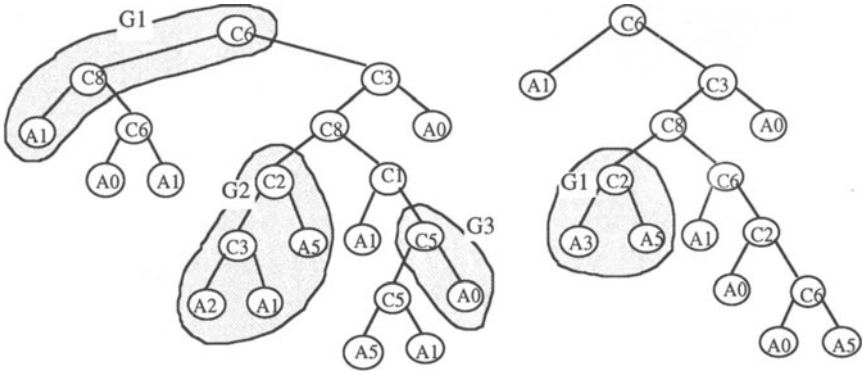
• The behavior tree changes with the environment:
when in small map containing a little works, the rule set become convergent quickly with the smallest condition and action element set involves *one step, working, if reach work*, Figure 5, (a) is the tree of the experiment 1, the random search is enough for the task (see Figure 4 for reference of condition and action meaning).



(a) The tree of experiment 1 (b) The tree of experiment 2
Figure 5 The tree of experiment 1 and 2.

When changed to the Map2 with more works and search field, the rule set extend to some set with the action element of communication, one step and other necessary element, Figure 5, (b) show the tree constructed by the condition *C3*, *C6*, action *A1*, *A0*, *A5* .

When the robot number increases, the possibility of collision and the need of coordination between the different species of robot developed, so the more complex rule set created for the adaption of the environment, see Figure 6, (a).



(a) the tree of species 1 in experiment 3 (b) the tree of species 2 in experiment 3
Figure 6 the result tree of experiment 3.

• The effective tree is usually made up with some effective sub trees which aim at the sub goals of the whole task. In some experiment, the sub tree also can be understood, for example, when we see the Figure 6, (a), the G1 is a branch for working, which can be expressed as:

IF REACH WORK
IF FIT WORK
working

and G3 is a branch for getting work and walking to work position, which also can be expressed like

```
IF NOT FIT WORK
        IF REACH DES
                working
        ELSE
                IF GET WORK
                        one-step
                ELSE
                        communication
while G2 coordinates with Figure's (b) G1 when collide with each other
species 1's G2                                  species 2's G1
    IF GET COLLIDE                              IF GET COLLIDE
        IF GET DIR RATION                           waiting
            waiting                             ELSE
        ELSE                                        random-step
            working
    ELSE
        one-step
```

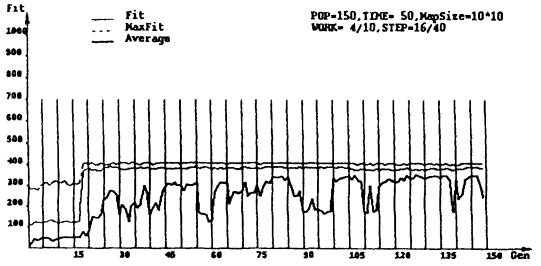
- Comparison between on island model and without island model

Figure 7 is the fitness comparison between experiment 4(Figure 7, a) and experiment 2(Figure 7, b), here Fitness = max($Fitness_i$), $i = 1 \sim POPULATION\_NUM$

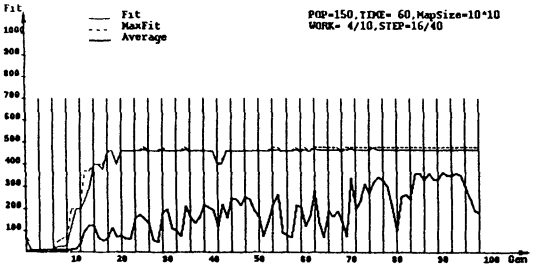
$MaxFit = max(MaxFit_i), i = 1 \sim POPULATION\_NUM$

$Average = \sum_i (Fitness_i), i = 1 \sim POPULATION\_NUM$

$Fitness_i$ and $MaxFit_i$ have been explained in segment 2.4. In Figure 7, each migrating interval, there is a vertical line to show the migration happening. We can see that, with ordinary coevolution method, the solve can tend to convergence steadily, while on island model, the migration lead to the change to the spatial structure, and make the fitness curve especially average curve variating intensely, but not damaging the up-tendency of the curve.



(a) fitness curve of experiment 4



(b) fitness curve of experiment 2

Figure 7 Comparison between the fitness curves of experiment 4 and 2.

4 CONCLUSION

This paper has described the scheme using GP to generate and evolve the hierarchical behavior structure for the automatic working factory problem.
We can see that, when using distributed planning and control policy instead of traditional planning and control policy, the practical problem can be solved with flexibility on high level, i.e., be solved through the cooperation and adaption of the individuals.
The behavior rule set generated and evolved by GP can well fit the environment, makes the robot individual present some effective performance in the working environment. It seems that the GP can be used to explore more complex behavior structure of practical problem.
Using the Island Model leads to the reduction of the calculation, while not damaging the up-tendency of the fitness curves.

5 REFERENCES

[1]Jonalthan H. Connell, Sridhar Mahadevan et al (1993). Robot Learning. Kluwer Academic Publishers.
[2]Branco Soucek and the IRIS Group (1992). Dynamic, Genetic and Chaotic Programming, the Sixth Generation. A Wiley-Interscience Publication.
[3]Lydia Kronsjo, Dean Shumsheruddin (1991). Advances in Parallel Algorithms. Black Scientific Publications.
[4] Gregory J.E. Rawlins el al (1991). Foundation of Genetic Algorithms. Morgan Kaufmann Publishers.
[5] Lawrence Davis et al (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold.