



HOKKAIDO UNIVERSITY

Title	Design Method of Pheromone Style Communication for Coordinated Agents Systems
Author(s)	Kawamura, Hidenori
Degree Grantor	北海道大学
Degree Name	博士(工学)
Dissertation Number	甲第5118号
Issue Date	2000-03-24
DOI	https://doi.org/10.11501/3168682
Doc URL	https://hdl.handle.net/2115/51643
Type	doctoral thesis
File Information	000000353875.pdf



Design Method of
Pheromone Style Communication
for Coordinated Agents Systems

Hidenori KAWAMURA

Submitted in partial fulfillment of the
requirements for the Degree of Doctor of
Engineering

Hokkaido University

December 1999

①

Contents

Design Method of Pheromone Style Communication for Coordinated Agents Systems

Hidenori KAWAMURA

Submitted in partial fulfillment of the
requirements for the Degree of Doctor of
Engineering

Hokkaido University

December 1999

Contents

1	Multi-Agent System and Ant Colony System	12
1.1	Introduction	12
1.2	Multi-Agent System	13
1.2.1	Problem Solving from Nature	14
1.2.2	Recent Works and Issues on Multi-Agent Systems	17
1.2.2.1	The Tileworld	19
1.2.2.2	The Pursuit Problem	20
1.2.2.3	The Prisoner's Dilemma	21
1.2.3	Formalization of Agent System and Communication	23
1.2.3.1	Direct Communication	25
1.2.3.2	Indirect Communication	27
1.3	Ant Colony System and Multi-Agent System	29
1.3.1	The Origin of Coordinated Behavior on Ant Colony System	29
1.3.2	Altruistic Behaviors and Total Group Rationality .	33
1.3.3	Pheromone Communication on Ant Colony System	33
1.3.4	Pheromone Style Communication on Multi-Agent System	38
1.4	The Survey of Related Works about Ant-like Agents and Pheromone Style Communication	41
1.4.1	The Simulation Models of Ant-like Agents and Pheromone Style Communication	41
1.4.2	Ant Colony Optimization	44
1.4.2.1	Similarities and differences with real ants	45
1.4.2.2	The ACO meta-heuristic	46
1.4.2.3	Applications of ACO Algorithms	50
1.4.2.4	Ant Colony System for Traveling Sales- man Problems	52
1.5	Conclusions	55

2	Multiple Ant Colonies Algorithm for Traveling Salesman Problems	62
2.1	Introduction	62
2.2	Multiple Ant Colonies Algorithm	65
2.2.1	The Formalizations of Multiple Ant Colonies Algorithm	65
2.2.2	The Details of Multiple Ant Colonies Algorithm	66
2.3	Computer Experiments	70
2.3.1	Experiment 1	71
2.3.2	Experiment 2	75
2.3.3	Experiment 3	77
2.4	Discussion	80
2.5	Conclusions	82
3	Distributed Optimization Algorithm Based on Pheromone Style Communication for Vehicle Routing Problem	87
3.1	Introduction	87
3.2	Vehicle Routing Problem	88
3.3	Distributed Scheduling Algorithm Based on Multi-agents with Pheromone Style Communication	90
3.3.1	The Operations of Agents	92
3.3.2	The Management of Pheromone Information on the Environment	96
3.4	Computer Experiments	97
3.5	Conclusions	98
4	Ants War with Pheromone Style Communication	106
4.1	Introduction	106
4.2	Ants War	107
4.3	The Formalization of Ant Agents and Pheromone Communication	111
4.4	The Ant Agent with Continuous Pheromone Sensor	112
4.4.1	The Agent Architecture	112
4.4.2	Evolutionary Strategy	113
4.4.3	Experimental Results	116
4.5	The Ant Agent with Descrete Pheromone Sensor	122
4.5.1	The Agent Architecture	122
4.5.2	Computer Experiments	124
4.5.2.1	Result 1 about Attractive Pheromone	124
4.5.2.2	Result 2 about Repulsive Pheromone	143

4.5.3	Discussion	159
4.6	Conclusions	161
5	Summary	164

List of Figures

1.1	The impact of the environment on the development of the individual	24
1.2	The relationship between the individual and the environment	25
1.3	An example of a system	26
1.4	An example of a system	26
1.5	The concept of a system	27
1.6	The concept of a system	28
1.7	The concept of a system	29
1.8	The concept of a system	30
1.9	The concept of a system	31
1.10	The concept of a system	32
1.11	The concept of a system	33
1.12	The concept of a system	34
1.13	The concept of a system	35
1.14	The concept of a system	36
1.15	Experimental setting by Dehaene	37
1.16	Results for a typical single case by Dehaene	38
1.17	The range of phenomena which are covered by the concept	39
1.18	The environment and the individual	40
1.19	The concept of a system	41
1.20	The concept of a system	42
1.21	The concept of a system	43
1.22	The concept of a system	44
1.23	The concept of a system	45
1.24	The concept of a system	46
1.25	The concept of a system	47
1.26	The concept of a system	48
1.27	The concept of a system	49
1.28	The concept of a system	50
1.29	The concept of a system	51
1.30	The concept of a system	52
1.31	The concept of a system	53
1.32	The concept of a system	54
1.33	The concept of a system	55
1.34	The concept of a system	56
1.35	The concept of a system	57
1.36	The concept of a system	58
1.37	The concept of a system	59
1.38	The concept of a system	60
1.39	The concept of a system	61
1.40	The concept of a system	62
1.41	The concept of a system	63
1.42	The concept of a system	64
1.43	The concept of a system	65
1.44	The concept of a system	66
1.45	The concept of a system	67
1.46	The concept of a system	68
1.47	The concept of a system	69
1.48	The concept of a system	70
1.49	The concept of a system	71
1.50	The concept of a system	72
1.51	The concept of a system	73
1.52	The concept of a system	74
1.53	The concept of a system	75
1.54	The concept of a system	76
1.55	The concept of a system	77
1.56	The concept of a system	78
1.57	The concept of a system	79
1.58	The concept of a system	80
1.59	The concept of a system	81
1.60	The concept of a system	82
1.61	The concept of a system	83
1.62	The concept of a system	84
1.63	The concept of a system	85
1.64	The concept of a system	86
1.65	The concept of a system	87
1.66	The concept of a system	88
1.67	The concept of a system	89
1.68	The concept of a system	90
1.69	The concept of a system	91
1.70	The concept of a system	92
1.71	The concept of a system	93
1.72	The concept of a system	94
1.73	The concept of a system	95
1.74	The concept of a system	96
1.75	The concept of a system	97
1.76	The concept of a system	98
1.77	The concept of a system	99
1.78	The concept of a system	100
1.79	The concept of a system	101
1.80	The concept of a system	102
1.81	The concept of a system	103
1.82	The concept of a system	104
1.83	The concept of a system	105
1.84	The concept of a system	106
1.85	The concept of a system	107
1.86	The concept of a system	108
1.87	The concept of a system	109
1.88	The concept of a system	110
1.89	The concept of a system	111
1.90	The concept of a system	112
1.91	The concept of a system	113
1.92	The concept of a system	114
1.93	The concept of a system	115
1.94	The concept of a system	116
1.95	The concept of a system	117
1.96	The concept of a system	118
1.97	The concept of a system	119
1.98	The concept of a system	120
1.99	The concept of a system	121
1.100	The concept of a system	122

List of Figures

1.1	The approach from nature to artificial intelligence and artificial life.	14
1.2	The relationship between artificial systems and real life systems.	15
1.3	An example of Tileworld.	19
1.4	An example of Pursuit Problem.	20
1.5	The Prisoner's Dilemma.	22
1.6	The image of relationship between an agent system and the environment.	24
1.7	The image of direct communication among agents.	25
1.8	The examples of direct communication.	27
1.9	The image of indirect communication among agents based on communication space.	28
1.10	The mechanism of crossover on real ant colonies.	30
1.11	The kin degrees between parents and children on real ant colony.	31
1.12	The reaction of real ant according to sensing the pheromone trail.	34
1.13	The image of whole behavior of foraging ants.	35
1.14	The image of whole behavior of foraging ants when the obstacle suddenly appears.	36
1.15	Experimental Setting by Deneubourg.	37
1.16	Results for a typical single trail by Deneubourg.	37
1.17	The image of pheromone style communication among agents. 39	
1.18	The environment and internals of ants in AntFarm.	41
1.19	The ACO meta-heuristic in pseudo-code.	49
2.1	An image of the multiple ant colonies algorithm for a TSP with four colonies.	64
2.2	An image of moving ant agent on pheromone trail.	67
2.3	The effectiveness of parameter α	70

2.4	Relationship between ant agents, colonies and problem domain with control parameters.	71
2.5	The algorithm of the multiple ant colonies.	72
2.6	The colony structure used in the experiment 1.	73
2.7	The transition of the pheromone and the best solutions obtained in experiment 1.	74
2.8	The colony structure used in the experiment 2.	76
2.9	A network colony structure with six colonies. $\alpha(h, h) = 1.5$. The values of $\alpha(h, l)$ not shown were all set to zero.	77
2.10	The time to obtain the best solution and the dimension of solved TSP.	79
2.11	Examples of best solutions obtained by each colony in experiment 3 (1).	83
3.1	The example of a solution for VRP including three vehicle routes.	89
3.2	The example of agent's partial solutions for the Vehicle Routing Problem.	90
3.3	The image of meanings of pheromones in the algorithm.	91
3.4	The multi-agent system and the architecture of agents for distributed scheduling.	93
3.5	The example flow of operations generating a new partial schedule.	95
3.6	The example of agent's partial solutions for the Vehicle Routing Problem.	96
3.7	The changes of agent routes and the best solution obtained through the experiment.	99
3.8	The change of intensity of pheromone.	100
3.9	The change of evaluation values.	100
3.10	The change of TE^k	101
4.1	Ants war world and architecture of the ant agent.	108
4.2	The image of evaporation and diffusion of pheromone.	108
4.3	A series of GA operations in a GA time.	113
4.4	The crossover and mutation operations used in the GA.	114
4.5	The graph representing the ratios to win against the criterion final population.	115
4.6	The graph representing the average ratios to win against the criterion in 7 trials.	116
4.7	The screen shots in the ants war of type 1 agents.	118

4.8	The screen shots in the ants war of type 2 agents.	119
4.9	The screen shots in the ants war of type3 agents.	120
4.10	The inputs-actions bitmap.	121
4.11	The ratio to fire the sensor neuron according to the input value of pheromone.	123
4.12	The screen shots of the ants war in experiment 2-1(a).	126
4.13	The screen shots of the ants war in experiment 2-1(b).	127
4.14	The ratio to win against the final population in the experiment 2-1.	130
4.15	The transitions of $H(\text{Output})$, $I(\text{Output}, \text{Sensor})$, and $I(\text{Output}, \text{Pheromone})$ in experiment 2-1.	130
4.16	The transition of the ratio to put the pheromone in the experiment 2-1.	131
4.17	The ratio to win against the ant agents which were forbidden to put the pheromone in the experiment 2-1.	131
4.18	The weights from input $i_1 - i_3$ to outputs in the experiment 2-1 (A).	134
4.19	The weights from input $i_4 - i_6$ to outputs in the experiment 2-1 (B).	135
4.20	The weights from input $i_7 - i_9$ to outputs in the experiment 2-1 (C).	136
4.21	The weights from input $i_{10} - i_{11}$ to outputs in the experiment 2-1.	137
4.22	The transitions of intensity of pheromone in experiment 2-1 (A).	138
4.23	The transitions of intensity of pheromone in experiment 2-1 (B).	139
4.24	The contour line about the intensity of pheromone in the experiment 2-1.	140
4.25	The contour line about the intensity of pheromone indicated by log scale in the experiment 2-1.	141
4.26	The transitions of ratio of forcing ants and the number of food items in the experiment 2-1.	142
4.27	The screen shots of the ants war in experiment 2-2.	144
4.28	The ratio to win against the final population in the experiment 2-2.	146
4.29	The transitions of $H(\text{Output})$, $I(\text{Output}, \text{Sensor})$, and $I(\text{Output}, \text{Pheromone})$ in experiment 2-2.	146

4.30	The transition of the ratio to put the pheromone in the experiment 2-2.	147
4.31	The ratio to win against the ant agents which were forbidden to put the pheromone in the experiment 2-2.	147
4.32	The weights from input $i_1 - i_3$ to outputs in the experiment 2-2 (A).	149
4.33	The weights from input $i_4 - i_6$ to outputs in the experiment 2-2 (B).	150
4.34	The weights from input $i_7 - i_9$ to outputs in the experiment 2-2 (C).	151
4.35	The weights from input $i_{10} - i_{11}$ to outputs in the experiment 2-2.	152
4.36	The transitions of intensity of pheromone in experiment 2-2 (A).	153
4.37	The transitions of intensity of pheromone in experiment 2-2 (B).	154
4.38	The transitions of intensity of pheromone in experiment 2-2 (C).	155
4.39	The contour line about the intensity of pheromone in the experiment 2-2.	156
4.40	The contour line about the intensity of pheromone indicated by log scale in the experiment 2-2.	157
4.41	The transitions of ratio of forcing ants and the number of food items in the experiment 2-2.	158
4.42	The ratio to win in various number of agents.	159

List of Tables

1.1	List of applications of ACO algorithms to static combinatorial optimization problems.	51
1.2	List of applications of ACO algorithms to dynamic combinatorial optimization problems.	51
1.3	Comparison of ACS with the other approximate optimization algorithms.	54
1.4	ACS performance for some bigger TSPs.	54
2.1	Values of parameters used in experiment 1.	73
2.2	Values of parameters used in experiment 2.	75
2.3	Results of experiment 2.	75
2.4	Results of experiment 3 (1) using a network colony structure.	80
2.5	Results of experiment 3 (2) using a network colony structure based on the positive pheromone effects.	80
2.6	Results of experiment 3 (3) using a network colony structure based on no effects.	80
2.7	Results for other TSPs.	81
3.1	The values of parameters.	98
4.1	The Action and action number on ants war.	133

Preface

In this doctoral dissertation, multi-agent systems are focused as distributed problem solving systems. In these systems, many agents as problem solvers autonomously and adaptively effect their actions to solve given tasks. Each agent is designed by a designer based on bottom-up approach, where, it is expected for the whole of agents to emerge self-organized behavior to solve their complex tasks exceed the sum of individual abilities. For achievement of such behavior, agents are needed to autonomously make own decisions by using of interactions between others, that is, communications. To design such system is generally said to be difficult.

Real ant colonies are noticed here as an excellent archetype system based on flexible self-organized behaviors. Real ant is called a *social insect*, and a colony of real ants is known to have surprising high abilities although each individual is based on simple action pattern. In such colony, working labors are distributed to cope various tasks for their lives, e.g., forage food items, keep their children, attack enemies, develop their nest, and so on, and distributed ants play various roles in adaptive organizations. The objective of designing multi-agent systems is to achieve like a colony of ants, thus, the analogy of real ant colonies is worthwhile.

In ant colonies, various kinds of communication means are used for organizations having high abilities. Especially, communications via chemical means, that is, *pheromone communication*, play important roles for keeping organization abilities. Owing to pheromone, ants can communicate distinct means to others according to kinds of pheromone, moreover, ants can be affected by whole behaviors without sensing whole behaviors, because pheromone space is formed on the environment by a set of individual behaviors. No conventional communication mean have this characteristics, and the analogy of pheromone communication is very useful to design adaptive multi-agent systems.

Therefore, this dissertation is related to a study on design method of pheromone style communication for coordinated agents systems. Especially, multi-agent systems based on pheromone style communication are

applied to combinatorial optimization problems, namely, Traveling Salesman Problems and Vehicle Routing Problems. The experimental results show the effectiveness of these applications. Moreover, the multi-agent system with evolutive pheromone style communication based on the Genetic Algorithm is proposed in the ants war game world. The hybrid of pheromone style communication and genetic algorithm enable agents to autonomously construct their communication according to their objective without a detail drawing of agent communication by a designer. The abstract of contained sections are as follows.

Chapter 1 is related to multi-agent systems and real ant colonies. Communication means of multi-agent systems and some issues are surveyed. Real ant colonies are discussed in view points of multi-agent systems and total group rationality of agents. The design concept of pheromone style communication is described, and some works are surveyed based on proposed design concept of pheromone style communication.

In chapter 2, I focus on ant algorithm, which Colorni et al. proposed in 1991 to solve Traveling Salesman Problems, as a start point of this study. It is cleared that ant algorithm has some defect that it can not do variety search. To overcome this weakness, multiple ant colonies are introduced to ant algorithm, and multiple ant colonies algorithm is proposed with pheromone style communications based on colony level interaction. In addition, the computer experimental results that multiple ant colonies algorithm is superior to the original ant algorithm are showed.

In chapter 3, the distributed search algorithm based on pheromone style communication is proposed to solve Vehicle Routing Problems, which is known as harder problems than Traveling Salesman Problems because it is necessary to determine multi vehicle routes at once with minimizing an objective function. Pheromone style communication is used to solve competitions between agents and harmonize behaviors of agents according to their objective. Some computer experimental results shows the effectiveness of this proposed multi-agent system.

In chapter 4, the multi-agent system with evolutive pheromone style communication based on the Genetic Algorithm is proposed in the ants war game world. In ants war, two colonies of ant agents scramble limited food resource by using of pheromone style communication, and this game is considered as the test problem in which agents are required adaptive resource allocation, adaptability to dynamic environment and planning ability under limited sensed information. An agent in the multi-agent system is constructed with an artificial neural network, and the genetic

algorithm evolves pheromone style communication. The effectiveness of pheromone style communication is analyzed based on some evaluations according to experimental results.

In chapter 5, the summarize of this study is described.

Chapter 1

Multi-Agent System and Ant Colony System

1-1 Introduction

Recently, multi-agent system (MAS) has been attracting much attention. MAS is a system in which multiple autonomous agents interact with each other to solve a problem. The agents are represented by software or hardware entities that can communicate with each other and perform actions in the environment. MAS has been applied to various fields such as artificial intelligence, robotics, and distributed systems. In this chapter, we will introduce the basic concepts of MAS and Ant Colony System (ACS).

Multi-Agent System (MAS) is a system in which multiple autonomous agents interact with each other to solve a problem. The agents are represented by software or hardware entities that can communicate with each other and perform actions in the environment. MAS has been applied to various fields such as artificial intelligence, robotics, and distributed systems. In this chapter, we will introduce the basic concepts of MAS and Ant Colony System (ACS).

Chapter 1

Multi-Agent System and Ant Colony System

1.1 Introduction

Recently, multi-agent systems gather attentions as a new computing paradigm to solve complex problems which are difficult to be solved by conventional artificial intelligence techniques based on the top-down approach, in which systems are constructed with hierarchized and distributed organizations according to intention of designers, and the centralized supervisor component of system manages whole behavior of system. Multi-agent systems consist of many agents as distributed problem solvers, which autonomously and adaptively effect their actions to solve given tasks. Agents are usually designed with adaptability and learnability to adapt the complex environment concerning with solved problems. In such systems, many agents with various relationship between others interact with each other. It is expected that these interactions enable multi-agent systems to emerge organizations with abilities to achieve flexible and autonomous problem solving against the dynamic environment. That is, self-organizations of multi-agent systems consisting of such autonomous agents must be exceeded a mere sum of components of systems.

However, it is difficult to design intelligent agents and multi-agent systems, of course, including designing of relationships and interactions among agents. We can find archetypes of an excellent agent in various levels of real lives, e.g., genes, cells, organizations, individuals, swarms, societies and cultures, and the analogies of those have caused many artificial intelligence technologies.

Especially, real ant colonies are noticed here as an excellent archetype system based on flexible self-organized behaviors. A real ant is called a

social insect, and a colony of real ants is known to have surprising high abilities although each individual is only based on simple action pattern. In such colony, working labors are distributed to cope various tasks for their lives, e.g., forage food items, keep their children, attack enemies, develop their nest, and so on, and distributed ants play various roles in adaptive organizations. The objective of designing multi-agent systems can be said to achieve a system like a colony of ants, thus, the analogy of real ant colonies is valid.

In ant colonies, various kinds of communication means are used to emerge organizations having high abilities. Especially, communications via chemical means, that is, pheromone communications, play important roles for keeping organization abilities. Owing to characteristics of pheromone, ants can communicate distinct means to others according to kinds of pheromone, moreover, ants can be affected by whole behaviors without sensing whole behaviors, because pheromone space is formed on the environment by a set of individual behaviors. No conventional communication mean has this characteristics, and the analogy of pheromone communication is very useful to design adaptive multi-agent systems.

In this section, recent works and issues of multi-agent systems will be described, and formalizations of multi-agent systems in the view point of communication means will be represented. On the other hand, a real ant colony as an archetype of multi-agent system will be introduced, and some example of excellent organizations will be shown. Thus, the multi-agent system based on pheromone style communication will be formalized, and several related works will be introduced.

1.2 Multi-Agent System

This section is concerning with the survey of multi-agent systems and brief introduction of artificial intelligence implemented in it. In multi-agent systems, artificial intelligence technologies are very important because according to solved tasks it is difficult to manage organizations consisting of many agents. Many artificial intelligence techniques were proposed by simulating the mechanisms of real lives. On the other hand, three important problems, the pursuit problem, the tailworld, and the prisoner's dilemma are introduced in this section as the test problems of multi-agent systems. These problems are deeply related to essential issues caused in multi-agent systems. Moreover, the formalization of multi-agent systems are described in the viewpoint of communication

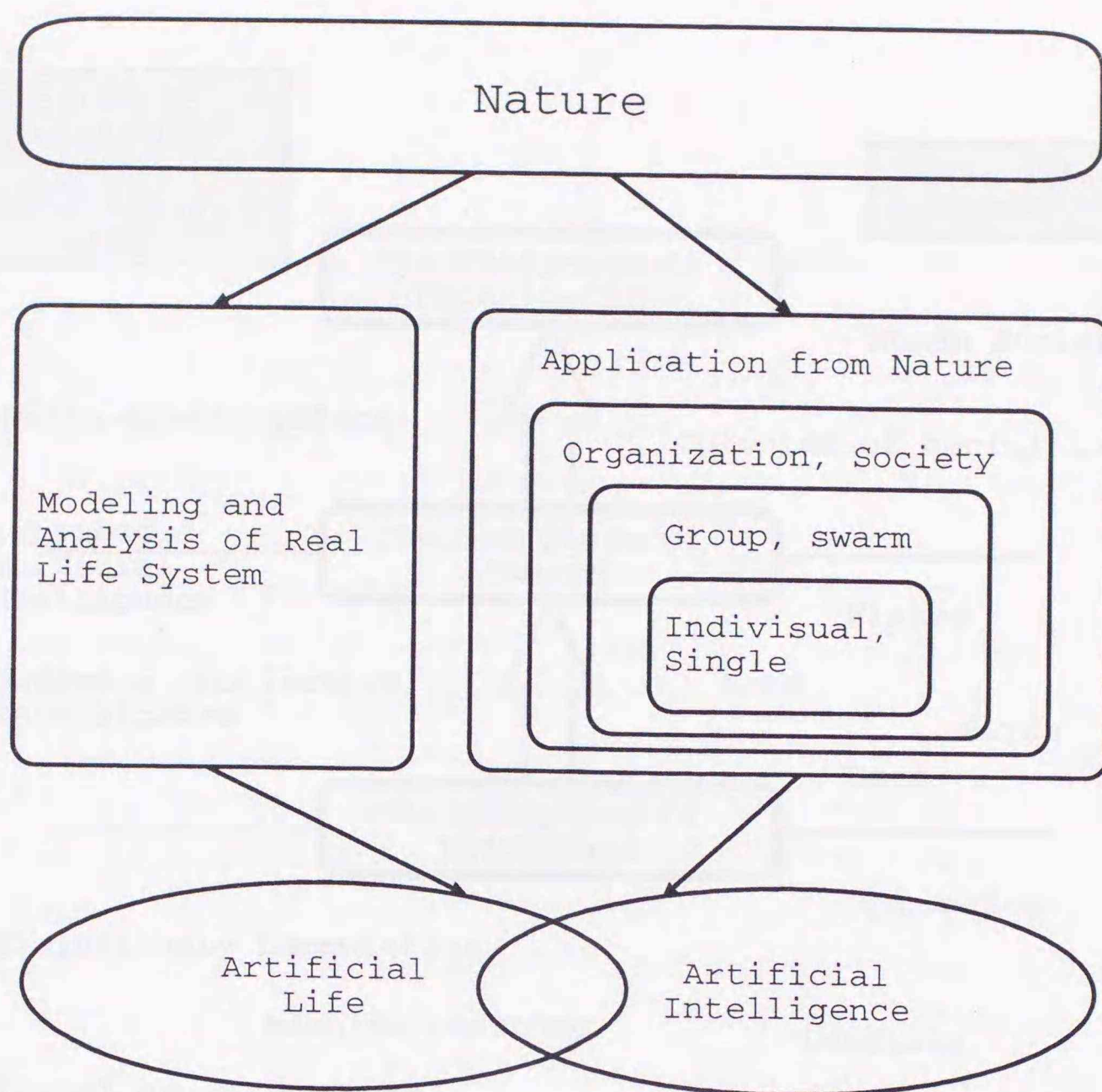


Figure 1.1: The approach from nature to artificial intelligence and artificial life.

forms. In the formalization, two communication means, namely, direct communication and indirect communication are formalized.

1.2.1 Problem Solving from Nature

New artificial intelligent systems are expected to solve many various and complex problems which are difficult to be solved by old fashioned artificial intelligence technologies. Generally, a solved problem is on the dynamic environment which is based on many complicated interactions between elements of environment, and it is difficult for designers to construct well problem solving systems according to conventional top-down approaches because we cannot predict all behaviors and all results caused in dynamic relations in advance. Thus, it is necessary for designing of problem solving systems to make such systems have abilities of flexibility, adaptability, learnability, and so on, like real life systems. In fact, there are many adaptive real systems and many levels of organizations in nature, so it is very worthful to emulate systems of nature.

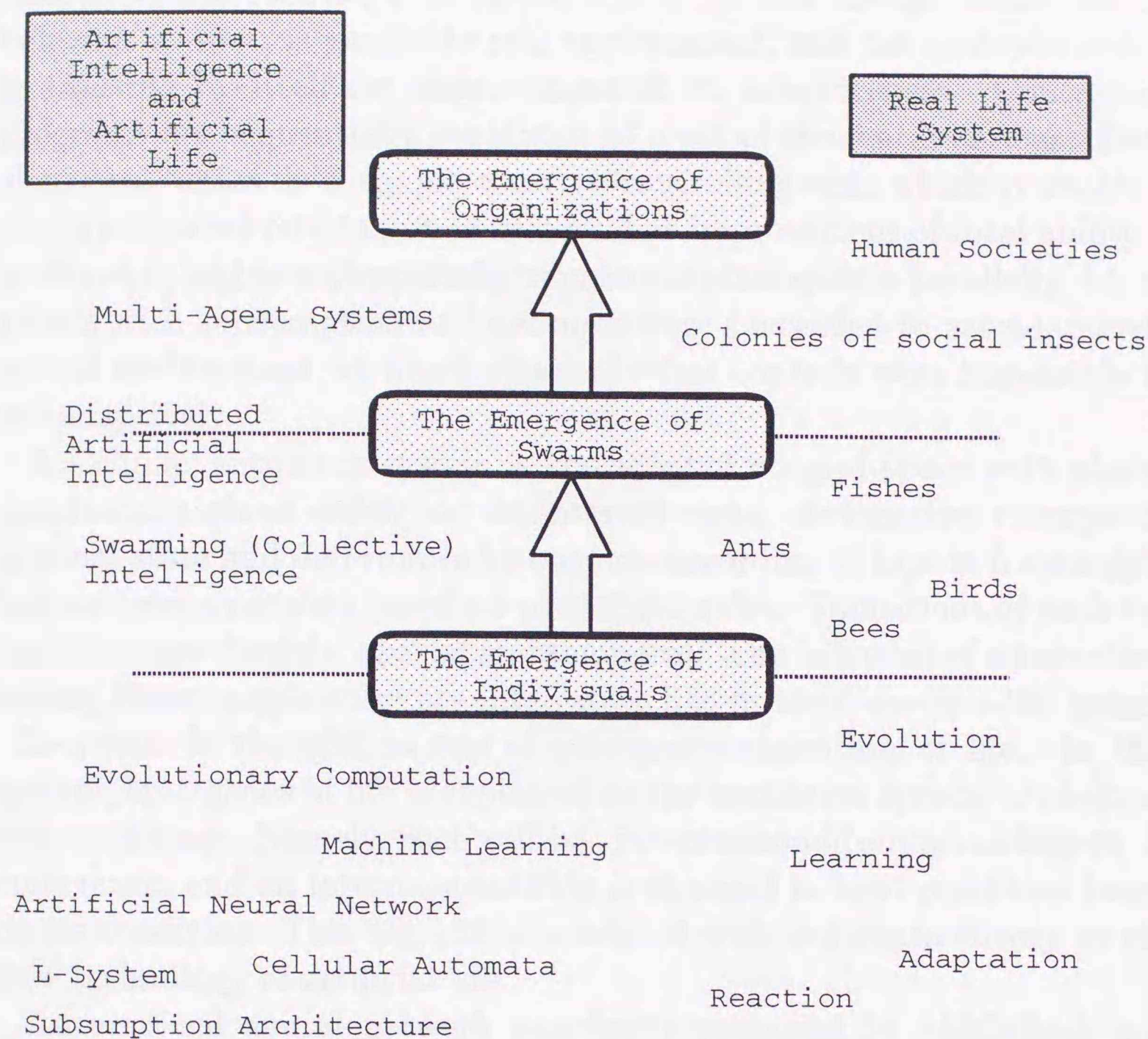


Figure 1.2: The relationship between artificial systems and real life systems.

Fig. 1.1 shows the approach from nature to artificial intelligence and artificial life [44][20]. The research area of artificial life mainly focus on essences of real life systems through modeling of real life systems and analyzing the behavior of models. On the other hand, various levels of real life systems are focused on for artificial intelligence, and abilities in each level are studied as archetypes of adaptive systems, e.g., the levels of individual abilities, swarm abilities, organization abilities, society abilities, and so forth.

In Fig. 1.2, the relationship between artificial systems and real life systems is shown according to levels of organizations. In the emergence level of individuals, some kinds of technologies for individual abilities are widely studied on, and these technologies are introduced here briefly.

Subsumption architecture was proposed by L. Brooks for controlling

intelligent real robots [5]. A robot, which has not enough ability for its behavior at first, is grazed to real environment, and the ability of robot is adjusted by itself for improvement of its adaptiveness. Controlling programs are sequentially generated as a set of elemental actions. Each elemental action is a simple reactive controlling unit which connects a sensory input of robot to an actuator. The improvement of total ability is achieved to add new elemental action to a control system parallelly. Many robots with subsumption architectures were succeeded to act adaptively in real environment, in which classical robot controls were impossible to act adaptively.

A Cellular automata system is constructed on grid space with placed automata, each of which has its internal state. Automaton changes its internal state and determines its outputs according to inputs from neighbor automata outputs based on predefined rules. Transitions of each cell condition are discrete, spatial and temporal. The behavior of interactions among these simple rules are interesting like in the Conway's life game.

L-system is thought as one of emergence algorithm of life. In this system, emergence of life is explained as the transition system of abstract cell conditions. Namely, each cell has internal condition according to its emergence, and an internal condition is changed to next condition based on its condition. This algorithm is related with automata theory as the core technology of artificial life.

An artificial neural network was firstly proposed by McCulloch and Pitts by means of modeling a biological neural network. In an artificial neural network, there are many neurons, each of which has inputs from others and single output to others. A neuron is connected to others with weighted arcs, and an output signal is decided by the function of sum of weighted input signals. A neural network can learn many kinds of functions by adjusting of weights by learning methods, e.g., back propagation method.

There are some kinds of evolutionary computation such as genetic algorithm, genetic programming, classifier system, and so on. These paradigms basically consist of a set of solutions or rules which are concerning to given problems, and a solution or rule is generally called a "chromosome". The objective of these paradigms is to find a best chromosome of many possible ones. At first, several chromosomes are randomly generated in genetic pool, and each chromosome is evaluated by a specific evaluation function which measures a goodness of chromosome related with the characteristics of solved problem. According to evalua-

tion values, some chromosomes having high evaluations are stochastically selected as parents, and these chromosomes generate new chromosomes as children with genetic operators, namely, crossover and mutation. Oppositely, chromosomes with bad evaluations tend to be disappeared from the genetic pool. This process is repeated until a stop criterion comes, and final chromosomes in genetic pool are probably better solutions of problems. Especially, genetic programming uses lisp programs as chromosomes, and classifier systems use input-output action rules of an agent.

Those technologies for individual abilities base on higher level intelligence. As the emergence model of swarms, the boid was proposed by C. Reynolds [40]. The boid is a simulation program which represents the swarm behavior of bird flying. Each bird has very simple distributed flying rules not affected whole swarm behaviors, e.g., the rule to be apart of too close mates for avoidance of collision, to adjust its speed to other mates' speed, and to tend going the center of swarm. It is known that the boid emerge excellent group behavior closely like real birds. However, this algorithm was proposed to simulate real bird behaviors, and each bird behave its opinion only. To construct group level intelligence, the group interactions based on total objectives of systems must be considered.

In the emergence level of organizations, many agents of systems learn from adaptive actions to the environment, cooperative and competitive relationships between others, evolve from results of behaviors, and exchange of knowledge between others. The total group of agents self-organizes for their objectives with balancing cooperation and competition based on individual abilities. In these systems, the objectives of systems can be performed by group behavior caused by behaviors of individual agents. Thus, multi-agent approach must construct effective and flexible problem solving systems which can not be made by means of old fashioned artificial intelligence approaches.

1.2.2 Recent Works and Issues on Multi-Agent Systems

It is expected for the research area of distributed artificial intelligence to construct intelligent problem solvers for various and complex problems including dynamic environment. As examples of such problems, we can pick up distributed sensing problems, management problems of network, cooperative robots, distributed scheduling, and so on. One kind of distributed artificial intelligence technologies, multi-agent systems are got attention as flexible problem solving systems, in which many components of systems called "agents" work together for solving given problems.

In multi-agent systems, many agents designed based on bottom-up approach interact with each other to coordinate for solving given tasks and communicate to exchange knowledge regarding with problem domain including themselves. Organizations of agents, which are emerged by a large number of interactions, behave with divided roles of agents, and it is expected that the total group behavior of agents exceeds a mere sum of individuals for solving given complex tasks.

To design intelligent multi-agent systems, namely, to design intelligent agents, it is difficult to predict how agent design is good to construct intelligent multi-agent systems because multi-agent systems are generally established on many interactions of agents, moreover, the environment in that agents exist is difficult to be known previously while designing. There is another important problem what evaluation an agent should independently conform in the situation that each agent can not perceive the total behavior and the evaluation of such behavior. Moreover, agents must appropriately allocate their resource, i.e., agents, to deal with various tasks and sub-tasks which is set to achieve global tasks of agents.

Therefore, agents are required to have ability by which agents can autonomously adapt their dynamic environment and construct proper relationships between other agents for their sake. Agents should be implemented with some abilities like adapting, learning or evolving, and it is needed for a designer to construct agents implemented such abilities. According to the above, following three items are noted as basic and important subjects on the research area of multi-agent systems[38].

1. Agent Architecture

The environment seems to dynamically change for each agent because many other agents independently behave to perform their tasks. What agent architecture is well designed in the condition that it is difficult for an agent to predict changes of the environment including other agents?

2. Coordinated Organization Scheme

Generally, agents in multi-agent systems exist to coordinately perform their tasks, which are given to the total of agents. Agents are required to realize own roles for efficient achievement of their tasks. Thus, what schema is needed to have agents decide own policies and own sub-tasks with coordinated behaviors?

3. Agent Policy in Un-cooperated Relation

The relations between agents are roughly classified into two rela-

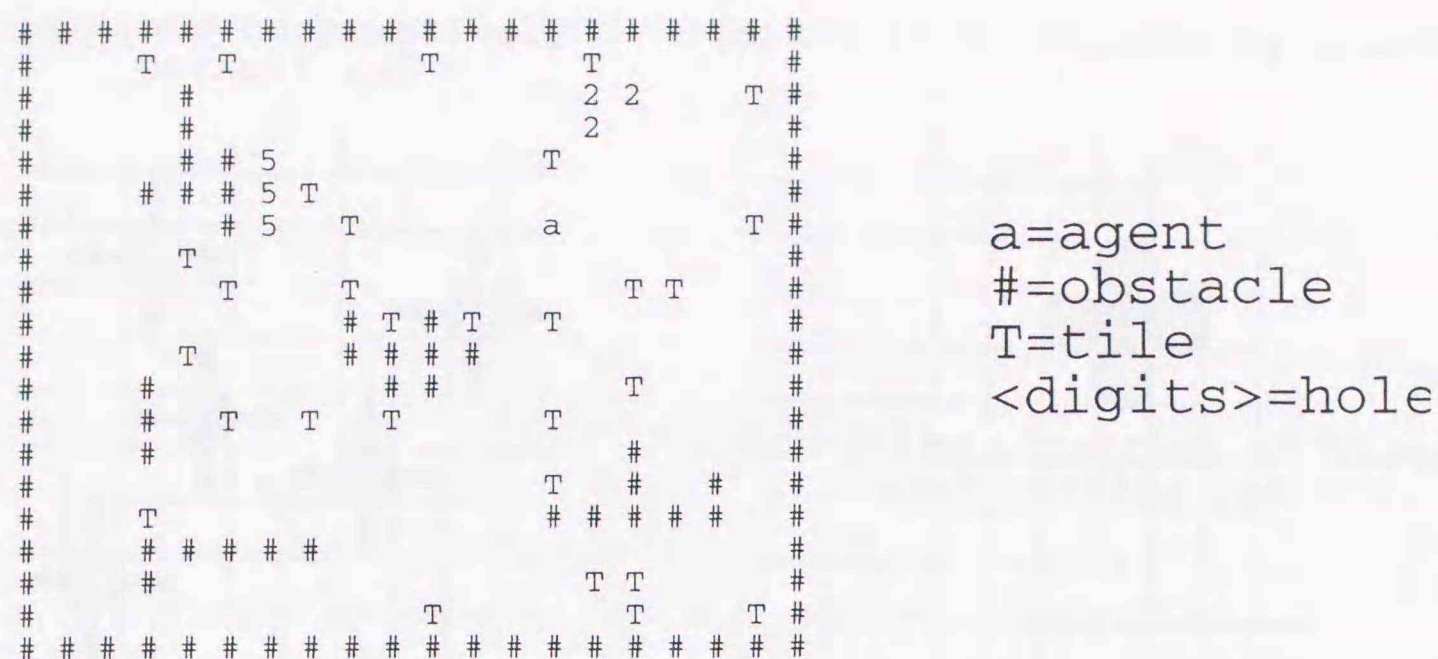


Figure 1.3: An example of Tileworld.

tions, that is, cooperated relation and un-cooperated relation. It is said that cooperated relation is based on total group rationality and un-cooperated relation is based on individual rationality. Moreover, un-cooperated relation is divided into two relations, competitive relation and un-competitive relation. In competitive relation, agents are opposed each other to maximize own utilities. In these un-coordinated relation, how do agents make a decision to maximize own utilities?, and what results are brought to agents according to the process of total behavior consisting of such agents? [37]

1.2.2.1 The Tileworld

Concerning with the first of the above, it is one of important issue for agents how to achieve from a starting situation to an ending situation through transitions of situations according to lots of decided actions of agents. This problem is generally called a planning problem. In classical plannings, an agent previously generates and evaluates possible plans with numerous computation costs, and selects best one of candidate plans. Blocks world is used as a test of planning problem on the assumption of applying to robot decision-making. The basic planning methods are based on forward planning and backward planning. forward planning applies all transition rules to situations from the starting one until a goal situation is obtained. This planning requires to enumerate all possible situations because applied transition rules are a lot. On the other, backward planning starts from a hoped goal situation to backward situations reversely compared with a forward planning, so backward planning

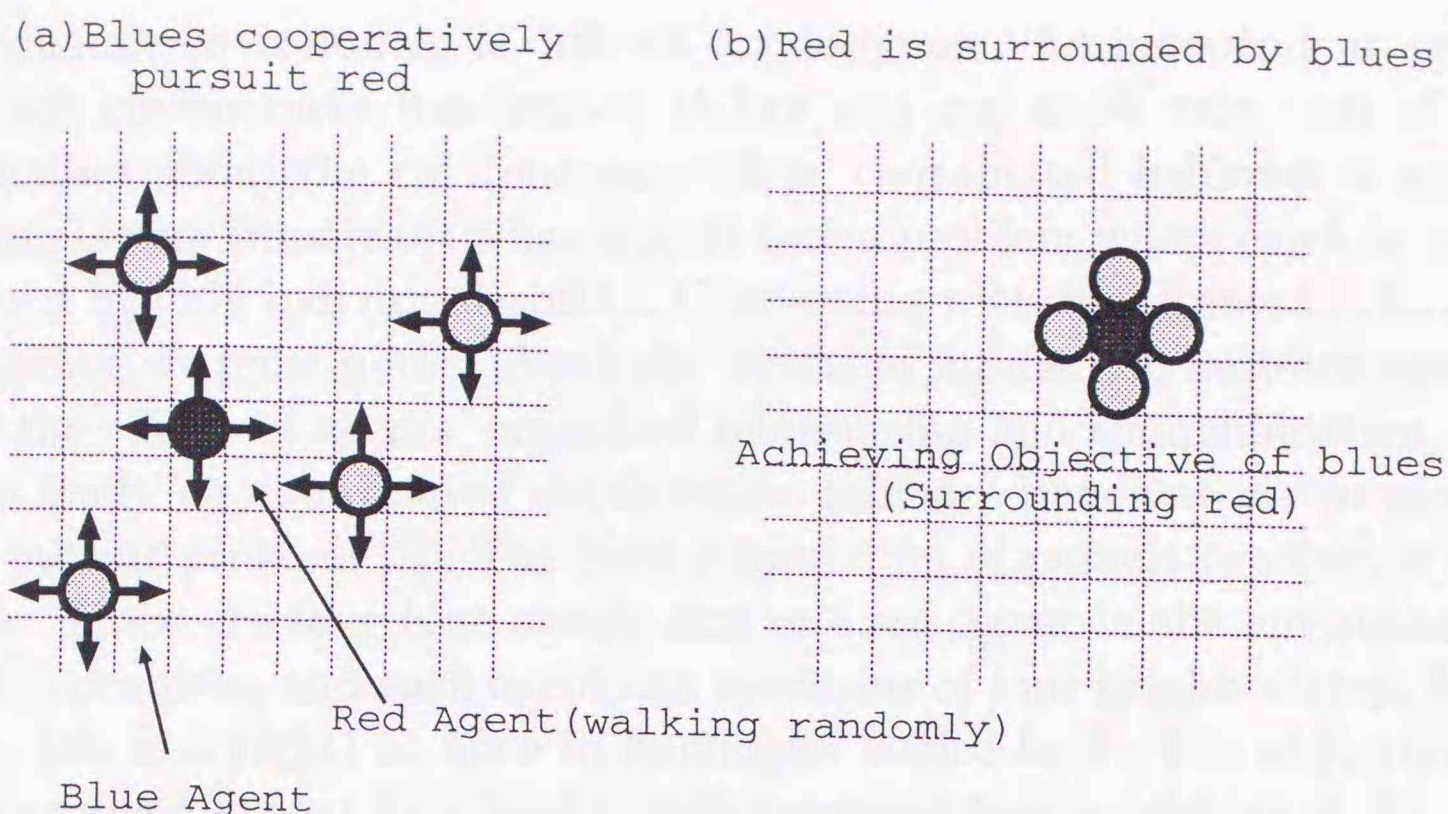


Figure 1.4: An example of Pursuit Problem.

also has to enumerate all possible situations. In dynamic environment of multi-agents, these classical planning methods sometimes cause the frame problem that there are too many situations for an agent to decide an appropriate action, and need a large time to make a decision. Thus, an agent is necessary to decide its plan immediately because the environment of theirs is dynamic. This problem is called as real time planning, or reactive planning, and Pollack et al. proposed tileworld to evaluate agent architectures in viewpoint of such planning problem [39]. The brief introduction of tileworld is as follows.

In tileworld, there are an agent, some tiles, obstacles and holes on grid field (see Fig. 1.3). Each object of tileworld occupies just one grid, and appears and disappears according to probability parameters except an agent. An agent can move to one of four neighbors (top, bottom, left and right) at once. A tile can be pushed to move to one of neighbors. No obstacle can be moved, and a hole can be filled with a tile. A hole filled with a tile by an agent becomes an empty grid. The score of an agent in this world is concerned with the number of filling holes. In this tileworld, an agent is needed to plan its action reactively to maximize its score. This world is used to evaluate agent architectures in that resource bounded environment.

1.2.2.2 The Pursuit Problem

In second subject of the above, controlling of total behavior of agents

in dynamic environment is difficult for designers. An independent agent in such environment has limited ability and can sense only part of information about the environment. Thus, coordinated behavior of multi agents is very important when agents face a problem which can't be performed by only individual ability. Concerning with coordinated behavior of agents, we must notice about the means of interaction between agents and the effects of agents' organized relationship and communication.

To study on coordinated organization scheme, Benda et al. proposed the pursuit problem [2]. The brief formulation of pursuit problem is follows. There are four blue agents and one red agent in the environment of this problem, and each agent can move one of four neighbors (top, bottom, left and right) at once in boundless lattice field. The objective of this problem is that four blue agents surround four neighbors of the red agent which moves randomly. To achieve this objective, there are some constraints for agents: (1) the red agent and blue agents can not occupy the same lattice, (2) arbitrary two blue agents can occupy the same lattice, (3) group of four blue agents and the red agent move alternately. A set of movements combined one movement of all blue agents with one movement of the red agent is called one cycle. This problem requires cooperation between blue agents because it is impossible that only one blue agent attempts to surround the red agent, which coalition with other blue agents is absolutely necessary. Benda investigated this problem from viewpoint how agent organization got better performance to pursue the red agent. He prepared following three relation between two blue agents. (1) Communication agents relation (CAs): one agent can require data for other agent. (2) Negotiating agents relation (NAs): two agents can negotiate goal position or data. (3) Controlling agents relation (CLAs): this relation consists of directional link. The agent in the beginning of the link can control other agents in the close of link. In those three relation, data means the red agent position from one blue agent and goal position of one blue agent. That is, blue agent in this work can observe the position of the red agent. He concluded this work as an organization which had hierarchical controlling line got better performance.

1.2.2.3 The Prisoner's Dilemma

To study of agent policies, the game theory and prisoner's dilemma have been widely used. Here, the prisoner's dilemma is introduced briefly. In the prisoner's dilemma game, there are two players. Each has two choices, namely cooperate or defect. Each must make the choice without knowing

		Column Player	
		Cooperate	Defect
Row Player	Cooperate	$R = 3, R = 3$ Reward for mutual cooperation	$S = 0, T = 5$ Sucker's payoff, and temptation to defect
	Defect	$T = 5, S = 0$ Temptation to defect and sucker's payoff	$P = 1, P = 1$ Punishment for mutual defection

Figure 1.5: The Prisoner's Dilemma.

what the other will do. No matter what the other does, defection yields a higher payoff than cooperation. The dilemma is that if both defect, both do worse than if both had cooperated. The way the game works is shown in Fig. 1.5. One player chooses a row, either cooperating or defecting. The other player simultaneously chooses a column, either cooperating or defecting. Together, these choices result in one of the four possible outcomes shown in that matrix. If both players cooperate, both do fairly well. Both get R , the reward for mutual cooperation. In the concrete illustration of Fig. 1.5 the reward is 3 points. This number might, for example, be a payoff in dollars that each player gets for that outcome. If one player cooperates but the other defects, the defecting player gets the temptation to defect, while the cooperating player gets the sucker's payoff. In the example, these are 5 points and 0 points respectively. If both defect, both get 1 point, the punishment for mutual defection.

The prisoner's dilemma is simply an abstract formulation of some very common and very interesting situations in which what is best for each agent individually leads to mutual defection, whereas everyone would have been better off with mutual cooperation. The definition of prisoner's dilemma requires that several relationships hold among the four different potential outcomes. The first relationship specifies the order of the four payoffs. The best a player can do is get T , the temptation to defect when the other player cooperates. The worst a player can do is get S , the sucker's payoff for cooperating while the other player defects. In ordering the other two outcomes, R , the reward for mutual cooperation, is assumed to be better than P , the punishment for mutual defection. This leads to a preference ranking of the four payoffs from best to worst as T, R, P and S .

The second part of the definition of the prisoner's dilemma is that the

players cannot get out of their dilemma by taking turns exploiting each other. This assumption means that an even chance of exploitation and being exploited is not as good an outcome for a player as mutual cooperation. It is therefore assumed that the reward for mutual cooperation is greater than the average of the temptation and the sucker's payoff. This assumption, together with the rank ordering of the four payoffs, defines the prisoner's dilemma.

Thus two egoists playing the game once will both choose their dominant choice, defection, and each will get less than they both could have gotten if they had cooperated. If the game is played a known finite number of times, the players still have no incentive to cooperate. This is certainly true on the last move since there is no future to influence. On the next-to-last move neither player will have an incentive to cooperate since they can both anticipate a defection by the other player on the very last move. Such a line of reasoning implies that the game will unravel all the way back to mutual defection on the first move of any sequence of plays that is of known finite length. This reasoning does not apply if the players will interact an indefinite number of times. And in most realistic settings, the players cannot be sure when the last interaction between them will take place. With an indefinite number of interactions, cooperation can emerge. The issue then becomes the discovery of the precise conditions that are necessary and sufficient for cooperation to emerge.

The prisoner's dilemma seems very simple, but it is difficult to obtain an optimal strategy for this game. Moreover, we have to consider many factors of the game, e.g., game situations, other player strategies, and so on in order to construct effective strategies. A lot of studies were discussed about the prisoner's dilemma concerning equilibrium strategies, roles of rationality in cooperated relationship, emergence and evolution of cooperation, and so forth.

1.2.3 Formalization of Agent System and Communication

In this sub section, the formalizations of agent systems are introduced. An agent performs a series of basic actions, detecting inputs from the environment, making a decision according to inputs, and effecting its outputs to the environment. In addition, an agent has its policy for decision-making of its actions, and is usually required to have learnability and adaptability for being an intelligent one, namely, it must have its internal memory used for learning and adapting. An agent decides its actions for the environment and changes its internal state. Therefore, an

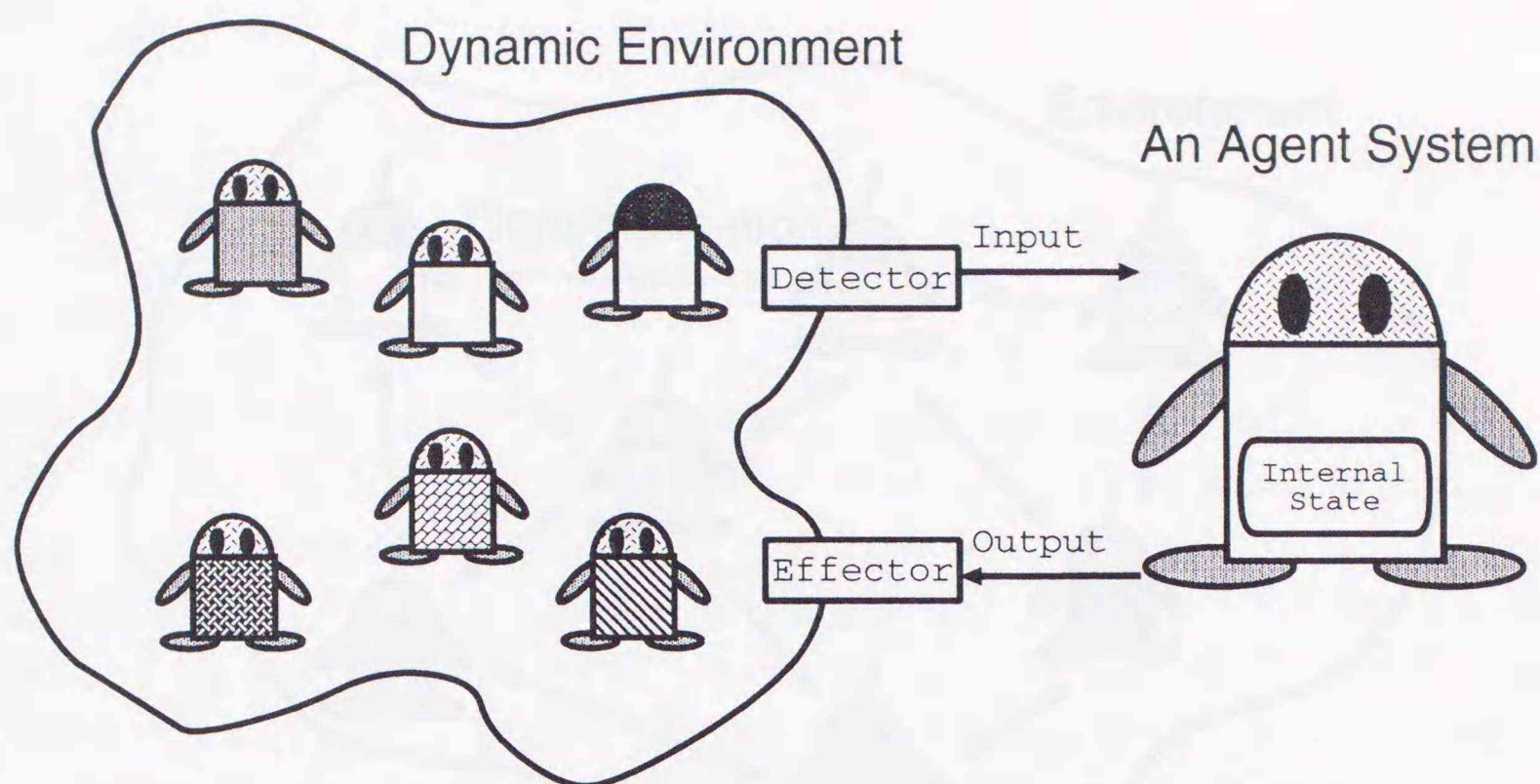


Figure 1.6: The image of relationship between an agent system and the environment.

agent is thought as a system based on inputs, outputs, and internal states (see Fig. 1.6). An agent receives input informations from the environment through a detector, and effects its outputs to the environment through an effector. Because a detector and effector are deeply related with the problem domain or the environment of agents, inputs and outputs of agents are limited characteristically. The formalization of a basal agent system A is given as follows.

$$A = \langle \mathbf{u}, \mathbf{x}, \mathbf{y}, f, g \rangle \quad (1.1)$$

where,

$\mathbf{u}(t)$: Input from the environment at time t

$\mathbf{x}(t)$: Internal state at time t

$\mathbf{y}(t)$: Output to the environment at time t

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (1.2)$$

$$\mathbf{y}(t+1) = g(\mathbf{x}(t), \mathbf{u}(t)) \quad (1.3)$$

In equation 1.2, f is the function to change its internal state according to both of an input and a previous internal state. The function g in equation 1.3 is to decide its output to the environment according to both. Next, I introduce two basic kinds of communication forms, namely, direct communication and indirect communication, to that basic agent system.

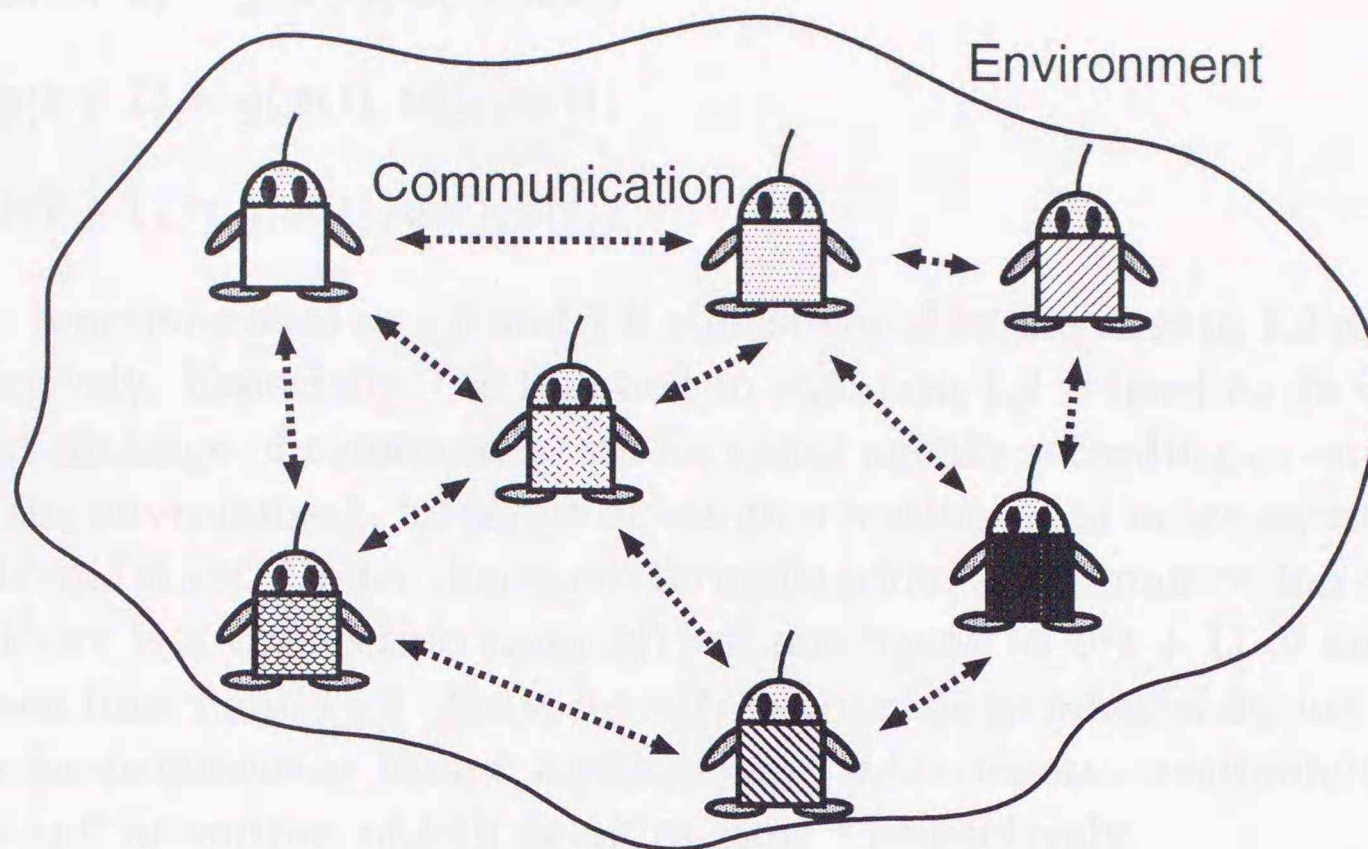


Figure 1.7: The image of direct communication among agents.

1.2.3.1 Direct Communication

The first communication mean can be called the *direct communication*, such like the contract net communication based on agent negotiation. In direct communication, agents directly communicate each other like shown in Fig. 1.7. The connections of communications are dynamically constructed in each time according to a relationship among agents, constraints in the environments, a requirement of communications, etc, although it is often permanent as system settings. The formalization of an agent system based on direct communication can be described as follows.

$$A = \langle u, v, x, y, z, f, g, h \rangle \quad (1.4)$$

where,

$u(t)$: Input from the environment at time t

$v(t)$: Input message of communication from other agents at time t

$x(t)$: Internal state at time t

$y(t)$: Output to the environment at time t

$z(t)$: Output message of communication to other agents at time t

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.5)$$

$$\mathbf{y}(t+1) = g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.6)$$

$$\mathbf{z}(t+1) = h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.7)$$

The functions used in 1.5 and 1.6 almost equal to the ones in 1.2 and 1.3 respectively. Especially, the function in equation 1.7 is used to decide an output message of communication for other agents according to an input from the environment, an input of communication from other agents and an internal state. Under this agent formalization, a communication means that there is a connection from $\mathbf{z}(t)$ of one agent to $\mathbf{v}(t+1)$ of another between time t and $t+1$. Here, let $n(t)$ denote the number of agents being in the environment at time t , and $\mathbf{z}_i(t)$ and $\mathbf{v}_i(t)$ denote a communication input and an output of i -th agent at time t respectively.

Generally, a communication input of an agent consists of communication outputs of others. Thus, let $R(t)$ be a $n(t) \times n(t+1)$ communication structure matrix between time t and $t+1$. Here, let $r_{ij}(t)$ be the element of $R(t)$, the value of $r_{ij}(t)$, $0 \leq i \leq n(t)$, $0 \leq j \leq n(t+1)$, is decided as follows.

$$r_{ij}(t) = \begin{cases} 1 & \text{if } i\text{-th agent sends a message to } j\text{-th agent} \\ 0 & \text{otherwise} \end{cases} \quad (1.8)$$

It is supposed that there is no connection from i -th agent to itself, and $r_{ii}(t)$ usually equals to zero. Accordingly, the communication output and input of i -th agent, $\mathbf{z}_i(t)$ and $\mathbf{v}_i(t)$ is formalized as follows.

$$\mathbf{z}_i(t) = \{\mathbf{z}_{ij}(t) | 0 \leq j \leq n(t), r_{ij}(t) = 1\} \quad (1.9)$$

$$\mathbf{v}_i(t) = \{\mathbf{z}_{ji}(t-1) | 0 \leq j \leq n(t-1), r_{ji}(t) = 1\} \quad (1.10)$$

The communication structure matrix $R(t)$ is dynamically changed according to conditions of the multi-agent system and the environment. For example in Fig. 1.8, the communication structure matrixes are decided as follows.

$$R^a(t) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R^b(t) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (1.11)$$

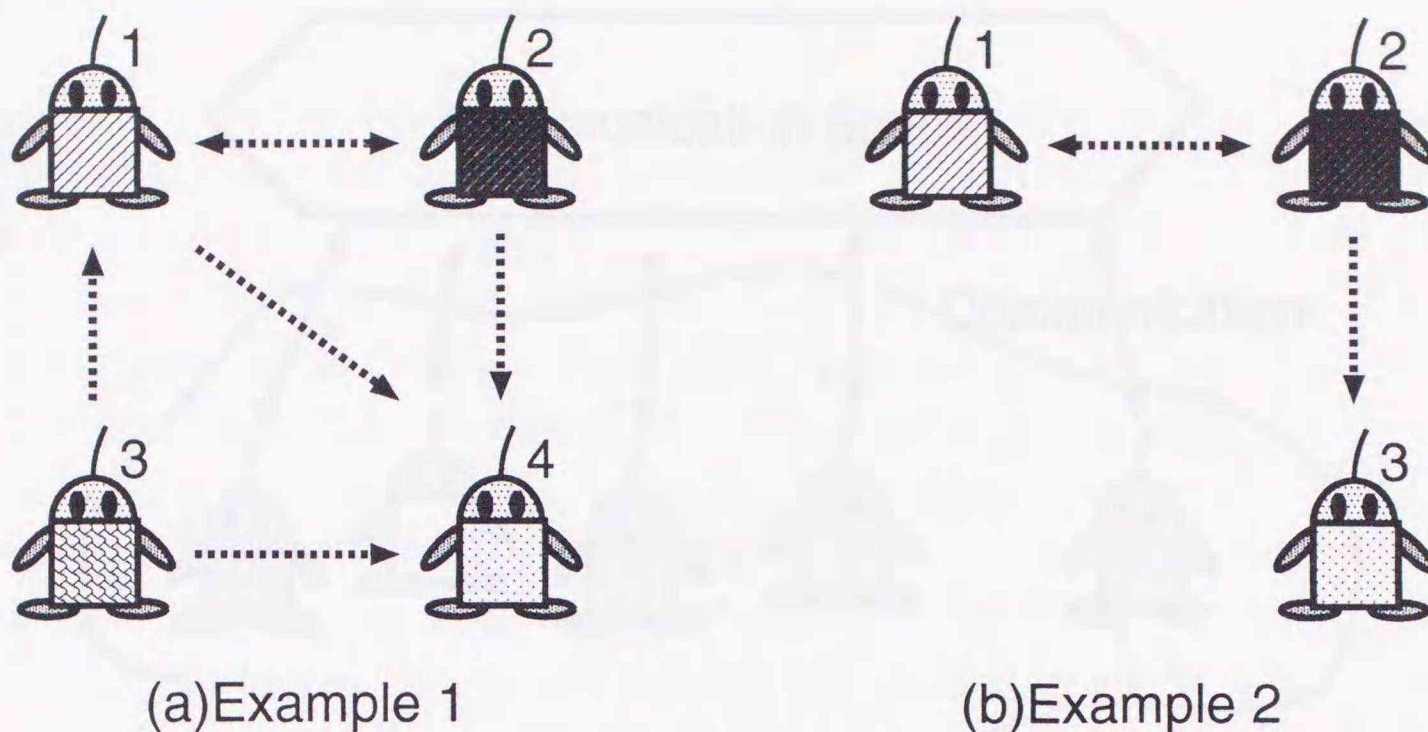


Figure 1.8: The examples of direct communication.

1.2.3.2 Indirect Communication

The second communication mean is the *indirect communication*. In this communication mean, the shared communication space is prepared by the designer in advance. Each agent sends a communication output to the shared communication space, and receives a communication input from the one. Namely, the communication is executed via the shared communication space (see Fig. 1.9). The blackboard model of multi-agent system is included in this communication mean. Therefore, the formalization of an agent system based on indirect communication can be described as follows.

$$A = \langle \mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}, \mathbf{z}, f, g, h \rangle \quad (1.12)$$

where,

$\mathbf{u}(t)$: Input from the environment at time t

$\mathbf{v}(t)$: Communication input from the communication space at time t

$\mathbf{x}(t)$: Internal state at time t

$\mathbf{y}(t)$: Output to the environment at time t

$\mathbf{z}(t)$: Communication output to the communication space at time t

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.13)$$

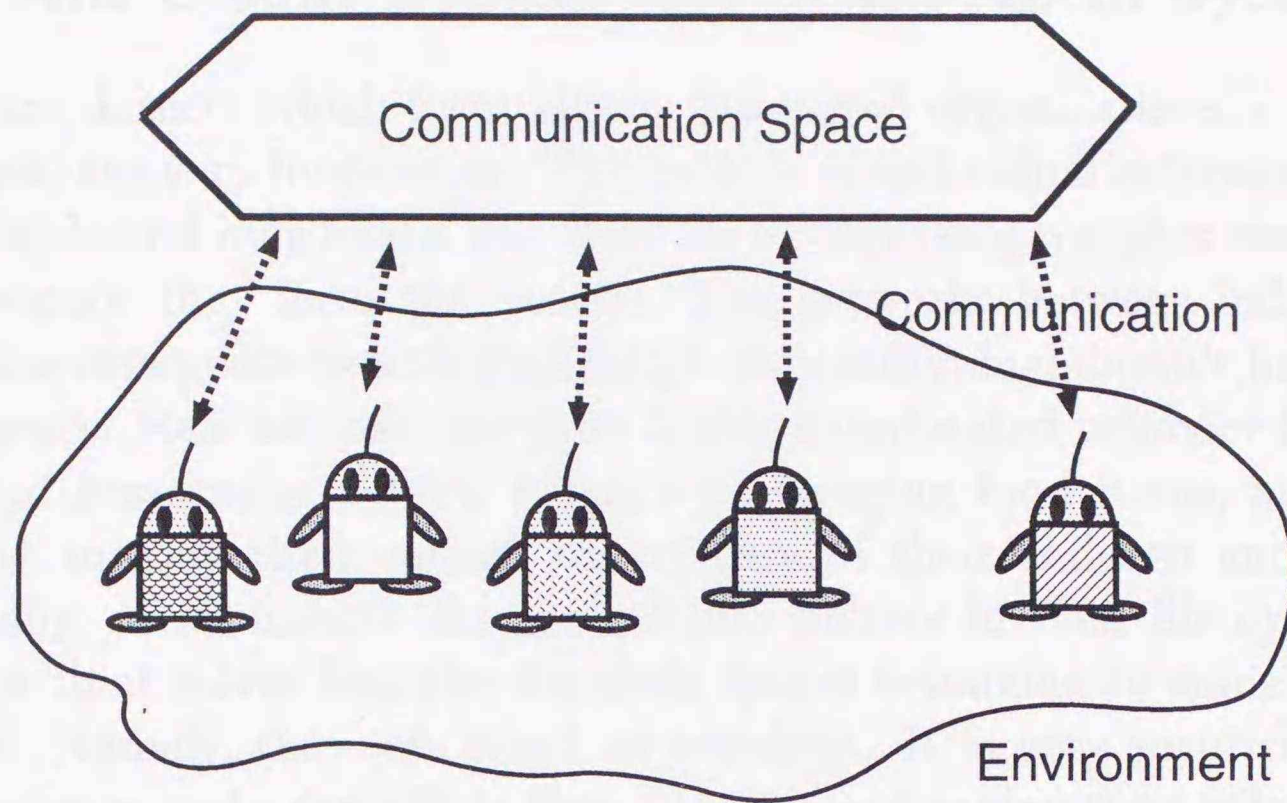


Figure 1.9: The image of indirect communication among agents based on communication space.

$$\mathbf{y}(t+1) = g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.14)$$

$$\mathbf{z}(t+1) = h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1.15)$$

Here, the shared communication space is considered. The shared communication space manages all communication inputs and outputs between agents, namely, it is thought as the system which is connected by the inputs from all agents and outputs to ones. Therefore, the shared communication space system C is formalized as follows.

$$C = \langle \mathbf{V}, \mathbf{X}, \mathbf{Z}, \mathbf{F}, \mathbf{G} \rangle \quad (1.16)$$

$\mathbf{V}(t)$: Input messages of agents as time t

$\mathbf{X}(t)$: Internal state at time t

$\mathbf{Z}(t)$: Output messages of agents at time t

$$\mathbf{X}(t+1) = \mathbf{F}(\mathbf{X}(t), \mathbf{Z}(t)) \quad (1.17)$$

$$\mathbf{V}(t+1) = \mathbf{G}(\mathbf{X}(t), \mathbf{Z}(t)) \quad (1.18)$$

where, let $\mathbf{z}_i(t)$ and $\mathbf{v}_i(t)$ be a communication output and input of i -th agent, and $\mathbf{Z}(t)$ and $\mathbf{V}(t)$ are formalized as follows.

$$\mathbf{Z}(t) = \{\mathbf{z}_1(t), \mathbf{z}_2(t), \dots, \mathbf{z}_{n(t)}(t)\} \quad (1.19)$$

$$\mathbf{V}(t) = \{\mathbf{v}_1(t), \mathbf{v}_2(t), \dots, \mathbf{v}_{n(t)}(t)\} \quad (1.20)$$

The function $\mathbf{F}$ is to update communication space, and $\mathbf{G}$ decides input messages of agents at next time.

1.3 Ant Colony System and Multi-Agent System

In nature, insects which form highly structured organizations, e.g., ants and bees, are very interesting. The pattern of individual activities seems very simple and inordinate, but they show surprising complex swarm behaviors once they form the swarm. The networks between individuals cause the emergence behavior although each individual doesn't have high intelligence. Real ant colonies show highly coordinated behavior for their lives and performing several tasks, e.g., foraging food items, attacking enemies, making their colony, taking care of their children and so on. Especially, social insects like ant colonies behave in their life cycles not only for their selves but also for their mates belonging to same organizations. Namely, they are based on altruism. It is very surprising that they sometimes die for killing their enemies and saving their colonies. In ant colonies, a policy of each individual is strongly connected to total group rationality which have been acquired through the effect of natural evolution and adaptation for keeping their colonies. It is thought that the origin of their total group rationality is caused by their genetic characteristics of breeding system. For example, the genes of females and males are amphiploid and haploid respectively, and the particular genetic crossover mechanism is used. In subsection 1, the breeding system of real ant colonies is described in viewpoint of the origin of organized and coordinated behavior.

Here, it is naturally wondered how they can perform their highly coordination although the number of actions they generally show is basically not many. To answer this question, it is very easy to image that interactions play very important roles in ant colonies. They adaptively use two ways of interactions, direct meeting interactions and pheromone communication. Examples of direct meeting interactions are trading food items, keeping their children, etc. Pheromone communication is the communication via *pheromone* which is chemical substances used by many living creatures. In subsection 2, pheromone communication is surveyed, and the design concept of multi-agent system based on pheromone style communication is proposed. According to subsection 2, some researches concerning with problem solving based on ant-like agents are introduced.

1.3.1 The Origin of Coordinated Behavior on Ant Colony System

The genetic systems of many animals are generally based on ploidy,

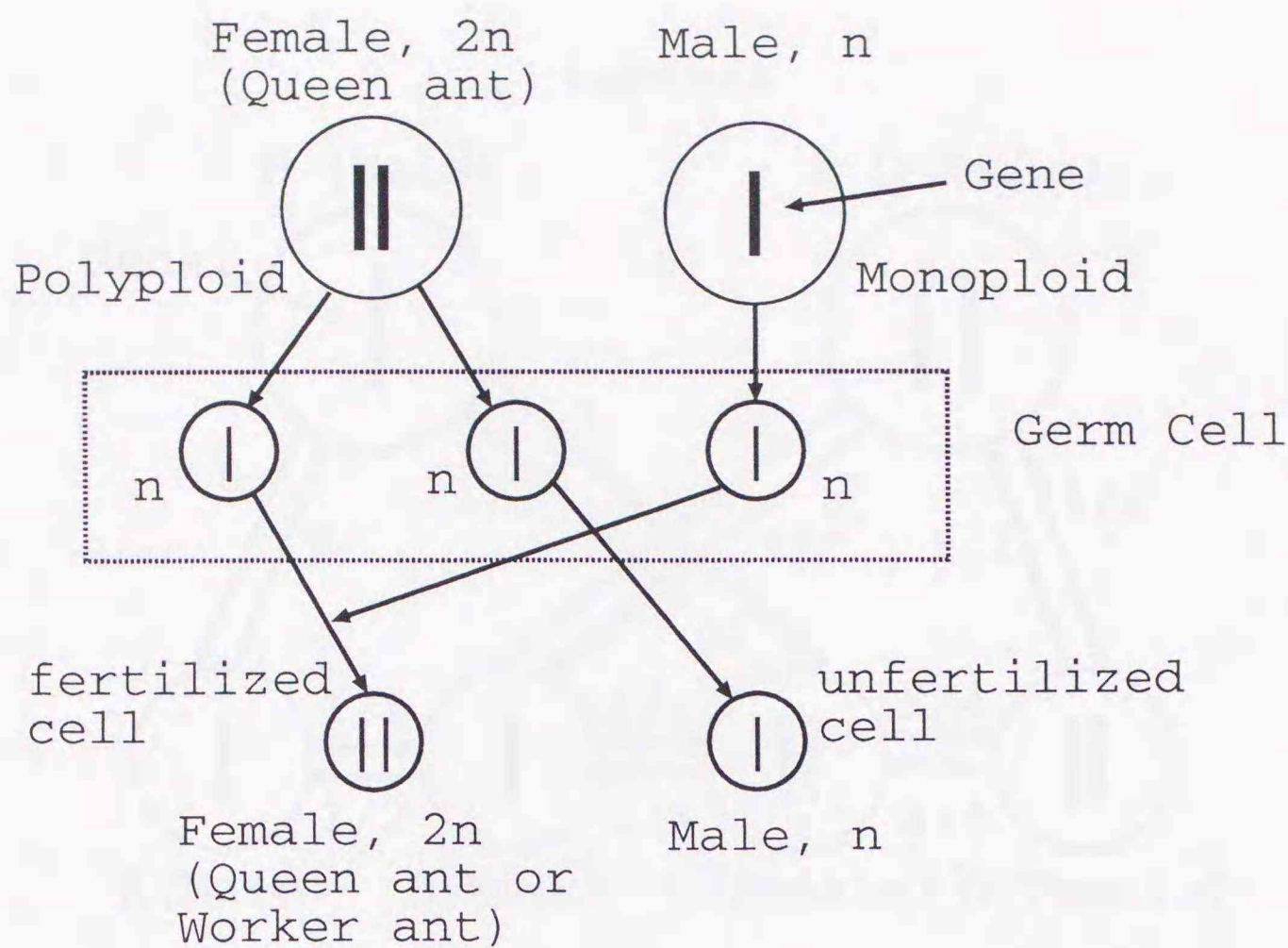


Figure 1.10: The mechanism of crossover on real ant colonies.

in which both of parents, male and female have a pair of genes. For production of children, parents make monoploid germ cells respectively by dividing a pair of genes and crossing over these genes, and new pair of genes for a child is produced by conception of two germ cells. While, females of real ants also have a pair of genes, but males of real ants have only single gene. This genetic system is called half-polyploid system. Fig. 1.10 shows the mechanism of crossover between polyploid female (in real ant colony, it is almost only queen ant) and monoploid male. In this figure, the number of n represents the one of chromosomes including single gene. Thus, polyploid queen ant has $2n$ chromosomes in a pair of genes, and monoploid male ants have n chromosomes. When both of parents generate germ cells for producing their children, each germ cell consists of the number of n chromosomes. Here, if the fertilization of germ cells occurs, fertilized cell is generated with the number of $2n$ chromosomes. If the fertilization does not occur, unfertilized cell remains with keeping the number of n chromosomes. In addition, fertilized cells always become female ants (in many cases workers), and unfertilized cells become male ants. Namely, it is always keeping that females are polyploid and males are monoploid.

Next, it is consider what results this genetic system causes concerning

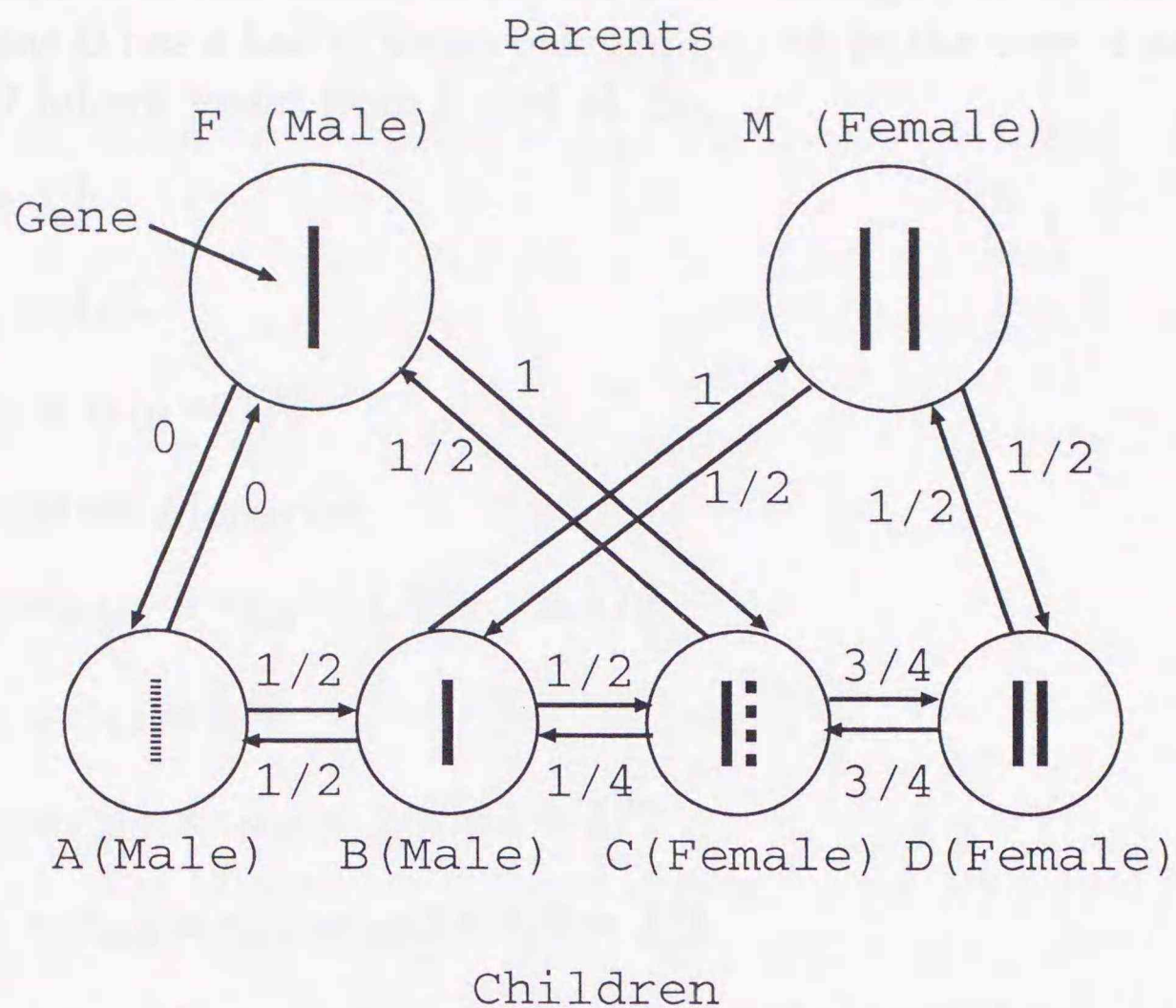


Figure 1.11: The kin degrees between parents and children on real ant colony.

with “kin degree”. The kin degree from A to B (denoted as r_{AB}) is defined as the probability of that B has same particular homoeologous chromosomes included in A. If this degree is large value, both of A and B are probably kin. Oppositely if this value is small, A is not related to B. In ploidy genetic system, it is natural that the kin degree between parents and children always takes the value of $1/2$ because both of parents are polyploid and their children inherit a pair of genes each of which is provided from either parent. In the case of half-polyploid system such as real ant colonies, the relation of kin degree between parents and children is very interesting. Fig. 1.11 shows example relation of kin degree between parents and children on ant colony. In this figure, kin degrees between parents (female and male) and children (two sons and two daughters) are shown. First, the relation between parent and child is considered. The sons A and B are born from unfertilized cells which are generated only by female and inherit one of pair. Accordingly,

$$r_{AF} = r_{FA} = 0 \quad (1.21)$$

$$r_{BM} = 1 \quad (1.22)$$

$$r_{MB} = 1/2 \quad (1.23)$$

Where, equation 1.22 means M has all genes belonging to B, and equation 1.23 means B has a half of genes belonging to M. In the case of daughters, *C* and *D* inherit genes from F and M. So,

$$r_{CF} = 1 \quad (1.24)$$

$$r_{FC} = 1/2 \quad (1.25)$$

$$r_{DM} = r_{MD} = 1/2 \quad (1.26)$$

are calculated. Moreover,

$$r_{AB} = r_{MB} \times r_{AM} = 1/2 \times 1 = 1/2 \quad (1.27)$$

$$r_{BA} = r_{AB} = 1/2 \quad (1.28)$$

$$r_{BC} = r_{MC} \times r_{BM} = 1/2 \times 1 = 1/2 \quad (1.29)$$

$$r_{CB} = r_{MB} \times r_{CM} = 1/2 \times 1/2 = 1/4 \quad (1.30)$$

$$r_{CD} = r_{FD} \times r_{CF} + r_{MD} \times r_{CM} = 1 \times 1/2 + 1/2 \times 1/2 = 3/4 \quad (1.31)$$

$$r_{DC} = r_{CD} = 3/4 \quad (1.32)$$

According to the particular genetic system, the members of ant colony consists of only kin. So, altruistic behaviors for helping their members correspond to behaviors for saving own genes. To discuss altruistic behaviors, W. D. Hamilton proposed the *inclusive fitness* countering the number of genes including kin [27]. The inclusive fitness for the individual A, F_A is defined as follows.

$$F_A = W_A - C + \sum_i r_{Ai} \times B_i \quad (1.33)$$

Where, W_A is the fitness of A without relations to other individuals, C is the decrease fitness of A for altruistic behaviors to other individuals, and B_i is the increase fitness of i -th individual by altruistic behaviors of A. In equation 1.33, if the condition $-C + \sum_i r_{Ai} \times B_i > 0$ is satisfied, altruistic behaviors are rational because such behaviors increase the inclusive fitness. Thus, altruism had been probably fixed in evolutionary process according to this value. In ant colony, the values of r_{Ai} almost take 3/4, and it is easy to satisfy the condition $-C + \sum_i r_{Ai} \times B_i > 0$. This fact is considered as one of reason to cause altruistic behaviors of real ants.

1.3.2 Altruistic Behaviors and Total Group Rationality

In this subsection, the relationship between altruistic behaviors on ant colonies and total group rationality on multi-agent system is described [37]. In the multi-agent system $G = \{A_i : i = 1, 2, \dots, n\}$, let the utility function of agent $A_i \in G$ be denoted as $U_i(y(i), y_i)$ with regarding to other agents' strategies, where y_i is the strategy selected by agent A_i , and $y(i) = (y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$ is a set of strategies selected by other agents except the agent A_i . The policy satisfying individual rationality of agent A_i is described as following equation.

$$\phi_i(y(i)) = \arg \max_{y_i} U_i(y(i), y_i) \quad (1.34)$$

The strategy $\phi_i(y(i))$ means the strategy which maximizes the utility function of agent A_i when the strategies of other agents are given.

On one hand, the policy of agents based on total group rationality is considered. The agents based on total group rationality would maximize own utilities and increase the sum of other agents' utilities.

$$\hat{U}_i(y(i), y_i) = U_i(y(i), y_i) + \lambda_i \sum_{j \neq i} U_j(y(j), y_j) \quad (1.35)$$

According to the utility of equation 1.35, the policy based on total group rationality is described as

$$\phi_i(y(i)) = \arg \max_{y_i} \hat{U}_i(y(i), y_i). \quad (1.36)$$

It is approximately supposed that $W_A - C$ in the equation 1.33 equals to $U_i(y(i), y_i)$ in 1.35, and B_i equals to $U_j(y(j), y_j)$. Moreover, r_{Ai} is supposed to take the value of $3/4$, so the equation 1.33 corresponds to the equation 1.35 with the condition $\lambda_i = 3/4$. These supposes lead that each individual ant behaves based on total group rationality, and the usage of analogies with ant colony system is very available to design multi-agent systems as problem solvers.

1.3.3 Pheromone Communication on Ant Colony System

The colonies of real ants behave based on altruistic behaviors, that is, total group rationality. Moreover, the organization of them can be considered as the caste system, in which it is perfectly determined who does what. For example, the queen bears offsprings for their flourish, young workers take care of the queen, eggs, larvas and chrysalises in their colony, ripe workers are employed in dangerous works like defending their colony

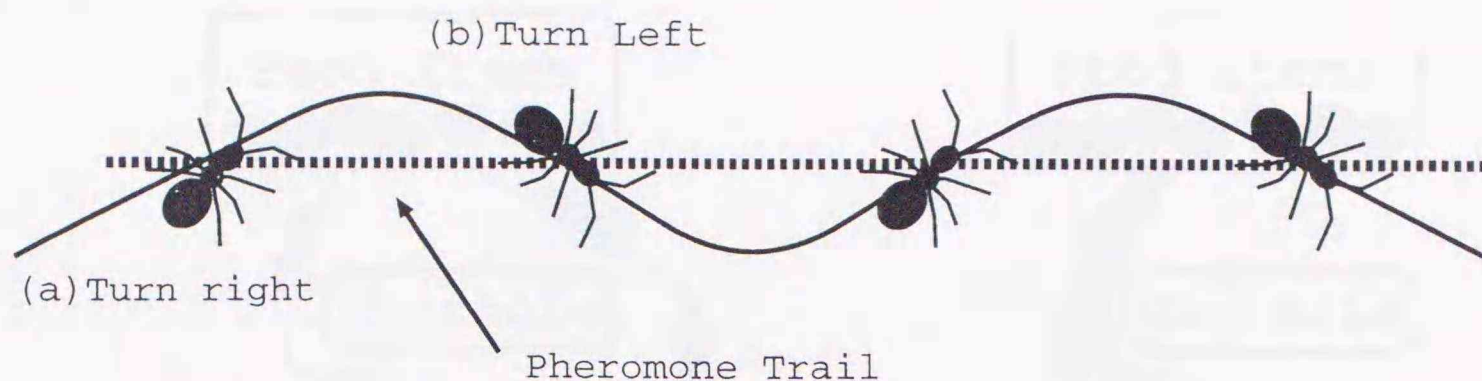


Figure 1.12: The reaction of real ant according to sensing the pheromone trail.

and foraging food items, and even larvae play roles providing nourishments and digestive enzyme to the members of colony [1]. Namely, real ants keep complex society under the distinct division of works. There are two types of division of works, temporal and permanent role. It is known that ants change these morphologies for adapting these temporal roles. Moreover, they autonomously can manage the allocation of labor resources without a specific supervisor (of course the queen doesn't have such ability).

For the emergence of such great organization, it is easily imagined to need excellent communication means. In fact, real ants adaptively use two types of communication means, *sign* and *signpost* [42]. The sign is used for communicating clear informations from one to another, and the signpost is used for incidental informations from conditions of the colony. Real ants can understand what to do by signpost, e.g., they have to forage food items when there are few foods in the storage, or they have to extend the nest when the colony is crowded.

Moreover, real ants use some kinds of chemical substances, called pheromones, for their communications. The word of 'pheromone' was generated by merging the Greek words 'pherein' meaning carry and 'hormon' meaning excite. B. Holldobler guessed that at least 10-20 kinds of signals are needed to employ real ants colonies [29]. Generally, pheromones in ant colonies consist of substances whose characteristics are volatilizable. Pheromones let ants do specific reactions according to these types.

The foraging behavior of real ants is explained briefly. As the basic character of foraging worker, they put the guide pheromone on the return way to their colony after they discover and obtain the food item, i.e., the pheromone trail connected to the food items is formed. On one hand, the ant detecting the pheromone trail with only left tactile sensor turns left, and oppositely the ant detecting with only right tactile sensor turns right

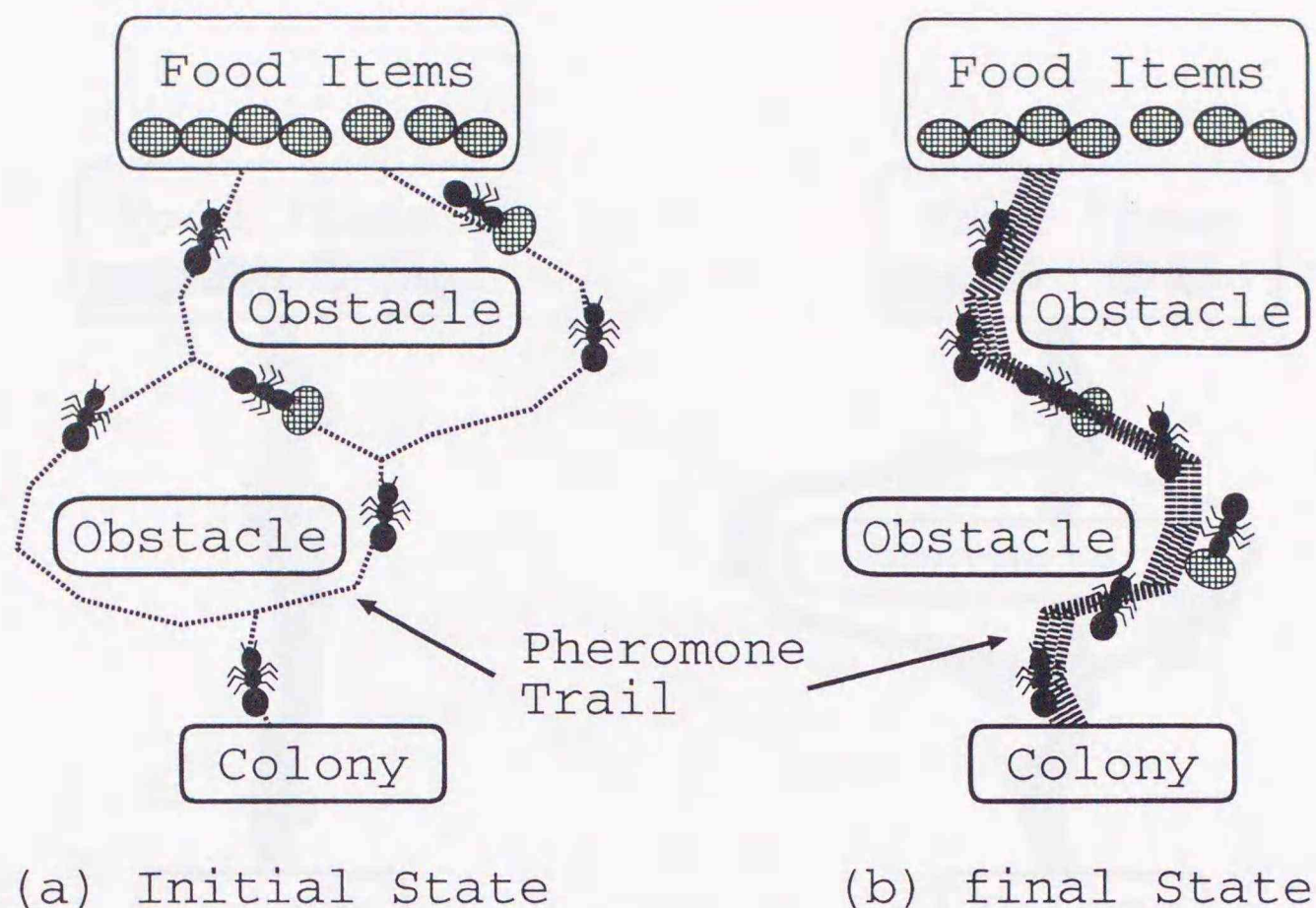
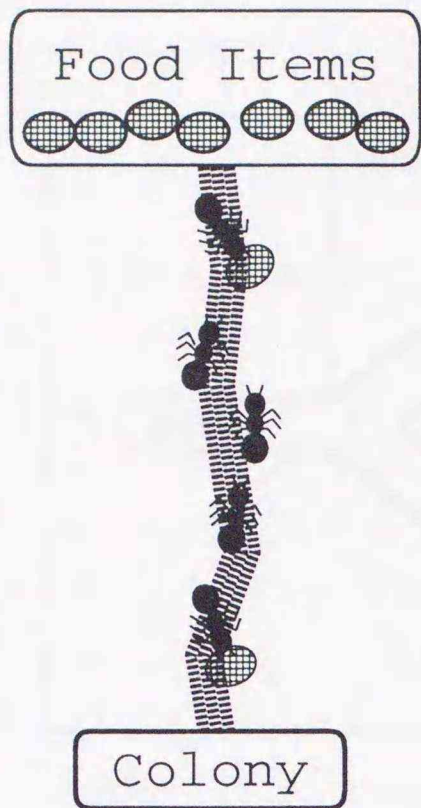


Figure 1.13: The image of whole behavior of foraging ants.

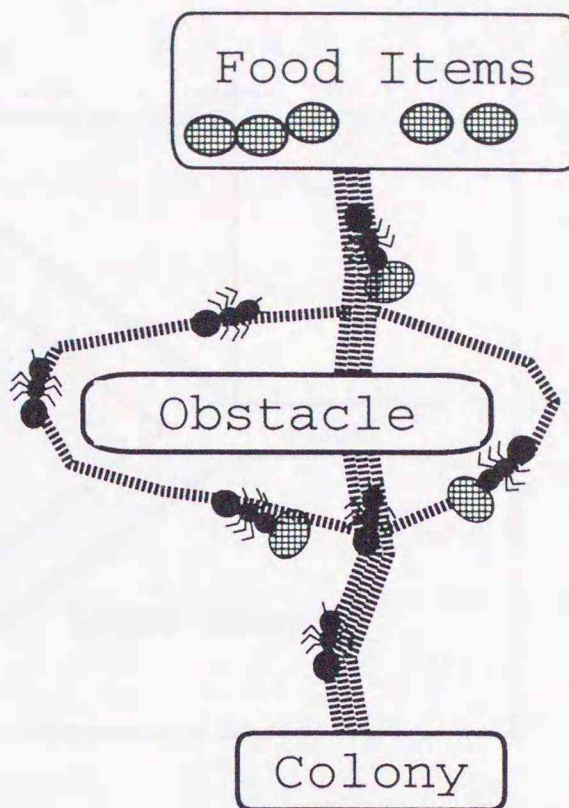
(see Fig. 1.12). This simple reaction enable ants to follow the pheromone trail and reach the food items. It has been shown experimentally that this pheromone trail following behavior can give rise, once employed by a colony of ants, to the emergence of shortest path. That is, when more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back (see Fig. 1.13 and Fig. 1.14).

In order to study in controlled conditions the ants' foraging behavior, the binary bridge experiment has been set up by Deneubourg et al. [16]. The nest of a colony of ants of the species *Linepithema humile* and a food source have been separated by a double bridge in which each branch has the same length (see Fig. 1.15). Ants are then left free to move between the nest and the food source and the percentage of ants which choose one or the other of the two branches is observed over time. The result is that after an initial transitory phase in which some oscillations can appear, ants tend to converge on a same path (see Fig. 1.16).

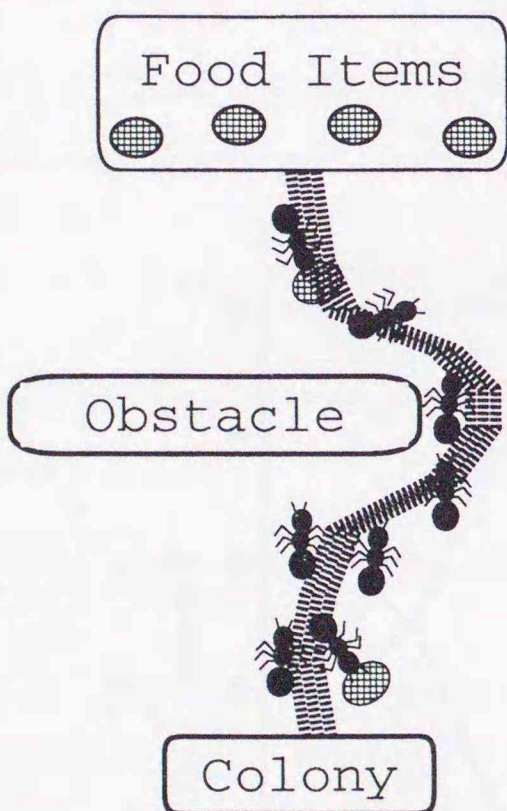
Pheromone indicates the clear sign meaning that they find the food source, moreover, the labor are properly managed and unnecessary pheromone informations are automatically disappeared owing to its characteristic.



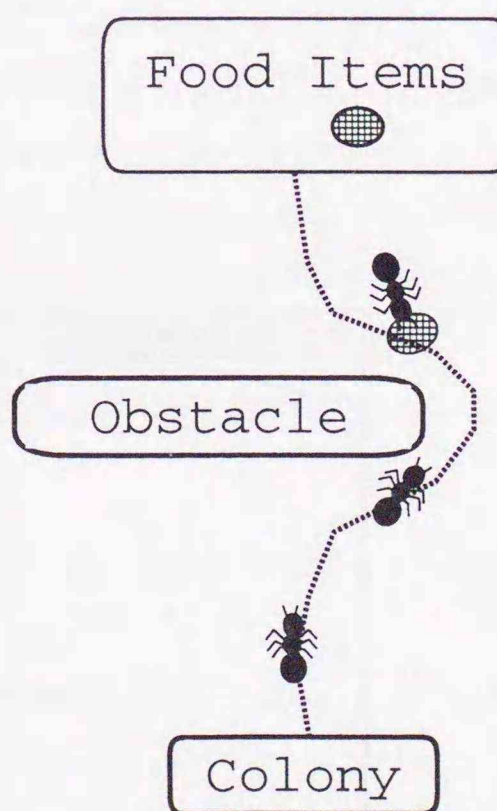
(a) Initial State



(b) The obstacle suddenly appears



(c) The foods are carried



(d) Final State

Figure 1.14: The image of whole behavior of foraging ants when the obstacle suddenly appears.

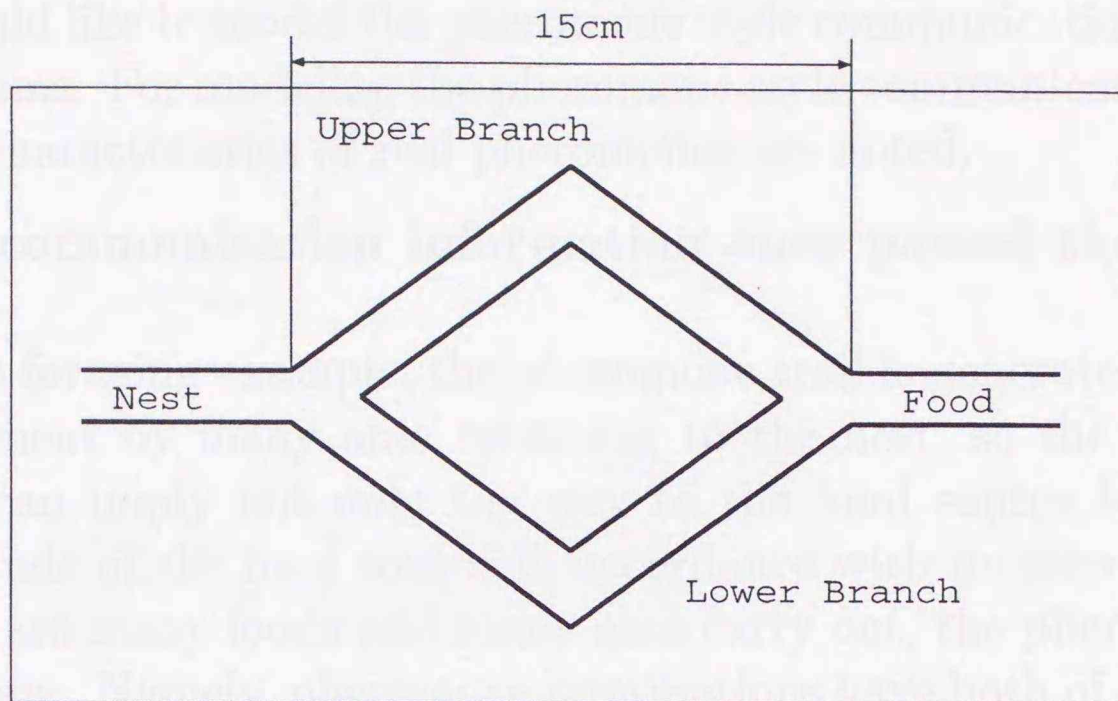


Figure 1.15: Experimental Setting by Deneubourg.

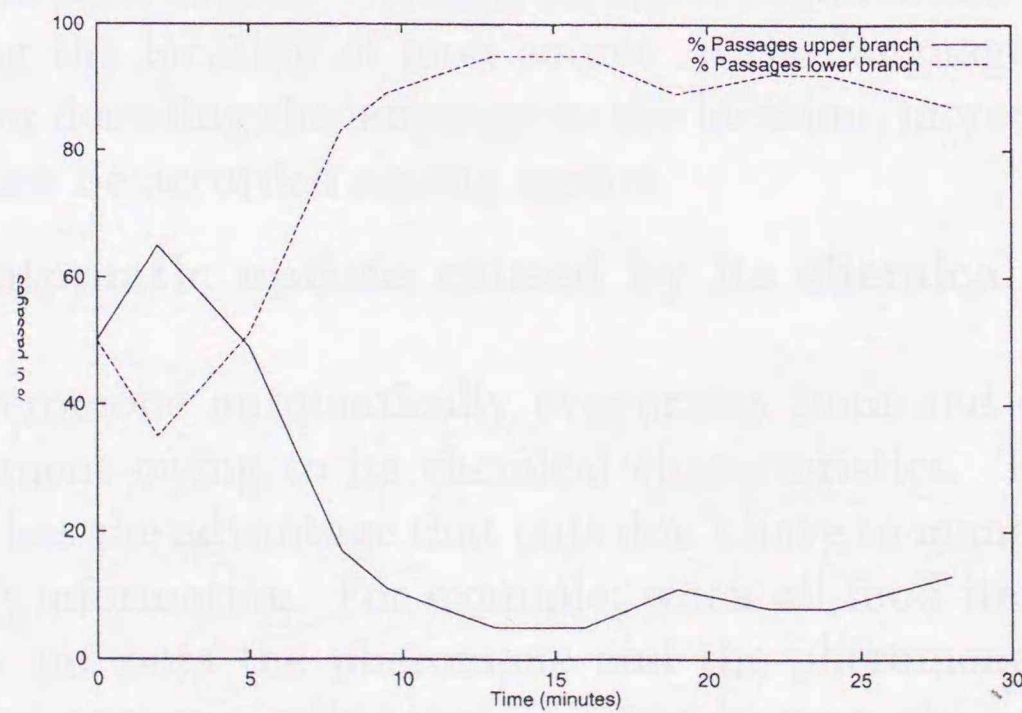


Figure 1.16: Results for a typical single trail by Deneubourg.

1.3.4 Pheromone Style Communication on Multi-Agent System

According to the above example of foraging behavior with pheromone trail, I would like to model the pheromone style communication for multi-agent systems. For modeling the pheromone style communication, following three characteristics of real pheromones are noted.

1. The communication information once passed the environment

In the foraging example, the pheromone trail is generated in the environment by many ants returning to the nest, so the pheromone trail can imply not only the way to the food source but also the plenitude of the food source in accordance with its strength. When there are many foods and many ants carry out, the pheromone trail is strong. Namely, pheromone informations have both of characteristics, the direct sign from others and the indirect sign from conditions of swarm. This feature is very important to emerge adaptive swarm behavior without a specific supervisor which observes and controls the total behavior of swarm. Moreover, it is not necessary for ants to construct the detail internal model like a world map of the environment because the pheromone space is directly constructed in the environment. If ants communicate by using something like languages, it is needed for ants to have the world map inside so as to reach the food source. Namely, an agent requires the world map for encoding the location of food source to the language, and also requires for decoding the language to the location, moreover, the world map must be accorded among agents.

2. The automatic update caused by its chemical characteristics

The pheromone automatically evaporates from and diffuses in the environment owing to its chemical characteristics. This automatic update has the advantage that ants don't have to manage its communicating information. For example, when all food items are carried out, no ant puts the pheromone and the pheromone trail are disappeared automatically, that is, already un-useless information is automatically cleared off without negating the pheromone.

3. The distribution of spatial densities

For communicate with pheromone, ants are not required the ability to process the context of message. In pheromone communication,

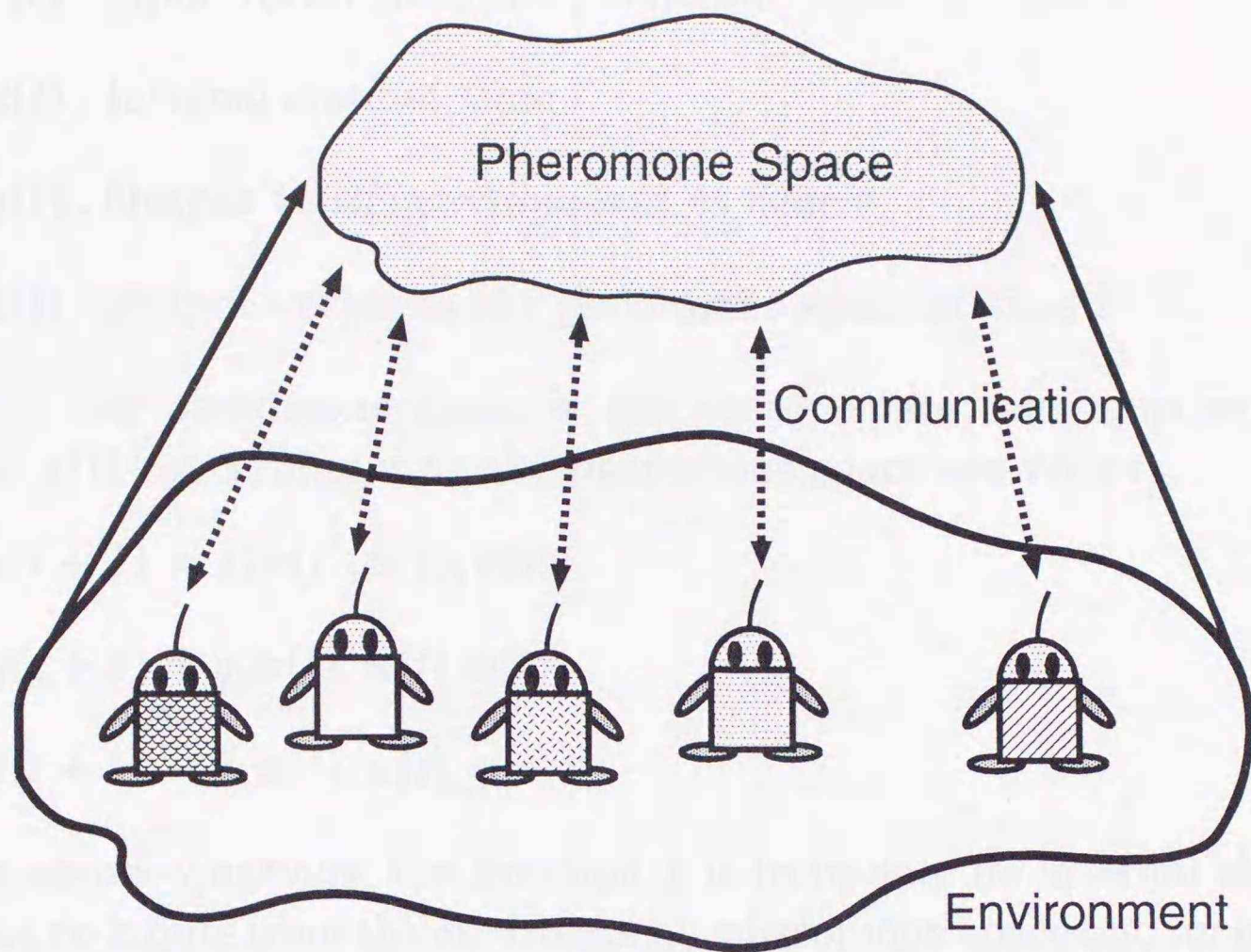


Figure 1.17: The image of pheromone style communication among agents.

the distribution of spatial densities is important, and it is enough for ants to have the ability to react in the environment. In the example of foraging behavior, an ant can reach the food source by only turning to the direction to the strong pheromone trail.

For modeling pheromone style communication, it is needed to formalize the pheromone space and agents. The pheromone space is modeled as the vector space which is designed according to the relation among the objectives of multi-agent systems, the abilities of agents, and the conditions of the environment. As the characteristics of pheromone space, the pheromone space updates a state of vector by itself. An agent behaves based on inputs from the environment and pheromone space, in addition, effects its outputs to the environment and pheromone space. The formalization of an agent is given as follows.

$$A = \langle u, v, x, y, z, f, g, h \rangle \quad (1.37)$$

where,

$u(t)$: Input from the environment at time t

$v(t)$: Input vector from the pheromone space at time t

$x(t)$: Internal state at time t

$y(t)$: Output to the environment at time t

$z(t)$: Output vector to the pheromone space at time t

Because the pheromone space is the vector space, an input $v(t)$ and output $z(t)$ concerning with the pheromone space are vectors.

$$x(t+1) = f(x(t), u(t), v(t)) \quad (1.38)$$

$$y(t+1) = g(x(t), u(t), v(t)) \quad (1.39)$$

$$z(t+1) = h(x(t), u(t), v(t)) \quad (1.40)$$

In the above equations, the function f is to update its internal state according to inputs from the environment, pheromone space and an internal state of mine. The function g is to decide its action to the environment, and h is to decide the output vector of pheromone.

Let $v_i(t)$ and $z_i(t)$ be an input and output of i -th agent at time t , and the pheromone space is formalized as follows.

$$P = \langle v, X, z, F \rangle \quad (1.41)$$

where,

$X(t)$: Internal state vector of pheromone space at time t

$$X(t+1) = F(X(t)) + \sum_i z_i(t) \quad (1.42)$$

$$v_i(t+1) = X(t) \quad (1.43)$$

The function F is concerning with the automatic update of pheromone, and usually includes the effects of evaporation and diffusion.

In addition, it is supposed that the vectors X , v and z are same size. Namely, let $|\cdot|$ means the size of vector, thus

$$\begin{aligned} |X| &= |v_1| = |v_2| = \cdots = |v_i| = \cdots \\ &= |z_1| = |z_2| = \cdots = |z_i| = \cdots \end{aligned} \quad (1.44)$$

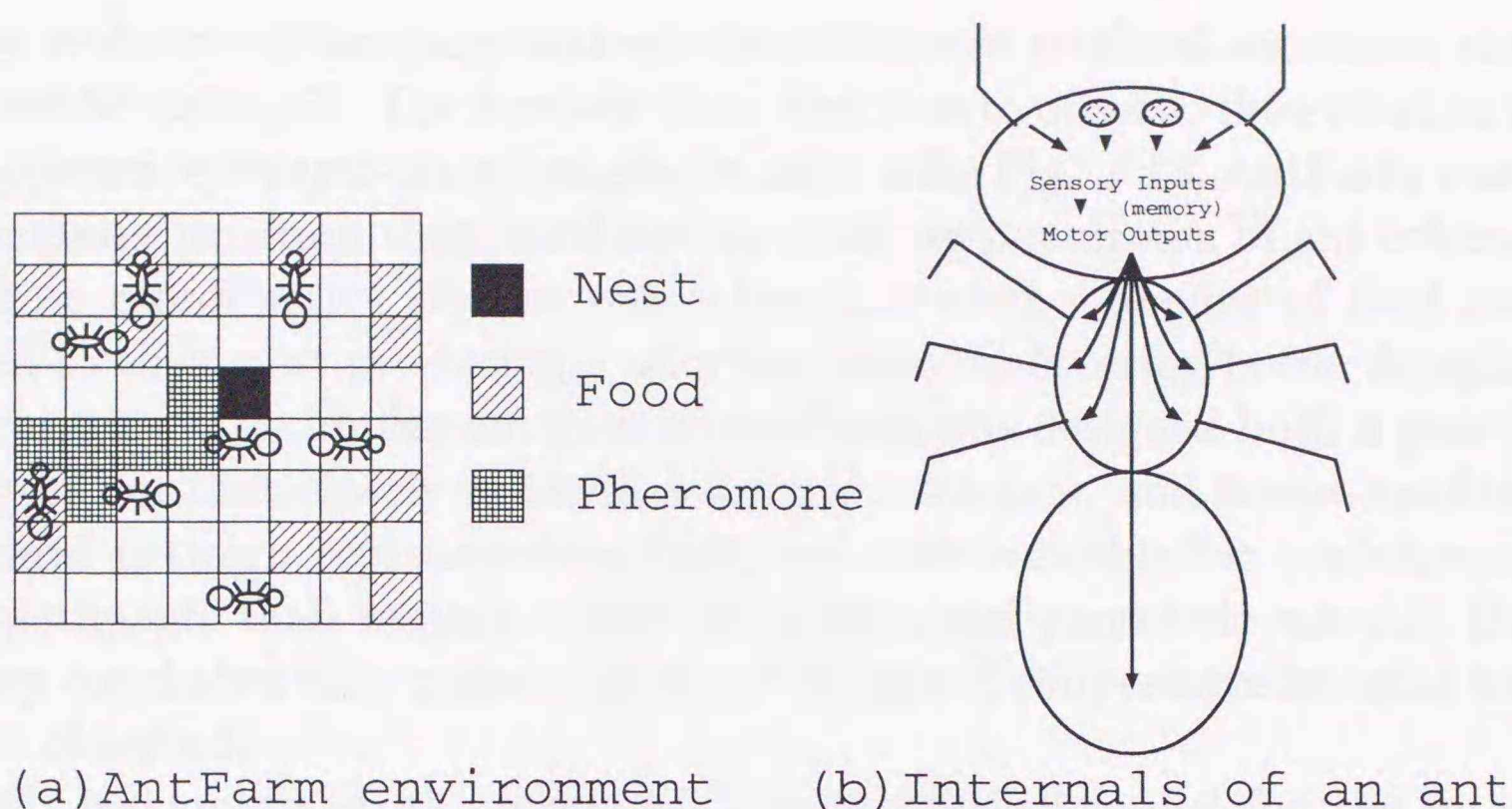


Figure 1.18: The environment and internals of ants in AntFarm.

1.4 The Survey of Related Works about Ant-like Agents and Pheromone Style Communication

In this section, multi-agent systems with ant-like agents and pheromone style communication are surveyed. The study cases are roughly classified two parts, the simulation models and the applications. The simulation models are constructed to study the swarm intelligence which is emerged by local behavior of limited abilities. In such studies, pheromone style communication is introduced to emerge swarm intelligence for achievement of given tasks. On the other hand, the applications are proposed to solve combinatorial optimization problems. Especially, most applications for combinatorial optimization problems are based on the ant colony optimization proposed by Colormi et al. in 1991 [9], and there are many variations of the ant colony optimization. Combinatorial optimization problems treated in these applications are not only static combinatorial problems but also dynamic optimization problems, e.g., telecommunication routing problems. The survey of those researches are described in following subsections.

1.4.1 The Simulation Models of Ant-like Agents and Pheromone Style Communication

Some simulation models are proposed, and the analysis of these models are reported. Collins et al. proposed AntFarm model which simulated

the evolution of foraging strategies in colonies of artificial organisms that resemble ants [15]. The focus of their work was to observe the evolution of cooperative central-place foraging in ants. Like Fig. 1.18, AntFarm was a computer program that simulated an evolving population of ant colonies whose reproductive success was a function of the amount of food carried to their nest producing a selection pressure favoring better foraging strategies. The implementation of AntFarm was designed both a genetic algorithm that closely resembled natural evolution, and a new artificial neural network representation that was more suitable for evolutionary experiments than mating, rather than the usual panmictic scheme. But they concluded their paper that the evolution of cooperative foraging was not observed.

M. Nakamura et al. proposed a mathematical model for the foraging behavior of an ant colony [36]. In the model, a homogeneous ant behaved according to the given rule, and interacted one another by artificial pheromone. As a result of cooperation of many ants' micro-scale behaviors, macro-scale foraging behavior of the colony emerged without any central control or hierarchical control. The model showed characteristic foraging behavior depending on colony size, that disrupted proper foraging. To avoid this behavior, they supposed a temporary desensitization of an ant to pheromone after perceiving a strong pheromone signal. This supposition influenced a formation process of a pheromone pattern, and this enabled us to control the colony's foraging behavior.

S. Kato et al. studied on the multi-agent system with cooperative learning based profit sharing [30]. This model was applied to the pursuit problem in which there were four blue agents as pursuers and one red agent as an evader. The pheromone was used as the communication mean among agents. In the model, when a blue agent couldn't sense a red agent, the pheromone signal was used instead of sensing information. They made some computer experiments to maintain the effectiveness of their proposal.

K. Suzuki et al. discussed reorganization of agents in the multiple monkey banana problem, which can be characterized as multiple targets processing problem [48][49]. In that problem, each target processing requires the cooperative organization that consists of the specified number of agents at least. Because the information of targets is shared insufficiently among agents, each agent has to decide which target should be processing based on its sensing information, and many organizations occur simultaneously for different targets. Thus, it is easily imagined that

some organizations of them will be short-handed for processing the targets. That is, the agents often face the dilemma of whether to give up the current organizing efforts or to wait until the other agents becoming member for filling up the vacancy of the organization. To defeat such dilemma situations, they are required to possess the effective reorganization abilities. They applied reinforcement learning based on classifier rules for acquisition of the pheromone style communication. Throughout the experiments, it was shown that the acquired communication rules succeed in the reorganization from the dilemma situations.

M. Kubo et al. introduced the pheromone style communication to the multi-agents on the soccer game. In the soccer game of agents, there are two teams consists of some agents, and the objective of each team is to get more goal than the opponent. In Kubo's model, an agent behaves with reactive rules based on If-Then rules. Sending a message to team mates is executed by putting the pheromone on arbitrary place on the soccer field. Interpreting pheromone is implemented in conditions of If-Then rules. In addition, learning of rules is based on bucket brigade algorithm. They concluded through computer experiments that (1) agents could emerge cooperative relationship by prediction of mates' behaviors, (2) agents could understand the characteristics of the environment and predict missing plays of the opponent, and (3) a pair of agents could play wall pass with short-term prediction.

F. H. Bennett III studied on emergence of a multi-agent architecture for the ant colony food foraging problem using genetic algorithm, and the automatic evolution of these decisions during a run of genetic programming using architecture-altering operations. In his research, artificial ants were constructed with the genetic programming, and applied to the foraging problem, in which the objective of ant agents is to collect more food items in the environment. The pheromone was used as the marking in the environment and evolved by the genetic programming technique. In the experiments, following interest six agents with coordinated behaviors were appeared through the evolutionary process.

- agent 0: The ant only moves in a straight line in the direction that it was initially facing, and doesn't pick up any food or drop pheromone.
- agent 1: The ant wanders randomly until it finds either a pheromone or food, then it stops next to that pheromone or food and does nothing else. When the pheromone the ant is next to decays or the food pellet the ant is next to is carried away it resumes random wandering.

- agent 2: When the ant happens upon the nest it stops there and does nothing else. Otherwise, it only moves in a straight line in the direction that it was initially facing, and doesn't pick up any food or drop pheromones.
- agent 3: The ant behaves like it does when under the control of agent 1 except that the ant stops on top the food instead of next to do it.
- agent 4: The ant moves to the corner of the world and stops there and does nothing else. However, if it encounters a food pellet and stays there until the food pellet is carried away, whereupon the ant moves to the corner of the world and stops.
- agent 5: Behaves the same as agent 3.

He concluded his work as that genetic programming was successfully used to discover multi-agent algorithms that solve a central-place foraging problem for an ant colony, and was able to simultaneously evolve time-efficient multi-agent algorithms, multi-agent architectures, and emergent cooperation within the colony of ants.

1.4.2 Ant Colony Optimization

In the ant colony optimization (ACO) meta-heuristic a colony of artificial ants cooperate in finding good solutions to difficult discrete optimization problems. Cooperation is a key design component of ACO algorithms: The choice is to allocate the computational resources to a set of relatively simple agents (artificial ants) that communicate indirectly by stigmergy. Good solutions are an emergent property of the agents' cooperative interaction.

Artificial ants have a double nature. On the one hand, they are an abstraction of those behavioral traits of real ants which seemed to be at the heart of the shortest path finding behavior observed in real ant colonies. On the other hand, they have been enriched with some capabilities which do not find a natural counterpart. It is reasonable to give artificial ants some capabilities that, although not corresponding to any capacity of their real ants counterparts, make them more effective and efficient. In the following, the nature inspired characteristics of artificial ants and how they differ from real ants are described.

1.4.2.1 Similarities and differences with real ants

Most of the ideas of ACO stem from real ants. In particular, the use of: (i) a colony of cooperating individuals, (ii) an (artificial) pheromone trail for local stigmergetic communication, (iii) a sequence of local moves to find shortest paths, and (iv) a stochastic decision policy using local information and no lookahead.

- **Colony of cooperating individuals** As real ant colonies, ant algorithms are composed of a population, or colony, of concurrent and asynchronous entities globally cooperating to find a good solution to the task under consideration. Although the complexity of each artificial ant is such that it can build a feasible solution (as a real ant can find somehow a path between the nest and the food), high quality solutions are the result of the cooperation among the individuals of the whole colony.
- **Pheromone trail and stigmergy** Artificial ants modify some aspects of their environment as the real ants do. While real ants deposit on the world's state they visit a chemical substance, the pheromone, artificial ants change some numeric information locally stored in the problem's state they visit. This information takes into account the ant's current history / performance and can be read / written by any ant accessing the state. By analogy, this numeric information is called artificial pheromone trail. In ACO algorithms, local pheromone trails are the only communication channels among the ants. This stigmergetic form of communication plays a major role in the utilization of collective knowledge. Its main effect is to change the way the environment (the problem landscape) is locally perceived by the ants as a function of all the past history of the whole ant colony. Usually, in ACO algorithms an evaporation mechanism, similar to real pheromone evaporation, modifies pheromone information over time. Pheromone evaporation allows the ant colony to slowly forget its past history so that it can direct its search towards new directions without being over-constrained by past decisions.
- **Shortest path searching and local moves** Artificial and real ants share a common task; to find a shortest (minimum cost) path joining an origin (nest) to destination (food) sites. Real ants do not jump, they just walk through adjacent terrain's states, and so do artificial ants, moving step-by-step through adjacent states of the

problem. Of course, exact definitions of state and adjacency are problem-specific.

- **Stochastic and myopic state transition policy** Artificial ants, as real ones, build solutions applying a probabilistic decision policy to move through adjacent states. As for real ants, the artificial ants' policy makes use of local information only and it does not make use of lookahead to predict future states. Therefore, the applied policy is completely local, in space and time. The policy is a function of both the a priori information represented by the problem specifications (equivalent to the terrain's structure for real ants), and of the local modifications in the environment (pheromone trails) induced by past ants.

Moreover, artificial ants have also some characteristics which do not find their counterpart in real ants.

1. Artificial ants live in a discrete world and their moves consist of transitions from discrete states to discrete states.
2. Artificial ants have an internal state. This private state contains the memory of the ant past actions.
3. Artificial ants deposit an amount of pheromone which is a function of the quality of the solution found.
4. Artificial ants timing in pheromone laying is problem dependent and often does not reflect real ants behavior. For example, in many cases artificial ants update pheromone trails only after having generated a solution.
5. To improve overall system efficiency, ACO algorithms can be enriched with extra capabilities like lookahead, local optimization, backtracking, and so on, that cannot be found in real ants. In many implementations ants have been hybridized with local optimization procedures, while, so far, only Muchel and Middendorf have used a simple one-step lookahead function and there are no examples of backtracking procedures added to the basic ant capabilities, except for simple recovery procedures used by Di Caro and Dorigo.

1.4.2.2 The ACO meta-heuristic

In ACO algorithms, a finite size colony of artificial ants with the above described characteristics collectively searches for good quality solutions to

the optimization problem under consideration. Each ant builds a solution, or a component of it, starting from an initial state selected according to some problem dependent criteria. While building its own solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem, as seen by the other ants. Ants can act concurrently and independently, showing a cooperative behavior. They do not use direct communication: it is the stigmergy paradigm that governs the information exchange among the ants.

An incremental constructive approach is used by the ants to search for a feasible solution. A solution is expressed as a minimum cost (shortest) path through the states of the problem in accordance with the problem's constraints. The complexity of each ant is such that even a single ant is able to find a (probably poor quality) solution. High quality solutions are only found as the emergent result of the global cooperation among all the agents of the colony concurrently building different solutions.

According to the assigned notion of neighborhood (problem-dependent), each ant builds a solution by moving through a (finite) sequence of neighbor states. Moves are selected by applying a stochastic local search policy directed (i) by ant private information (the ant internal state, or memory) and (ii) by publicly available pheromone trail and a priori problem-specific local information. The ant's internal state stores information about the ant past history. It can be used to carry useful information to compute the value / goodness of the generated solution and / or the contribution of each executed move. Moreover, it can play a fundamental role to manage the feasibility of the solutions. In some problems in fact, typically in combinatorial optimization, some of the moves available to an ant in a state can take the ant to an infeasible state. This can be avoided exploiting the ant's memory. Ants therefore can build feasible solutions using only knowledge about the local state and about the effects of actions that can be performed in the local state.

The local, public information comprises both some problem-specific heuristic information, and the knowledge, coded in the pheromone trails, accumulated by all the ants from the beginning of the search process. This time-global pheromone knowledge built-up by the ants is a shared local long-term memory that influences the ants' decisions. The decisions about when the ants should release pheromone on the environment and how much pheromone should be deposited depend on the characteristics of the problem and on the design of the implementation. Ants

can release pheromone while building the solution (online step-by-step), or after a solution has been built, moving back to all the visited states (online delayed), or both. Autocatalysis plays an important role in ACO algorithms functioning: the more ants choose a move, the more the move is rewarded (by adding pheromone) and the more interesting it becomes for the next ants. In general, the amount of pheromone deposited is made proportional to the goodness of the solution an ant has built (or is building). In this way, if a move contributed to generate a high-quality solution its goodness will be increased proportionally to its contribution.

A functional composition of the locally available pheromone and heuristic values defines ant-decision tables, that is, probabilistic tables used by the ants' decision policy to direct their search towards the most interesting regions of the search space. The stochastic component of the move choice decision policy and the previously discussed pheromone evaporation mechanism avoid a rapid drift of all the ants towards the same part of the search space. Of course, the level of stochasticity in the policy and the strength of the updates in the pheromone, trail determine the balance between the exploration of new points in the state space and the exploitation of accumulated knowledge. If necessary and feasible, the ants' decision policy can be enriched with problem-specific components like backtracking procedures or lookahead.

Once an ant has accomplished its task, consisting of building a solution and depositing pheromone information, the ant dies, that is, it is deleted from the system. The overall ACO meta-heuristic, beside the two above-described components acting from a local perspective (that is, ants generation and activity, and pheromone evaporation), can also comprise so, extra components which use global information and that go under the name of daemon actions in the algorithm reported in Fig. 1.19. For example, a daemon can be allowed to observe the ants' behavior, and to collect useful global information to deposit additional pheromone information, biasing, in this way, the ant search process from a non-local perspective. Or, it could, on the basis of the observation of all the solutions generated by the ants, apply problem-specific local optimization procedures and deposit additional pheromone off-line with respect to the pheromone the ants deposited online.

The three main activities of an ACO algorithm (ants generation and activity, pheromone evaporation, and daemon actions) may need some kind of synchronization, performed by the schedule activities construct of Fig. 1.19. In general, a strictly sequential scheduling of the activities


```

1. procedure ACO_Meta_heuristic()
2.   while (termination_criterion_not_satisfied)
3.     schedule_action
4.       ants_generation_and_activity();
5.       pheromone_evaporation();
6.       daemon_actions(); { optional }
7.     end schedule_activities
8.   end while
9. end procedure

10. procedure ants_generation_and_activity()
11.   while (available_resources)
12.     schedule_the_creation_of_a_new_ant();
13.     new_active_ant();
14.   end while
15. end procedure

16. procedure new_active_ant() { ant lifecycle}
17. initialize_and();
18.  $\mathcal{M}$  = update_ant_memory()
19.   while (current_state  $\neq$  target_state)
20.      $\mathcal{A}$  = read_local_ant_routing_table();
21.      $\mathcal{P}$  = compute_transition_probabilities( $\mathcal{A}$ ,  $\mathcal{M}$ , problem_constraints);
22.     next_state = apply_ant_decision_policy( $\mathcal{P}$ , problem_constraints);
23.     move_to_next_state(next_state);
24.     if (online_step-by-step_pheromone_update)
25.       deposit_pheromone_on_the_visited_arc();
26.       update_ant_routing_table();
27.     end if
28.      $\mathcal{M}$  = update_internal_state();
29.   end while
30.   if (online_delayed_pheromone_update)
31.     evaluate_solution();
32.     deposit_pheromone_on_all_visited_arcs();
33.     update_ant_routing_table();
34.   end if
35.   die();
36. end procedure

```

Figure 1.19: The ACO meta-heuristic in pseudo-code.

is particularly suitable for non-distributed problems, where the global knowledge is easily accessible at any instant and the operations can be conveniently synchronized. On the contrary, some form of parallelism can be easily and efficiently exploited in distributed problems like routing in telecommunications networks.

In Fig. 1.19, a high-level description of the ACO meta-heuristic is reported in pseudo-code. As pointed out above, some described components and behaviors are optional, like daemon activities, or strictly implementation-dependent like when and how the pheromone is deposited. In general, the online step-by-step pheromone update and the online delayed pheromone update components (respectively, lines 24-27 and 30-34 in the new active ant procedure) are mutually exclusive and only in few cases they are both present or both absent (in the case they are absent the pheromone is deposited by the daemon).

ACO algorithms, as a consequence of their concurrent and adaptive nature, are particularly suitable for distributed stochastic problems where the presence of exogenous sources determines a non-stationarity in the problem representation (in terms of costs and / or environment). For example, many problems related to communications or transportation networks are intrinsically distributed and non-stationary and it is often not possible to have an exact model of the underlying variability. On the contrary, because stigmergy is both the only inter-ant communication method and it is spatially localized, ACO algorithms could perform not at best in case of problems where each state has a big-sized neighborhood. In fact, an ant that visits a state with a big-sized neighborhood has a huge number of possible moves among which to choose. Therefore, the probability that many ants visit the same state is very small, and consequently there is little, if any, difference between using or not pheromone trails.

1.4.2.3 Applications of ACO Algorithms

There are now available numerous successful implementations of the ACO meta-heuristic (Fig. 1.19) applied to a number of different combinatorial optimization problems. Looking at these implementations it is possible to distinguish among two classes of applications: those to static combinatorial optimization problems, and those to dynamic ones.

Static problems are those in which the characteristics of the problem are given once and for all when the problem is defined, and do not change while the problem is being solved. A paradigmatic example of such prob-

Problem name	Authors	Year	Main references	Algorithm name
Traveling salesman	Dorigo, Maniezzo & Coloni	1991	[19]	AS
	Gambardella & Dorigo	1995	[21]	Ant-Q
	Dorigo & Gambardella	1996	[17][18][22]	ACS and ACS 3-opt
	Stutzle & Hoos	1997	[45]	MMAS
	Bullnheimer, Hartl & Strauss	1997	[7]	AS _{rank}
Quadratic assignment	Maniezzo, Coloni & Dorigo	1994	[34]	AS-QAP
	Gambardella, Taillard & Dorigo	1997	[23]	HAS-QAP ⁿ
	Stutzle & Hoos	1998	[46]	MMAS-QAP
	Maniezzo & Coloni	1998	[33]	AS-QAP ^b
	Maniezzo	1998	[32]	ANTS-QAP
Job-shop scheduling	Coloni, Dorigo & Maniezzo	1994	[9]	AS-JSP
Vehicle routing	Bullnheimer, Hartl & Strauss	1996	[6][8]	AS-VRP
	Gambardella, Taillard & Agazzi	1999	[24]	HAS-VRP
Sequential ordering	Gambardella & Dorigo	1997		HAS-SOP
Graph coloring	Costa & Hertz	1997	[10]	ANTCOL
Shortest common supersequence	Michel & Middendorf	1998	[35]	AS-SCS

Table 1.1: List of applications of ACO algorithms to static combinatorial optimization problems.

Problem name	Authors	Year	Main references	Algorithm name
Connection oriented network routing	Schoonderwoerd, Holland, Bruten & Rothkrantz	1996	[41]	ABC
	White, Pagurek & Oppacher	1998	[51]	ASGA
	Di Caro & Dorigo	1998	[13]	AntNet-FS
	Bonabeau, Henaux, Guerin, Snyers, Kuntz & Theraulaz	1998	[4]	ABC-smart ants
Connection-less network routing	Di Caro & Dorigo	1997	[11][12][14]	AntNet & AntNet-FA
	Subramanian, Druschel & Chen	1997	[47]	Regular ants
	Heusse, Guerin, Snyers & Kuntz	1998	[28]	CAF
	van der Put & Rothkrantz	1998	[50]	ABC-backward

Table 1.2: List of applications of ACO algorithms to dynamic combinatorial optimization problems.

lems is the classic traveling salesman problem, in which city locations and their relative distances are part of the problem definition and do not change at run-time. On the contrary, dynamic problems are defined as a function of some quantities whose value is set by the dynamics of an underlying system. The problem changes therefore at run-time and the optimization algorithm must be capable of adapting online to the changing environment.

Topological modifications (e.g., adding or removing a node), which are not considered by the above classification, can be seen as transitions between problems belonging to the same class. Tab. 1.1 and 1.2 list the available implementations of ACO algorithms.

1.4.2.4 Ant Colony System for Traveling Salesman Problems

Here, the original ant colony system for traveling salesman problems is explained as the basic of ACO algorithms. The ant colony system was proposed by Colormi et al. in 1991 [9], and in his work an artificial ant is an agent which moves from city to city on a TSP graph. It chooses the city to move to using a probabilistic function both of trail accumulated on edges and of a heuristic value, which was chosen here to be a function of the edges length. Artificial ants probabilistically prefer cities that are connected by edges with a lot of pheromone trail and which are close-by. Initially, m artificial ants are placed on randomly selected cities. At each time step they move to new cities and modify the pheromone trail on the edges used - this is termed *local trail updating*. When all the ants have completed a tour the ant that made the shortest tour modifies the edges belonging to its tour - termed *global trail updating* - by adding an amount of pheromone trail that is inversely proportional to the tour length.

These are three ideas from natural ant behavior to artificial ant colony: (i) the preference for paths with a high pheromone level, (ii) the higher rate of growth of the amount of pheromone on shorter paths, and (iii) the trail mediated communication among ants. Artificial ants were also given a few capabilities which do not have a natural counterpart, but which have been observed to be well suited to the TSP application: artificial ants can determine how far away cities are, and they are endowed with a working memory M_k used to memorize cities already visited (the working memory is emptied at the beginning of each new tour, and is updated after each time step by adding the new visited city).

There are many different ways to translate the above principles into a computational system apt to solve the TSP. In the ant colony system,

an artificial ant k in city r chooses the city s to move to among those which do not belong to its working memory M_k by applying the following probabilistic formula:

$$s = \begin{cases} \arg \max_{u \in M_k} \{ [\tau(r, u)] \cdot [\eta(r, u)]^\beta \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (1.45)$$

where $\tau(r, u)$ is the amount of pheromone trail on edge (r, u) , $\eta(r, u)$ is a heuristic function, which was chosen to be the inverse of the distance between cities r and u , β is a parameter which weights the relative importance of pheromone trail and of closeness, q is a value chosen randomly with uniform probability in $[0, 1]$, q_0 ($0 \leq q_0 \leq 1$) is a parameter, and S is a random variable selected according to the following probability distribution, which favors edges which are shorter and have a higher level of pheromone trail:

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \notin M_k} [\tau(r, u)] \cdot [\eta(r, u)]^\beta} & \text{if } s \notin M_k \\ 0 & \text{otherwise} \end{cases} \quad (1.46)$$

where $p_k(r, s)$ is the probability with which ant k chooses to move from city r to city s .

Pheromone trail is changed both locally and globally. Global updating is intended to reward edges belonging to shorter tours. Once artificial ants have completed their tours, the best ant deposits pheromone on visited edges; that is, on those edges that belong to its tour. (The other edges remain unchanged.) The amount of pheromone $\Delta\varphi(r, s)$ deposited on each visited edge (r, s) by the best ant is inversely proportional to the length of the tour: The shorter the tour the greater the amount of pheromone deposited on edges. This manner of depositing pheromone is intended to emulate the property of differential pheromone trail accumulation, which in the case of real ants was due to the interplay between length of the path and continuity of time. The global trail updating formula is $\varphi(r, s) \leftarrow (1 - \alpha) \cdot \varphi(r, s) + \alpha \cdot \Delta\varphi(r, s)$, where $\Delta\varphi(r, s) = (\text{shortest tour})^{-1}$. Global trail updating is similar to a reinforcement learning scheme in which better solutions get a higher reinforcement.

Local updating is intended to avoid a very strong edge being chosen by all the ants: Every time an edge is chosen by an ant its amount of pheromone is changed by applying the local trail updating formula: $\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \tau_0$, where τ_0 is a parameter. Local trail updating is also motivated by trail evaporation in real ants.

Problem name	ACS	GA	EP	SA	AG	Optimum
Oliver30 (30-city problem)	420 (423.74) [830]	421 (N/A) [3,200]	420 (423.74) [40,000]	424 (N/A) [24,617]	420 (N/A) [12,620]	420 (423.74)
Eil50 (50-city problem)	425 (427,96) [1,830]	428 (N/A) [25,000]	426 (427.86) [100,000]	443 (N/A) [68,512]	436 (N/A) [28,111]	425 (N/A)
Eil75 (75-city problem)	535 (542.31) [3,480]	545 (N/A) [80,000]	542 (549.18) [325,000]	580 (N/A) [173,250]	561 (N/A) [95,506]	535 (N/A)
KroA100 (100-city problem)	21,282 (21,285.44) [4,820]	21,761 (N/A) [103,000]	(N/A) (N/A) [N/A]	(N/A) (N/A) [N/A]	(N/A) (N/A) [N/A]	21,282 (N/A)

Table 1.3: Comparison of ACS with the other approximate optimization algorithms.

Problem name	ACS best result (1)	ACS average	Optimal result (2)	% Error (1)-(2) (2)	CPU seconds to generate a tour
d198 (198-city problem)	15,888 [585,000]	16,054 [71.1]	15,780	0.68 %	0.02
pcb442 (442-city problem)	51,268 [595,000]	51,690 [188.7]	50,779	0.96 %	0.05
att532 (532-city problem)	28,147 [830,658]	28,522 [275.4]	27,686	1.67 %	0.07
rat778 (778-city problem)	9,015 [991,276]	9,066 [28.2]	8,806	2.37 %	0.13
fl1577 (fl1577-city problem)	22,977 [942,000]	23,163 [116.6]	[22,137 - 22,249]	3.27 ÷ 3.79 %	0.48

Table 1.4: ACS performance for some bigger TSPs.

M. Dorigo applied the ant colony system to some traveling salesman problems and reported the experimental results [17]. Using the test problems listed in Tab. 1.3 the performance of ACS was compared with the performance of other naturally inspired global optimization methods: genetic algorithm (GA), evolutionary programming, simulated annealing (SA), and a combination of simulated annealing and genetic algorithm. Numerical experiments were executed with ACS, whereas the performance figures for the other algorithms were taken from the literature. The ACS parameters were reported to be set to the following values: $m = 10$, $\beta = 2$, $\alpha = 0.1$, $q_0 = 0.9$, $\tau_0 = (n \cdot L_{nn})^{-1}$, where L_{nn} is the tour length produced by the nearest neighbor heuristic, and n is the number of cities (these values were found to be very robust across a wide variety of

problems). In tab. 1.3, the best integer tour length, the real tour length and the number of tours required to find the best integer tour length were reported. The best results of each problem is boldface.

He also ran ACS on some bigger problems to study its behavior for increasing problem dimensions (see Tab. 1.4). For these runs he implemented a slightly modified version of ACS which incorporates a more advanced data structure known as *candidate list*, a data structure normally used when trying to solve big TSP problems. A candidate list is a list of preferred cities to be visited; it is a static data structure which contains, for a given city i , the cl closest cities. First column in the table is the best results obtained by ACS out of 15 trials, namely, the integer length of the shortest tour, and the number of tours which were generated before finding it in square brackets. Second column is ACS average on 15 trials and its standard deviation in square brackets. Third column is optimal result. Forth column is error percentage, a measure of the quality of the best result found by ACS. Fifth column is time required to generate a tour on a Sun Sparc-server (50 MHz).

Dorigo concluded his work as follows. The key to the application of ACS to a new problem is to identify an appropriate representation for the problem (to be represented as a graph searched by many artificial ants), and an appropriate heuristic that defines the distance between any two nodes of the graph. Then the probabilistic interaction among the artificial ants mediated by the pheromone trail deposited on the graph edges will generate good, and often optimal, problem solutions.

1.5 Conclusions

In this section, multi-agent systems and problem solvers from nature were noticed, and the relationship between them were described. In multi-agent systems, communication plays very important role for agents to emerge their organizations for achievement of their tasks. Thus, the formalization of multi-agent systems including two communication means, direct communication and indirect communication, were given in this section. On the other hand, a real ant colony and the origin of altruism of a real ant colony were picked up as the excellent archetype of multi-agent system. According to a real ant colony, I formalized the multi-agent system based on pheromone style communication which was modeled the characteristics of real pheromone communication. Three main characteristics, the communication information once passed the environment, the

automatic update caused by its chemical characteristics, and the distribution of spatial densities were included in the formalization. Finally in this section, the survey of multi-agent systems based on ant-like agents and pheromone style communications were described.

Bibliography

- [1] M. Arino, The earth is a planet of ants. *Hebbon-ya*, 1235, (in Japanese).
- [2] M. Beaulieu, V. Jeyaraman and L. Mammarella, On Optimal Co-ordination of Knowledge Agents. Technical Report DCS-G 1070-23, DePaul University, 1995.
- [3] F. H. Bennett III, The theory of a Multi-Agent Architecture and New Theory of the Ant. *Ant Colony Food Foraging Problem Solving: Genetic Programming, From models to animals*. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, P. Maes, M. J. Matarić, L. A. Meyer, J. Pollack, and S. W. Wilson (Eds.), MIT Press, pp. 410-418, 1995.
- [4] E. Bonabeau, R. Dorigo, S. Garcia, R. Guyon, P. Kuentz and G. Theraulaz, Foraging in telecommunication networks with "ant" and like agents and communication applications. *Proceedings of Second International Workshop on Intelligent Agents for Telecommunications Application*. Lecture Notes in AI vol. 1457, Springer Verlag, 1998.
- [5] R. A. Brooks, A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 1985.
- [6] E. Bullnheimer, R. F. Hartl and C. Strauss, An improved ant system algorithm for the vehicle routing problem. Technical Report FOM-10/97, Institute of Management Science, University of Vienna, 1997.
- [7] E. Bullnheimer, R. F. Hartl and C. Strauss, A new rule-based version of the ant system: a computational study. Technical Report FOM-10/97, Institute of Management Science, University of Vienna, 1997.
- [8] E. Bullnheimer, R. F. Hartl and C. Strauss, Applying the ant system to the vehicle routing problem. *Meta-Heuristics: Advances and Trends in Local Search Algorithms for Optimization*, pp. 109-120, 1999.

Bibliography

- [1] M. Azuma, The earth is a planet of ants, Heibon-sya, 1995. (in Japanese)
- [2] M. Benda, V. Jagannathan and R. Dodhiawalla, On Optimal Cooperation of Knowledge Sources, Technical Report BCS-G 2010-28, Boeing AI Center, 1985.
- [3] F. H. Bennett III, Emergence of a Multi-Agent Architecture and New Tactics For the Ant Colony Food Foraging Problem Using Genetic Programming, From animals to animats 4, Proceedings of the Fourth International Conference on Simulations of Adaptive Behavior, P. Maes, M. J. Mataric, J. A. Meyer, J. Pollack, and S. W. Wilson (Eds.), MIT Press, pp.430-439, 1996.
- [4] E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz and G. Theraulaz, Routing in telecommunication networks with "smart" ant-like agents telecommunication applications, Proceedings of Second International Workshop on Intelligent Agents for Telecommunication Applications. Lectures Notes in AI vol.1437, Springer Verlag, 1998.
- [5] R. A. Brooks, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, 1986.
- [6] B. Bullnheimer, R. F. Hartl and C. Strauss, An improved ant system algorithm for the vehicle routing problem, Technical Report POM-10/97, Institute of Management Science, University of Vienna, 1997.
- [7] B. Bullnheimer, R. F. Hartl and C. Strauss, A new rank-based version of the ant system: a computational study, Technical Report POM-10/97, Institute of Management Science, University of Vienna, 1997.
- [8] B. Bullnheimer, R. F. Hartl and C. Strauss, Applying the ant system to the vehicle routing problem, Meta-heuristics: Advance and Trends in Local Search Paradigms for Optimization, pp. 109-120, 1998.

- [9] A. Colorni, M. Dorigo, V. Maniezzo and M. Truibian, Ant system for job-shop scheduling, *Belgian Journal of Operations Research, Statistics and Computer Science*, 34, pp. 39-53, 1994.
- [10] D. Costa and A. Hertz, Ants can colour graphs, *Journal of the Operational Research Society*, 48, pp. 295-305, 1997.
- [11] G. D. Caro and M. Dorigo, AntNet: A mobile agents approach to adaptive routing, Technical Report 97-12, IRIDIA, Universite Libre de Bruxelles, 1997.
- [12] G. D. Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks, *Journal of Artificial Intelligence Research*, 9, pp.317-365, 1998.
- [13] G. D. Caro and M. Dorigo, Extending AntNet for best effort Quality-of-Service routing, Unpublished presentation at First International Workshop on Ant Colony Optimization, 1998.
- [14] G. D. Caro and M. Dorigo, Two ant colony algorithms for best-effort routing in datagram networks, *Proceedings of the Tenth International Conference on Parallel and Distributed Computing and Systems*, pp. 541-546, 1998.
- [15] R. J. Collins and D. R. Jeffers, AntFarm: Towards Simulated Evolution, *Artificial Life II, Proceedings of the Workshop on Artificial Life*, C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen (Eds.), pp. 579-601, 1990.
- [16] J. L. Deneubourg, S. Aron, S. Goss and J. M. Pasteels, The self-organizing exploratory pattern of the argentine ant, *Journal of Insect Behavior*, 3, pp. 159-168, 1990.
- [17] M. Dorigo and L. M. Gambardella, Ant colonies for the traveling salesman problem, *BioSystems*, 43, pp. 73-81, 1997.
- [18] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, 1 (1), pp.53-66, 1997.
- [19] M. Dorigo, V. Maniezzo and A. Colorni, The ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26 (1), pp. 29-41, 1996.

- [20] C. Emmeche, *The Garden in the Machine - The Emerging Science of Artificial Life -*, Princeton University Press, 1994.
- [21] L. M. Gambardella and M. Dorigo, A reinforcement learning approach to the traveling salesman problem, *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 252-260, 1995.
- [22] L. M. Gambardella and M. Dorigo, Solving Symmetric and asymmetric TSPs by ant colonies, *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 622-627, 1996.
- [23] L. M. Gambardella, E. Taillard and M. Dorigo, Ant colonies for the QAP, Technical Report 4-97, IDSIA, 1997.
- [24] L. M. Gambardella and M. Dorigo, HAS-SOP: An hybrid ant system for the sequential ordering problem, Technical Report 11-97, IDSIA, 1997.
- [25] B. Goodwin, *How the leopard changes its spots: the evolution of complexity*, Springer-Verlag, 1994.
- [26] T. R. Halliday and P. J. B. Slater, *Animal Behavior*, Vol. 1, 2, 3, Blackwell Scientific Publications, 1983.
- [27] W. D. Hamilton, The genetical evolution of social behaviour, I, II, *Journal of Theoretical Biology*, 7 (1), pp. 1-52, 1964.
- [28] M. Heusse, S. Guerin, D. Snyers and P. Kuntz, Adaptive agent-driven routing and load balancing in communication networks, Technical Report RR-98001-IASC, Department Intelligence Artificielle et Sciences Cognitives, 1998.
- [29] B. Holldobler and E. O. Willson, *The ants*, The Belknap press of Harvard University Press, 1990.
- [30] S. Kato, K. Matsuo and A. Iwata, Cooperative Learning of Multi-Agent System using Pheromone Information, *Proceedings of 58-th National Conference of IPSJ (CD-ROM)*, 1999. (in Japanese)
- [31] M. Kubo, K. Suzuki and Y. Kakazu, Union of Intentions and Prediction of Environment by using of Pheromone Communication, *Online Proceedings of MACC'95*, 1995. (in Japanese)
<http://www.csl.sony.co.jp/person/nagao/macc95/>

- [32] V. Maniezzo, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, Technical Report CSR 98-1, C. L. In scienze dell'Informazione, Università di Bologna, sede di Cesena, 1998.
- [33] V. Maniezzo and A. Coloni, The ant system applied to the quadratic assignment problem, IEEE Transactions, Knowledge and Data Engineering, 1999.
- [34] V. Maniezzo, A. Coloni and M. Dorigo, The ant system applied to the quadratic assignment problem, Technical Report IRIDIA / 94-28. Université Libre de Bruxelles, 1994.
- [35] R. Michel and M. Middendorf, An island model based ant system with lookahead for the shortest supersequence problem, Proceedings of Fifth International Conference on Parallel Problem Solving from Nature, pp. 692-701, Springer-Verlag, 1998.
- [36] M. Nakamura and K. Kurumatani, Formation Mechanism of Pheromone Pattern and Control of Foraging Behavior in an Ant Colony Model, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems, C. G. Langton and K. Shimohara (Eds.), MIT Press, pp. 67-74, 1997.
- [37] A. Namatame, Multi-Agents and Complex Systems, Morikita publishing, 1998. (in Japanese)
- [38] S. Numaoka, E. Osawa and T. Nagao, Multi-Agent System, Kyoritsu-shuppan, 1998. (in Japanese)
- [39] M. E. Pollack and M. Ringuette, Introducing the Tileworld: Experimentally Evaluating Agent Architectures, Proceedings of AAAI-90, pp. 183-189, 1990.
- [40] C. W. Reynolds, Flocks, herds, and schools: A distributed behavioral model, Proceedings of SIGGRAPH, Computer Graphics 21, pp. 25-34, 1989.
- [41] R. Schoonderwoerd, O. Holland and J. Bruten, Ant-like agents for load balancing in telecommunications networks, Proceedings of First International Conference on Autonomous Agents, pp. 209-216, 1997.
- [42] T. D. Seeley, The Wisdom of the Hive : The Social Physiology of Honey Bee Colonies, Harvard University Press, 1995.

- [43] J. M. Smith and E. Szathmary, The Major Transitions in Evolution, Spektrum Akademischer Verlag, 1995.
- [44] T. Shibata and T. Fukuda, The future of artificial life, jiji-tsushinsya, 1994. (in Japanese)
- [45] T. Stutzle and H. Hoos, The $MAX - MIN$ ant system and local search for the traveling salesman problem, Proceedings of IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, pp. 309-314, 1997.
- [46] T. Stutzle and H. Hoos, $MAX - MIN$ ant system and local search for combinatorial optimization problems, Meta-heuristics: Advance and Trends in Local Search Paradigms for Optimization, pp. 137-154, 1998.
- [47] D. Subramanian, P. Druschel and J. Chen, Ants and reinforcement learning: A case study in routing in dynamic networks, Proceedings of International Joint Conference on Artificial Intelligence, 1997.
- [48] K. Suzuki and A. Ohuchi, Emergent Organization with Pheromone Communication in Multiple Monkey Banana Problem, Proceedings of ITC-CSCC'97, Z. Nakao and Y. E. Chen (Eds.), pp. 479-482, 1997.
- [49] K. Suzuki and A. Ohuchi, Reorganizations of Agents with Pheromone Style Communication in Multiple Monkey Banana Problem, Proceedings of Intelligent Autonomous Systems, IAS-5, Y. Kakazu, M. Wada and T. Sato (Eds.), IOS Press, pp. 615-622, 1997.
- [50] R. van der Put and L. Rothkrantz, Routing in packet switched networks using agents, Simulation Practice and Theory, 1999.
- [51] T. White, B. Pagurek and F. Oppacher, Connection management using adaptive mobile agents, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 802-809. 1998.
- [52] G. C. Williams, The pony fish's glow, Orion Publishing Group Ltd., 1997.

Chapter 2

Multiple Ant Colonies Algorithm for Traveling Salesman Problems

2.1 Introduction

There are many kinds of creatures that show adaptive and flexible behavior to achieve various tasks. Ants, so-called social insects, are one example. Although ants have simple abilities, they exhibit collective behavior to perform tasks that can not be carried out by one individual, such as foraging, building a nest, caring for offspring, and defending their colony from enemies. As a result of cooperation in many kinds of micro-scale behavior, macro-scale complex behavior seems to emerge without any central or hierarchical control. In ant colonies, chemical substances called pheromones are used as a way of communicating important information concerning their colonies [1]. For example, ants looking for food lay the way back to their nest with a specific type of pheromone. Other ants can follow the pheromone trail and find their way to baits efficiently.

Studies on the behavior of ants in colonies and on pheromone communication may be useful for the development of artificial intelligence. Researchers in various fields have studied the behavior of ants. Nakamura, Suzuki and Mikami studied on multiagent systems based on the behavior of ants and their organization ability based on pheromone-style communication [16][20][14]. For optimizations, I have constructed multiagent-based optimization algorithms based on pheromone-style communication for Vehicle Routing Problems and Nurse Scheduling Problems [11][21]. Subramanian proposed distributed routing algorithms based on simple biological ants for routing in packet-switched communication networks [19]. Caro also proposed an approach using mobile agents based on the AntNet for adaptive communication networks routing [3]. Kuntz applied ant-like agents to partitioning problems in VLSI technology [12].

The ant algorithm, originally proposed by Coloni et al. in 1991 for Traveling Salesman Problems (TSPs) [4][5][8][9], is also one of optimization algorithms inspired from the analogy of the foraging behavior of ants and interactions between ants in a colony. The optimization of the ant algorithm is based on low-level interactions among a large number of cooperating simple agents, like ants, which are not aware of their cooperative behavior. To solve TSPs, the ant agents continuously move from one city to another unvisited city in a TSP with opinions to prefer subtours connecting near cities and they lay strong pheromones. The ant agents visiting all cities lay some intensity of the pheromones with subtours included in their completed tours according to the distances of tours. That is, subtours that have a possibility of being better in TSP tours tend to have strong pheromones, and the ant agents specify the good regions in the search space by using this feedback mechanism. Several variations of the ant algorithm, such as the ANT-Q algorithm by Gambardella, the MAX-MIN ant system by Stützle and the rank-based version of the ant system have been proposed [10][18][2]. Moreover, the ant algorithm has been applied not only to TSPs but also to Quadratic Assignment Problems by Maniezzo, Graph Coloring Problems by Costa and Job Shop Problems by Coloni [13][7][6].

I have extended the original ant algorithm to a multiple ant colonies algorithm in order to improve the basic performance of the algorithm, and the details and some experiments will be described in this chapter. This algorithm consists of some independent colonies that basically correspond to the original ant algorithm colony, but colony-level interactions have been introduced. Similar ideas were introduced in the parallel Genetic Algorithm, in which interactions among subpopulations are generally practiced by the exchanging operation of some individuals [15]. In our proposed algorithm, interactions among colonies are more naturally introduced by exchanging information on the pheromones as the schemata for solved problems. The pheromones belonging to one colony have different meanings for other colonies, i.e., there are *positive pheromone effects* and *negative pheromone effects*. The positive pheromone effects force ant agents to choose the way laid on, and the negative pheromone effects make the ant agents avoid to choose the way. This mechanism enables ant agents to provide good schemata to agents in other colonies and to share search regions with each other. Moreover, this mechanism seems to be appropriate for parallel computing in the sense of allocating one processor to one colony.

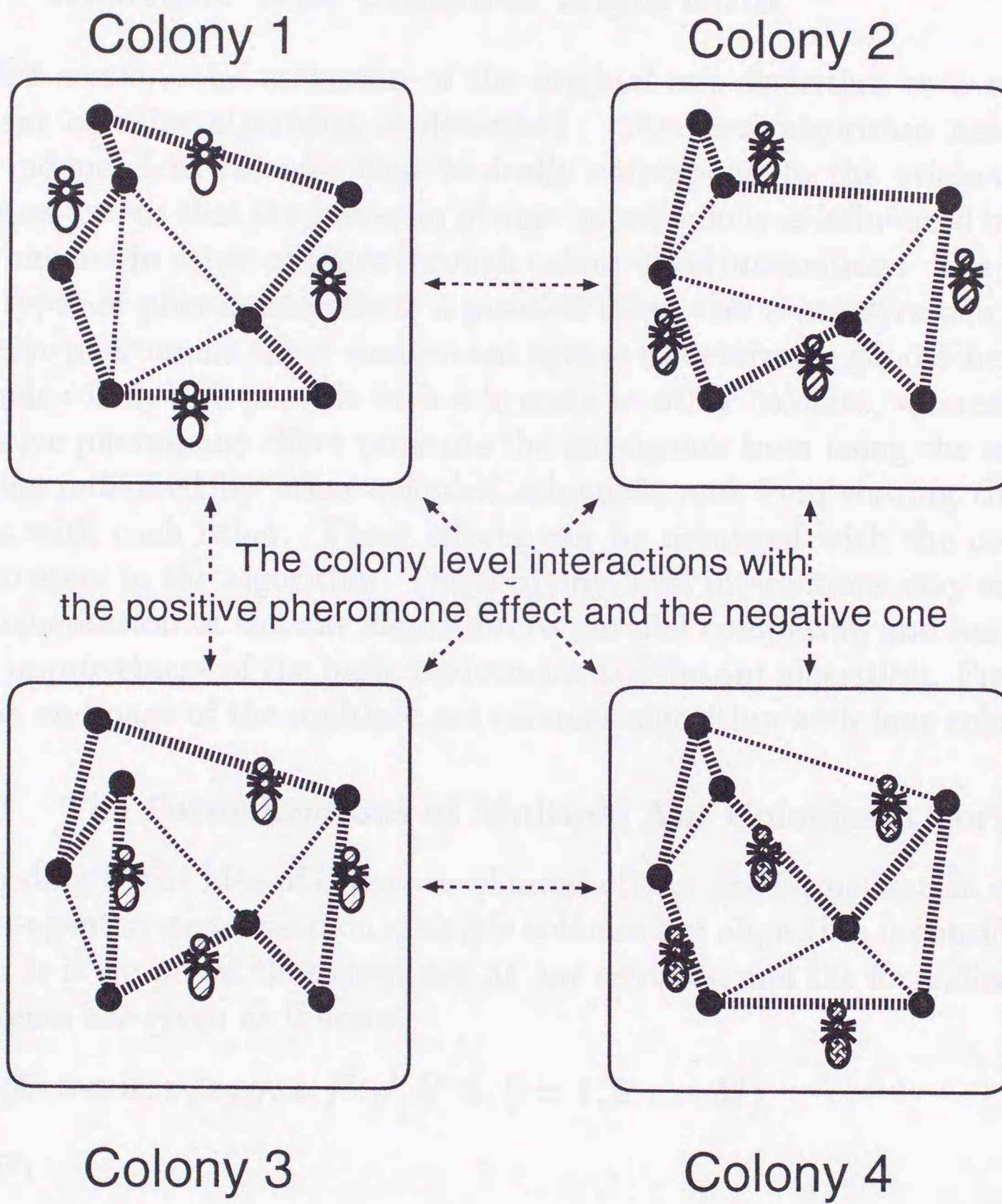


Figure 2.1: An image of the multiple ant colonies algorithm for a TSP with four colonies.

In Section 2 of this chapter, I will introduce the multiple ant colonies algorithm for TSPs. In Section 3, I will present the results of computer experiments on some TSPs. Our proposals will be discussed in Section 4, and conclusions will be presented in Section 5.

2.2 Multiple Ant Colonies Algorithm

In this section, the extension of the original ant algorithm to a multiple ant colonies algorithm is described. This new algorithm has several independent colonies that basically correspond to the original ant colonies except that the behavior of ants in one colony is influenced by the pheromones in other colonies through colony-level interactions. There are two types of pheromone effect: a positive effect and a negative one. The positive pheromone effect enables ant agents to determine good schemata in their colony and provide such schemata to other colonies, whereas the negative pheromone effect prevents the ant agents from using the search regions indicated by other colonies' schemata and from sharing the regions with each other. These effects can be managed with the control parameters in the algorithm. These colony-level interactions may enable implementation of the ant algorithm to parallel computing and may enable improvement of the basic performance of the ant algorithm. Fig. 2.1 shows an image of the multiple ant colonies algorithm with four colonies.

2.2.1 The Formalizations of Multiple Ant Colonies Algorithm

According to the idea of colony-level interactions, the formalization of the multi-agent system based on multiple colonies ant algorithm is considered here. It is supposed that there are M ant colonies, and the formalization of agents are given as follows.

$$A^i = \langle \mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}, \mathbf{z}, f^i, g^i, h^i \rangle, (i = 1, 2, \dots, M) \quad (2.1)$$

where,

- $\mathbf{u}(t)$: Input from the environment
(i.e., the problem domain of given TSP) at time t
- $\mathbf{v}(t)$: Input vector from the pheromone space at time t
- $\mathbf{x}(t)$: Internal State at time t
- $\mathbf{y}(t)$: Output to the environment
(i.e., the solution for given TSP) at time t
- $\mathbf{z}(t)$: Output vector to the pheromone space at time t

$$\mathbf{x}(t+1) = f^i(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (2.2)$$

$$\mathbf{y}(t+1) = g^i(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (2.3)$$

$$\mathbf{z}(t+1) = h^i(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (2.4)$$

In the above formalizations, the function f is to update its internal state according to inputs from the environment, pheromone space and an internal state of mine. The function g is to decide its solution (or pre-solution), and h is to decide the output vector of pheromone. In that, the sets of $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ are same among agents but the function f, g, h are different because all colonies belong to same problem domain and use same pheromone space but decision policies are different each other.

It is supposed that there are m ant agents in each colony, and $\mathbf{v}_j^i(t)$ and $\mathbf{z}_j^i(t)$ are the input and output vector of j -th ant agent in i -th colony respectively.

$$P = \langle \mathbf{v}, \mathbf{X}, \mathbf{z} \rangle \quad (2.5)$$

where,

$$\mathbf{X}(t) = \{\mathbf{X}^1(t), \mathbf{X}^2(t), \dots, \mathbf{X}^M(t)\} \quad (2.6)$$

$$\mathbf{X}^i(t+1) = \rho \cdot \mathbf{X}^i(t) + \sum_j \mathbf{z}_j^i(t), (i = 1, 2, \dots, M) \quad (2.7)$$

$$\mathbf{v}_j^i(t+1) = \mathbf{X}(t) \quad (2.8)$$

$\mathbf{X}$ is the total pheromone vector of all colonies. Namely, all agent can sense the total pheromone vector including in not only own colony's pheromone but also other colonies' one. In the equation 2.7, ρ is non negative parameter concerned with the evaporation of pheromone, and usually set to less than equal to 1. The details of algorithm is described following subsection.

2.2.2 The Details of Multiple Ant Colonies Algorithm

Next, the brief definition of a TSP is described [17]. Given a set of n cities, and distances d_{ij} between two cities, the TSP is a problem of finding the minimal length of a closed tour that passes through each city just once. An example of a TSP is given by a graph (N, E) , where $N(|N| = n)$ is the set of cities and E is the set of edges between cities (a fully connected graph in Euclidean TSP). TSPs become symmetric in the case of $d_{ij} = d_{ji}$ and asymmetric in the case of $d_{ij} \neq d_{ji}$.

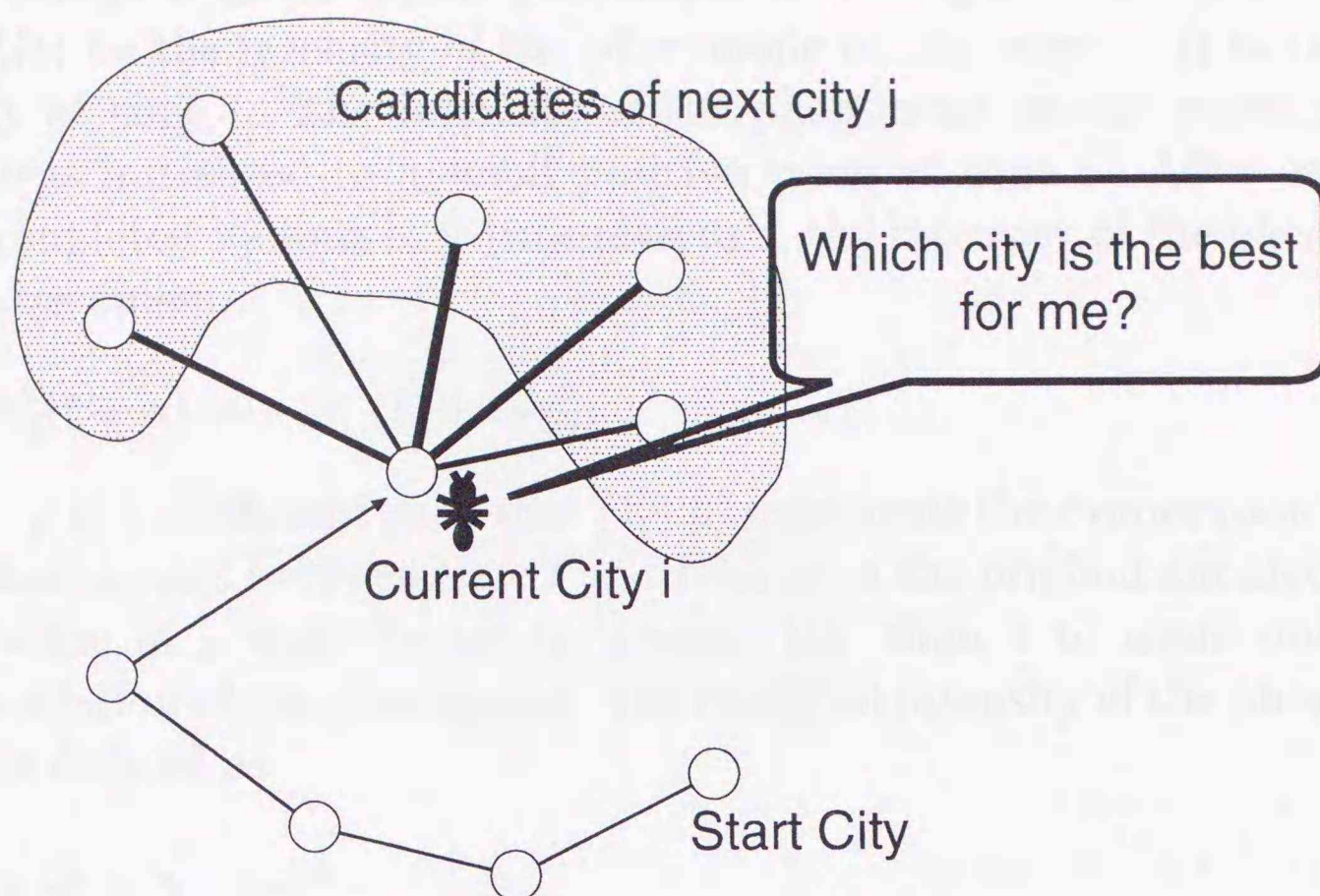


Figure 2.2: An image of moving ant agent on pheromone trail.

Next, I describe the multiple ant colonies algorithm. Let M and m be the number of colonies and the number of ant agents in each colony, respectively. The k -th ant agent in the h -th colony is denoted as the “ (h, k) ant”. The number of $M \cdot m$ ants search for TSPs at each time t . In the case of $M = 1$, this algorithm is the same as the original ant algorithm without colony-level interactions. Each ant agent is a simple agent with the following characteristics (with seeing Fig. 2.2):

- It continuously moves from one city i to the next city j in the time interval between t and $t + 1$; the next city j is chosen according to probability that is a function of the distance d_{ij} and intensity of the pheromone present on the connecting edge in each colony;
- It visits only unvisited city by itself in preceding times; this property, implemented to force the ant agents to make legal tours, holds until its tour is completed; then the ant memory is cleared to begin its tour again;
- After completing its tour (i.e., after visiting all cities), it lays pheromones of various intensities along each edges included in its tour; the intensity of the pheromone laid is decided as a function of the total distance of its tour.
- It behaves continuously until stop time t_{MAX} .

To extend a single colony pheromone to multiple-colony pheromones, let $\tau_{ij}^h(t)$ be the intensity of the pheromone on the edge (i, j) in the h -th colony at time t . The intensity of the pheromone on the edges in each colony is initialized to a small positive value at time 0. After each ant has completed its tour in n time intervals, the intensity of the pheromone $\tau_{ij}^h(t)$ becomes

$$\tau_{ij}^h(t+n) = \rho \cdot \tau_{ij}^h(t) + \Delta\tau_{ij}^h, \quad (2.9)$$

where ρ is a coefficient such that $(1-\rho)$ represents the evaporation rate of the pheromones between time t and $t+n$ as in the original ant algorithm. The value of ρ must be set to a value less than 1 to avoid unlimited accumulation of the pheromone. The modified intensity of the pheromone $\Delta\tau_{ij}^h$ is defined as

$$\Delta\tau_{ij}^h = \sum_{k=1}^m \Delta\tau_{ij}^{hk}, \quad (2.10)$$

where $\Delta\tau_{ij}^{hk}$ represents the intensity per unit of length of the edge (i, j) along which the pheromone is laid by ant (h, k) between time t and $t+1$, and this is given as

$$\Delta\tau_{ij}^{hk} = \begin{cases} Q/L^{hk} & \text{if ant } (h, k) \text{ uses edge } (i, j) \text{ on its tour} \\ & \text{between time } t \text{ and } t+n \\ 0 & \text{otherwise} \end{cases} \quad (2.11)$$

Here, Q is a constant and scarcely affects the behavior of the algorithm, as is the case in the original algorithm. L^{hk} is the total tour length of ant (h, k) ; i.e., the ant completing a tour with a shorter length is able to lay a larger intensity of the pheromone along edges included in its tour. The pheromones laid on probably better subtours would take large value as a result of whole behavior.

The new transition probability from city i to city j for ant (h, k) including colony-level interactions is defined as

$$p_{ij}^{hk}(t) = \begin{cases} \frac{\tau_{ij}^h(t)}{\sum_{l \notin \text{tabu}^{hk}(t)} \tau_{il}^h(t)} & \text{if } j \notin \text{tabu}^{hk}(t) \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

$$\pi_{ij}^h(t) = \left\{ \prod_{l=1}^M [\tau_{ij}^l(t) + C(h)]^{\alpha(h,l)} \right\} \cdot [\eta_{ij}]^{\beta(h)} \quad (2.13)$$

$$\eta_{ij} = 1/d_{ij} \quad (2.14)$$

Here, $\text{tabu}^{hk}(t)$ indicates the tabu list of ant (h, k) . This list consists of cities that have already been visited cities until time t , and the ant is forbidden to choose such cities repeatedly. This is reset to ϕ when the ant visits all cities and completes its tour, so the ant agents can usually keep legal tours. $\pi_{ij}^h(t)$ means the degree of preference for an edge connected to city j . If this value is large, ant (h, k) tends to choose city j as the next one to visit. The parameter $\alpha(h, l)$, which is must be set in advance, determines the type of pheromone effect and the degree of influence from colony l (see Fig. 2.3). If $\alpha(h, l)$ is set to a positive value, i.e., a positive pheromone effect, ant (h, k) tends to prefer schemata indicated by the pheromone in the l -th colony. Conversely, if $\alpha(h, l)$ is negative, i.e., a negative pheromone effect, ant (h, k) tends to avoid choosing edges that have been frequently used in the l -th colony. If the value of $\alpha(h, l)$ is zero, the pheromone in the l -th colony has no effect on the behavior of ants in the h -th colony. The absolute value of $\alpha(h, l)$ indicates the degree of the pheromone effect. The effect becomes stronger as the value increases and becomes weaker as the value decreases. In particular, $\alpha(h, h)$, the effect from itself, must be positive in order to cause a positive feedback in the colony. $C(h)$ represents the degree of insensitivity to the pheromones, and this parameter is introduced so that the ant agents will avoid overly sensitive effects towards the pheromones near zero. η_{ij} is used as a greedy heuristic term that makes the ants prefer nearby cities, and it seems to be effective for TSPs. The parameter $\beta(h)$ represents how important η_{ij} is for the ants in the h -th colony. Fig. 2.4 shows the structure of the ant agents with the control parameters, and Fig. 2.5 shows the procedure for the proposed algorithm.

The computational complexities from step 1 to step 5 are $O(Mn^2 + M^2n^2)$, $O(Mmn)$, $O(Mmn^2)$, $O(Mn + Mmn^2)$ and $O(Mn^2 + M^2n^2)$, respectively. Here, I suppose $M \ll m$ as the setting of the algorithm, and the computational complexity of the proposed algorithm is of $O(MTmn^2)$ where T is the number of cycles in the algorithm. The order of original ant algorithm is $O(Tmn^2)$ given by the case of $M = 1$ in the above. Therefore, the order of multiple ant colonies algorithm corresponds to the order of the ant algorithm in M times running. Namely, if it is showed that the multiple ant colonies algorithm has better performance than the ant algorithm in M times running, the proposed algorithm is superior to the original ant algorithm.

In this algorithm, more parameters than those in the original algorithm

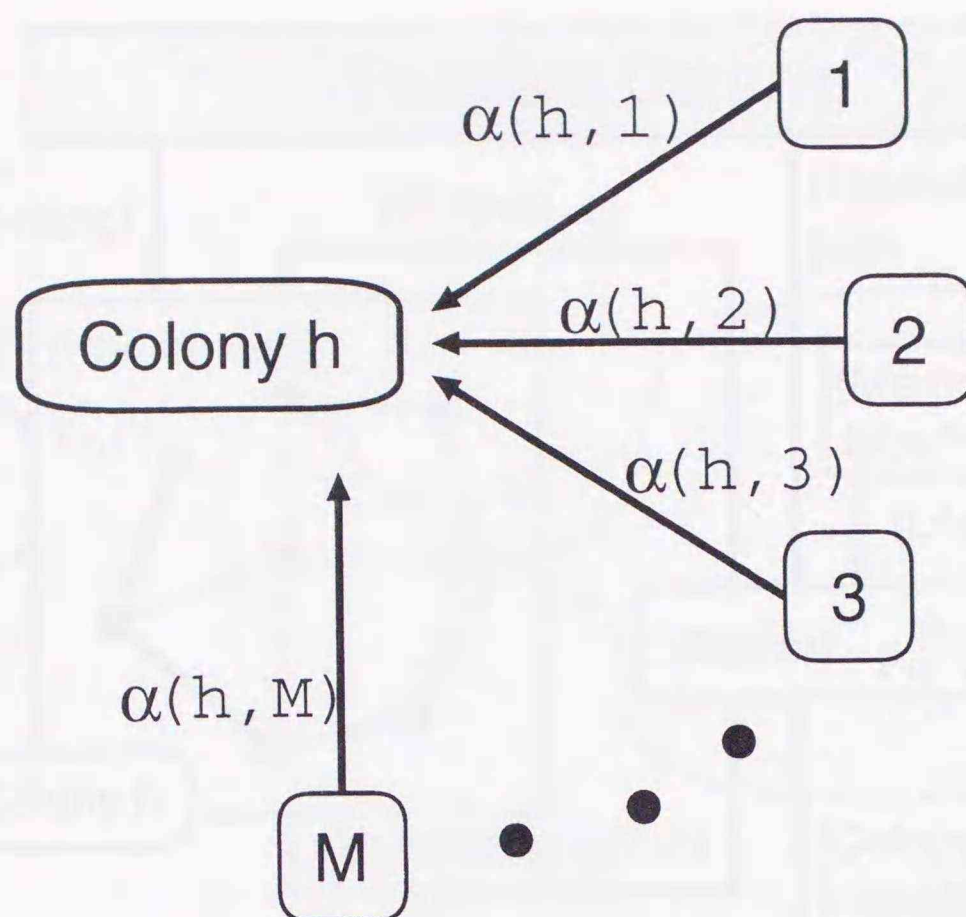


Figure 2.3: The effectiveness of parameter α .

must be determined in advance, but trials of various relationships among colonies can be done by controlling these parameters. The management of colony structure is very important in the case of multiple colonies. In addition, I don't focus on exquisite agent strategies or heuristics specialized to the solved problem but a very simple heuristic in this proposal. The objective of this study was how to extend the original ant algorithm to one for multiple ant colonies in order to improve the basic performance of the algorithm, and such specializations will be discussed in a future paper.

2.3 Computer Experiments

I performed some computer experiments to investigate the basic performance and the characteristics of the proposed algorithm compared with the original algorithm. The first experiment was carried out on a TSP with a form like a doughnut, and interesting results were obtained. In the second experiment, the pheromone effects based on colony-level interactions with eil101 benchmark TSP in TSPLIB were investigated. In the third experiment, the performance of the proposed algorithm was compared with that of the original one for some TSPs in TSPLIB. The values of parameters in each experiment were determined by reference to


```

1. Initialize
   Set  $t := 0$ 
   For  $h := 1$  to  $M$  do
     For every edge  $(i, j)$ 
       Set  $\tau_{ij}^h(t) := \text{initialvalue}$ 
2. Set all ants at the starting city
   Set  $s := 0$ 
   Empty all tabu lists
   For  $h := 1$  to  $M$  do
     For  $k := 1$  to  $m$  do
       Place the starting city of the  $(h, k)$  ant, and insert it in  $\text{tabu}^{hk}(t)$ 
3. Repeat until all cities have been visited
   For  $s := 1$  to  $n - 1$  do
     For  $h := 1$  to  $M$  do
       For  $k := 1$  to  $m$  do
         Choose the city  $j$  to move to, with probability  $p_{ij}^{hk}(t + s)$ 
           given by equation (2.12)

         Move the  $(h, k)$  ant to the city  $j$ 
         Insert the city  $j$  in  $\text{tabu}^{hk}(t + s)$ 
4. Compute changes of the pheromones in every colony
   For  $h := 1$  to  $M$  do
     For  $k := 1$  to  $m$  do
       Move the the  $(h, k)$  ant to the starting city
       Compute tour length  $L^{hk}$  by the  $(h, k)$  ant
       Update the shortest tour found
   For  $h := 1$  to  $M$  do
     For every edge  $(i, j)$ 
        $\Delta\tau_{ij}^h := 0$ 
       For  $k := 1$  to  $m$  do
         Compute  $\Delta\tau_{ij}^{hk}$  according to equation (2.11)
          $\Delta\tau_{ij}^h := \Delta\tau_{ij}^h + \Delta\tau_{ij}^{hk}$ 
5. Update the pheromones in every colony
   For  $h := 1$  to  $M$  do
     For every edge  $(i, j)$ 
        $\tau_{ij}^h(t + n) := \rho \cdot \tau_{ij}^h(t) + \Delta\tau_{ij}^h$ 
   Set  $t := t + n$ 
6. Check stop time
   If  $(t < t_{MAX})$ 
     then
       Goto step 2
     else
       Print shortest tour
       Stop

```

Figure 2.5: The algorithm of the multiple ant colonies.

M	m	ρ	Q	$\tau_{ij}^h(0)$	t_{MAX}
3	40	0.65	200	5	240

	1	2	3
$\alpha(1, -)$	2	0	0
$\alpha(2, -)$	-0.5	2	0
$\alpha(3, -)$	-0.2	-0.2	2

$\beta(1)$	5
$\beta(2)$	5
$\beta(3)$	5

$C(1)$	1
$C(2)$	2
$C(3)$	20

Table 2.1: Values of parameters used in experiment 1.

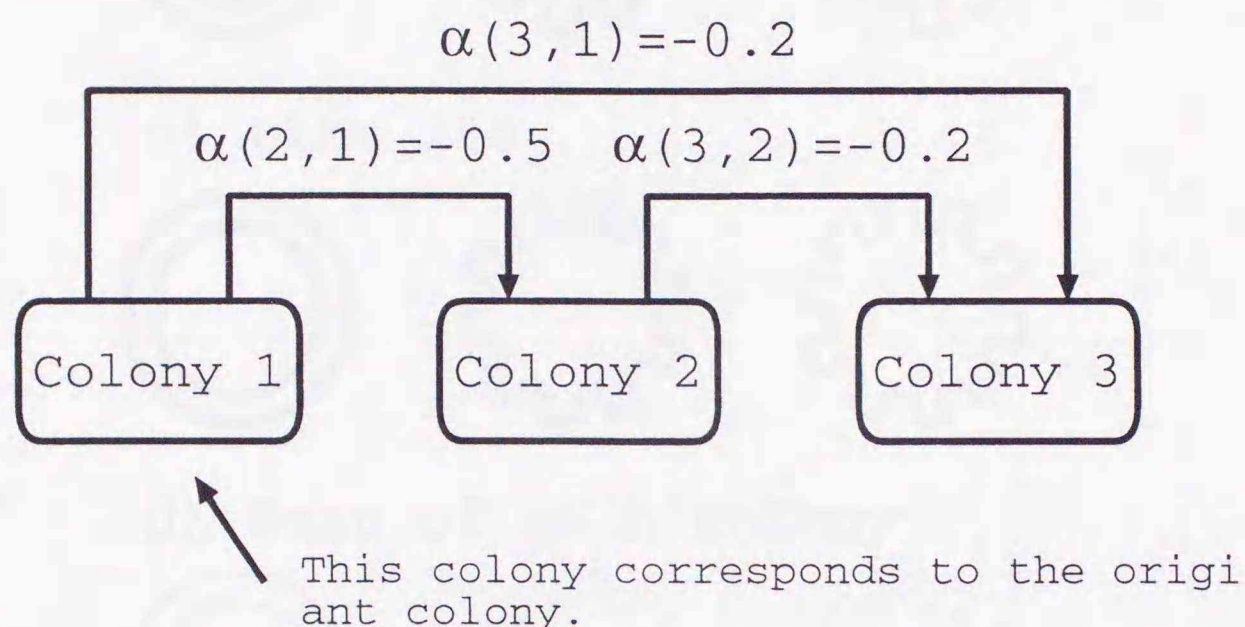


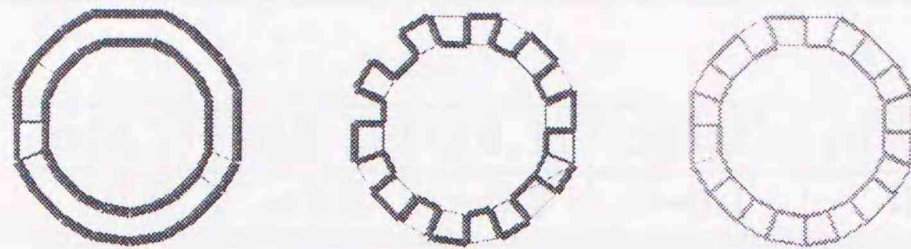
Figure 2.6: The colony structure used in the experiment 1.

be greater. This indicates that the existence of unsearched regions would be a characteristic of the original ant algorithm with a simple heuristic. Thus, care should be taken in using heuristics in families of ant algorithms with conscious of problem features.

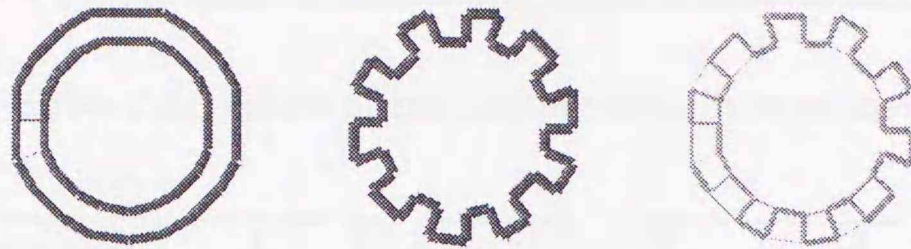
I applied the proposed algorithm to this TSP using the parameters listed in Tab. 2.3.1. With these settings, only negative pheromone effects were used as colony-level interactions. Like Fig. 2.6, the first colony was not affected by other colonies, and the ants in the first colony could search at will. The colony was affected only by the first colony, and the third colony was affected by the first and second colonies (which can be called a *hierarchical colony structure*). $C(3)$ took a larger value than $C(1)$ and $C(2)$ because the third colony would be strongly influenced by two colonies.

Interesting results are shown in Fig. 2.7. Figures 4(a)-(c) represent the transition process of the pheromones at each time. The left, middle

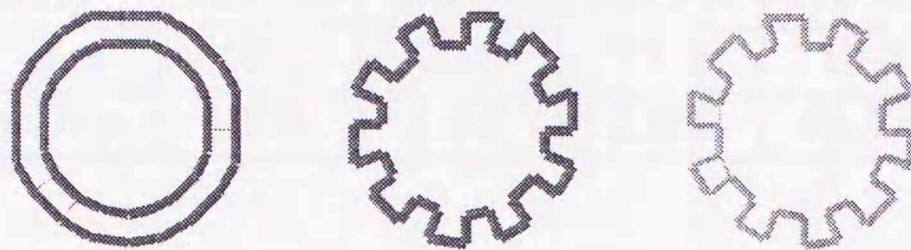
(a) time 80



(b) time 160



(c) time 240



(d) Best of each colony

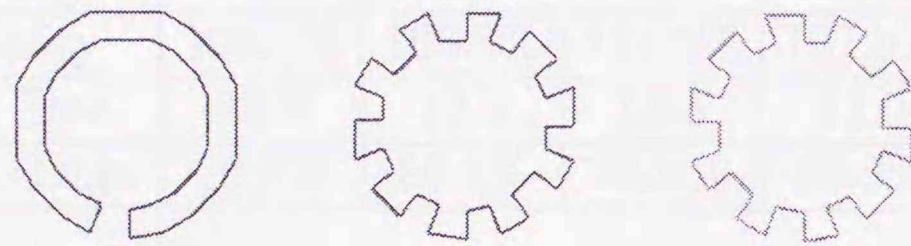


Figure 2.7: The transition of the pheromone and the best solutions obtained in experiment 1.

and right figures are related to the intensities of pheromones in the first, second and third colonies respectively. The width of each line represents the intensity of the pheromone on each edge. In (a), ant agents in the first colony put effective pheromone on "C" type tours rapidly, then ant agents in the second colony searched different regions from those searched in the first due to the negative pheromone effect. Finally in (c), ant agents in the third colony succeeded to specify different regions from those in the first and second colonies. Fig. (d) shows the best solutions obtained in each colony through this experiment. The first colony found a solution with a distance of 218.1, and a global optimal solution with a distance of 209.5 was found by the second and third colonies. Namely, global optimal

M	m	ρ	Q	$\tau_{ij}^h(0)$	t_{MAX}	$\beta(h)$	$C(h)$
10	101	0.65	200	5	303000	1.5	1

$\alpha(2, 1)$	$\alpha(3, 1)$	$\alpha(4, 1)$	$\alpha(5, 1)$	$\alpha(6, 1)$
-1.0	-0.5	-0.2	-0.1	0
$\alpha(7, 1)$	$\alpha(8, 1)$	$\alpha(9, 1)$	$\alpha(10, 1)$	$\alpha(h, h)$
0.1	0.2	0.5	1.0	1.5

Table 2.2: Values of parameters used in experiment 2.

Colony No	1	2	3	4	5
Average	691.9	896.0	881.5	724.3	693.9
Quality	107.7%	139.5%	137.2%	112.8%	108.0%
Std. Dev.	13.1	20.6	17.9	15.5	20.0
Ratio Dup.	100%	1.3%	2.5%	40.3%	61.4%

Colony No	6	7	8	9	10
Average	685.5	685.9	689.1	693.1	693.7
Quality	106.7%	106.8%	107.3%	107.9%	108.0%
Std. Dev.	15.7	12.5	13.6	12.0	10.1
Ratio Dup.	76.6%	86.4%	88.8%	88.9%	88.4%

Table 2.3: Results of experiment 2.

solutions that could not be found by the first colony were found by the second and third colonies, and three local and global optimal solutions could be found at same time. These results are very interesting, although I admit that this is an adhock case.

2.3.2 Experiment 2

Experiment 2 was performed to confirm the general properties of colony-level interactions with each setting of parameter α for eil101, 101 cities symmetric benchmark in TSPLIB. For this confirmation, I prepared ten colonies with one-way interactions, in which the first colony was not affected by other colonies, and other colonies were affected by only the first colony with various values of α . The parameter values used in this experiment are listed in Tab. 2.3.2 and Fig. 2.8. In addition, the values

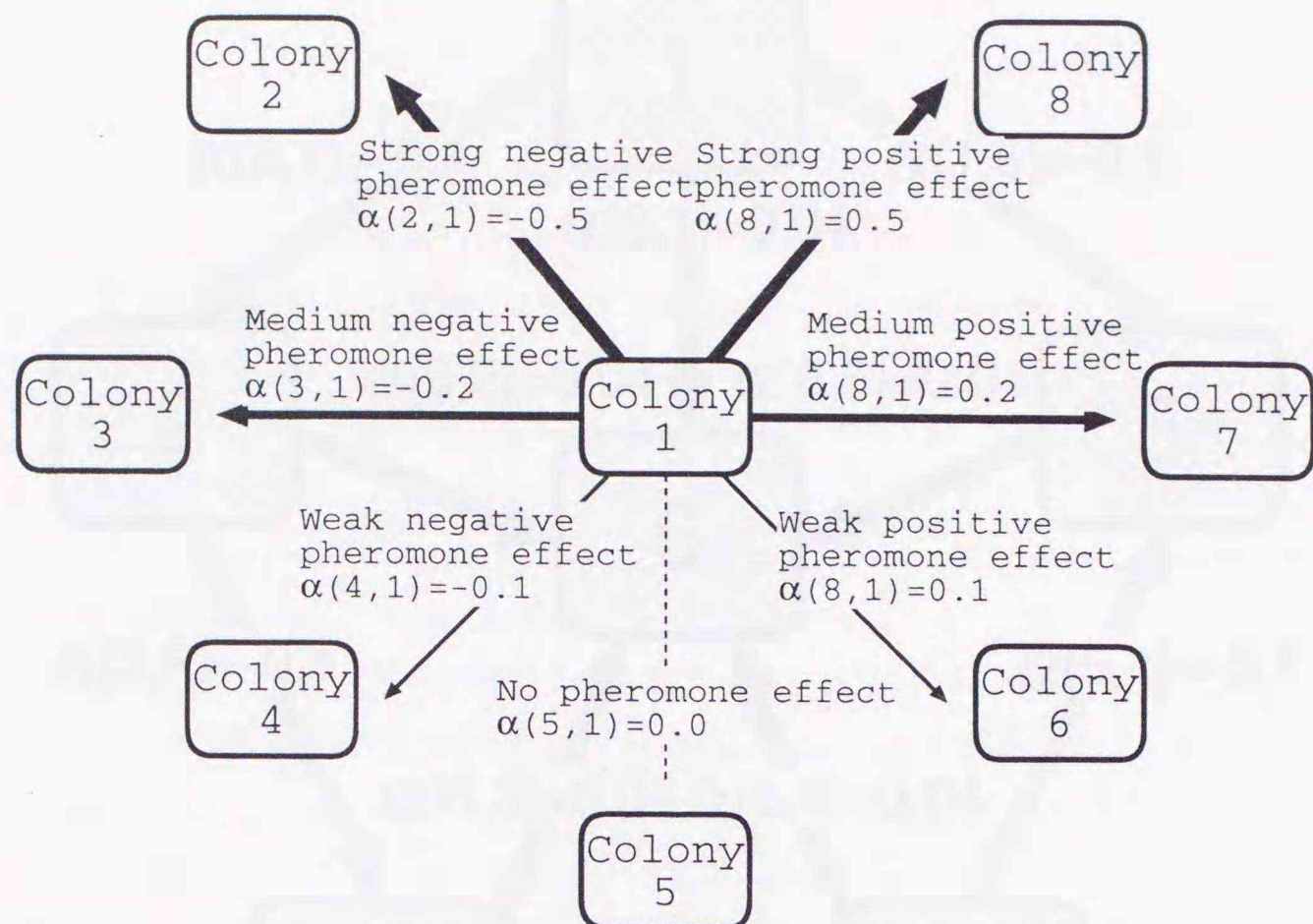
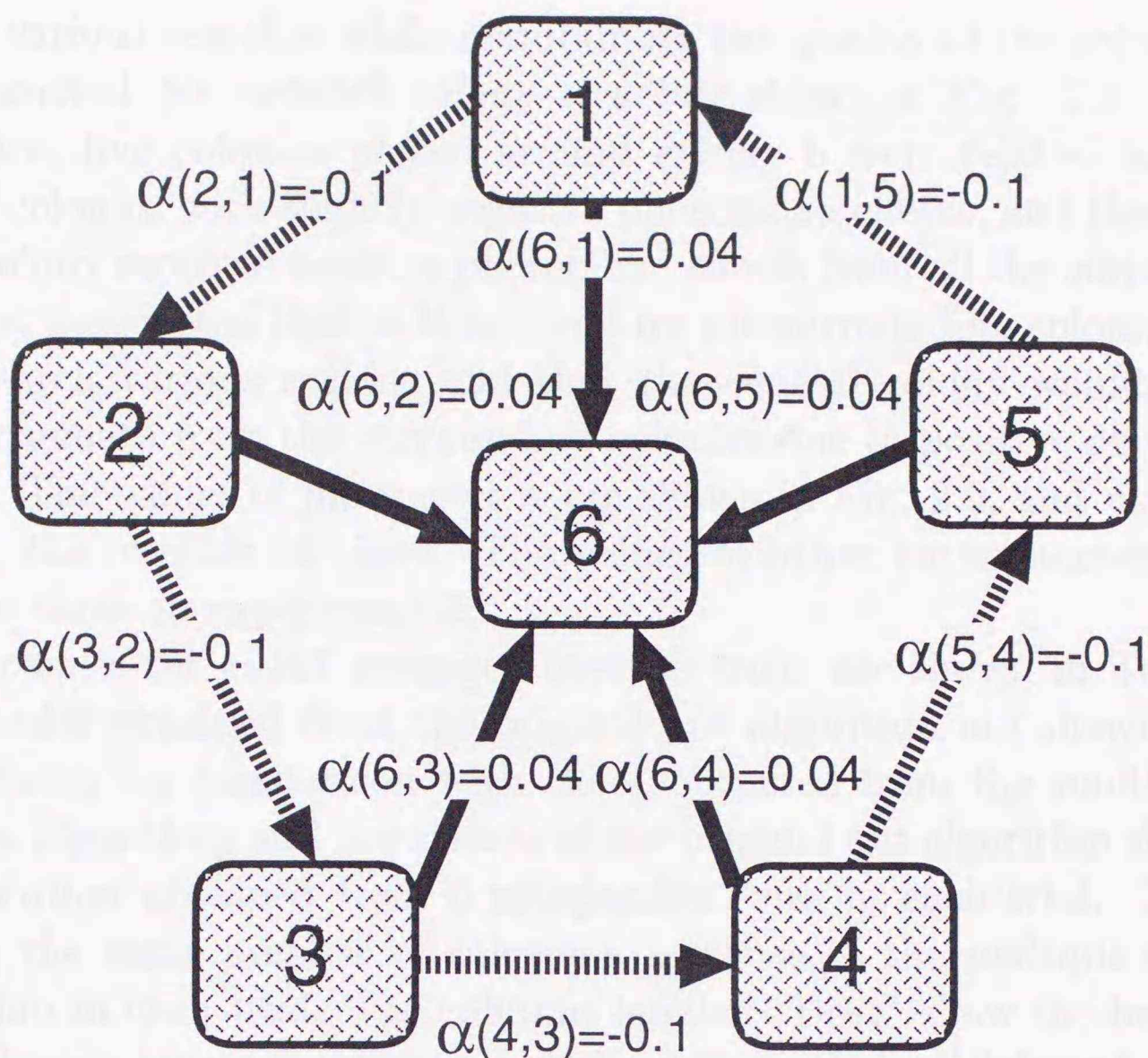


Figure 2.8: The colony structure used in the experiment 2.

of $\alpha(h,l)$, which are not listed in the table, were set to zero. The second, third, fourth and fifth colonies would receive negative pheromone effects from the first, while the sixth colony was neutral, and the seventh, eighth, ninth and tenth colonies would receive positive pheromone effects from the first.

Tab. 2.3.2 shows the results averaged over 40 trials with the average distance of the best solution, the quality of a known global optimal solution (642.3), the standard deviation, and the ratio of duplicate subtours compared with the best solution of the first colony. The ratio of duplicate subtours between the first and the sixth colonies, i.e., a pair with no interaction, was 76.6%, which indicates that the original ant algorithm has a tendency to always generate solutions with about 76.6% of the same subtours in each trial. The ratios between the first and second colonies and between the first and third colonies were very low, but the second and third colonies found worse solutions in strongly avoiding subtours well used by the first colony. The fifth colony which received a slightly negative pheromone effect, generated solutions of almost the same quality as those of the first colony. The ratio between the first and fifth colonies was lower than between the first and sixth colonies, and different regions were searched by the fifth colony. The seventh, eighth, ninth and tenth colonies had solutions of about the same quality with higher ratios; that



 Negative Pheromone Effect
 Positive Pheromone Effect

Figure 2.9: A network colony structure with six colonies. $\alpha(h, h) = 1.5$. The values of $\alpha(h, l)$ not shown were all set to zero.

is, these colonies searched similar regions due to the positive pheromone effects from the first colony.

I showed that it is possible to control the searching behavior of each colony with colony-level interactions based on parameter α . Strong negative pheromone effects are not effective and lead to solutions of inferior quality, while a slight negative pheromone effect, such as setting $\alpha(h, l)$ to 0.1, seems to be useful for various searches while maintaining the quality of the solution.

2.3.3 Experiment 3

In experiment 2, the negative pheromone effects were found to be use-

ful for various searches while maintaining the quality of the solution, so I constructed *the network colony structure* shown in Fig. 2.9. In this structure, five colonies placed around colony 6 were related to neighboring colonies with slightly negative pheromone effects, and the central sixth colony received positive pheromone effects from all the surrounding colonies. I expected that in this structure the surrounding colonies could each search various regions and that the central colony would receive good schemata from the surrounding colonies due to positive pheromone effects. The values of parameter α are shown in Fig. 2.9, and m was set to 101, the number of cities. The values of other parameters were the same as those in experiment 2.

The results for eil101 averaged over 20 trials are shown in Tab. 2.4. The results obtained from the original ant algorithm are shown in the last column for comparison with those obtained from the multiple ant colonies algorithm, and the results of the original ant algorithm show the best solution obtained from 6 independent run in each trial. That is, it gave the same number of solutions as those of the multiple colonies algorithm in each trial. The columns labeled "Best" show the best solutions obtained from 6 colonies in each trial. In the multiple ant colonies algorithm, the sixth colony discovered the best solutions 6 times and showed the best performance of all colonies, as was expected. The sixth colony succeeded to gather good schemata. Moreover, the best solutions obtained by the multiple ant colonies algorithm were of better quality than those obtained by the original ant algorithm. Examples of the best solutions obtained by each colony are shown in Fig. 2.11. It is clear from those examples that the colony-level interactions and the network colony structure are very effective.

Next, two experiments 3(2) and 3(3) were performed in order to confirm the effectiveness of negative pheromone effects in the network colony structure. In experiment 3(2), the values of parameters were the same as those in Fig. 2.9 except having the positive pheromone effects instead of the negative effects, i.e., $\alpha(2, 1) = \alpha(3, 2) = \alpha(4, 3) = \alpha(5, 4) = \alpha(1, 5) = 0.1$. The settings of experiment 3(3) were also the same as those in Fig. 2.9 except having no effects, i.e., $\alpha(2, 1) = \alpha(3, 2) = \alpha(4, 3) = \alpha(5, 4) = \alpha(1, 5) = 0$. The results of both experiments are shown in Tab. 2.3.3 and 2.6. In both cases, the surrounding colonies (No. 1-5) could generate solutions with similar quality to the experiment 2, but the center colony (No. 6) could generate worse qualities. Moreover, the best solutions of both cases had similar quality to the original ant algorithm. These results indi-

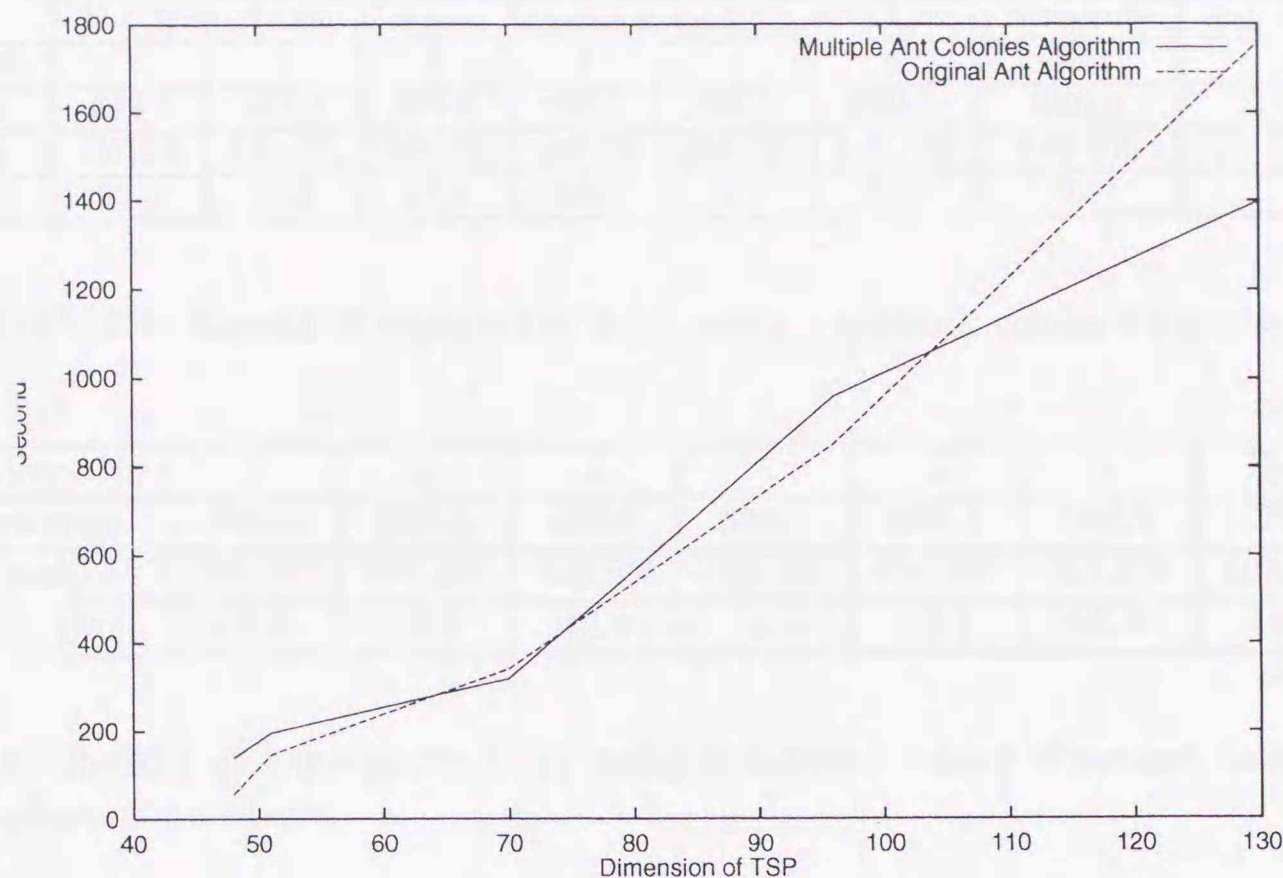


Figure 2.10: The time to obtain the best solution and the dimension of solved TSP.

cate that the network colony structure without the negative pheromone effects tend to search wasteful regions without varieties. Namely, the surrounding colonies behaved similarly, and the center colony couldn't gather effective schemata concerned with the search space. Therefore, I can say that the network colony structure with the negative pheromone effects are very effective against the original ant algorithm.

Tab. ?? shows the results averaged over 20 trials for other TSPs in TSPLIB with the same colony structure used in experiment 3. In all cases, the multiple ant colonies algorithm showed better performance than did the original ant algorithm. In addition, "Time Avg." columns in the table ?? show the time to find best solutions by Pentium III 500Mhz IBM-PC computer based on Linux Operating System. This result indicates the use of the multiple ant colonies algorithm is effective with better qualities against the original one because the time of both algorithms are almost the same. The time to find the best solutions increases linearly as the size of the problems increases. and it shows the possibility to apply the proposed algorithm to large TSPs (see Fig. 2.10).

However, the qualities of solution obtained by both algorithms do not seem to be better than the quality of solutions obtained by approximate methods, such as Genetic Algorithm, Simulated Annealing, and Tabu Search, due to a lack of careful adjustment to the parameters for solving

	The Multiple Ant Colonies Algorithm with Network Colony Structure							Ant Algorithm
Colony No	1	2	3	4	5	6	Best	Best
Average	692.2	683.6	695.5	685.7	689.7	680.1	665.4	676.6
Quality	107.8%	106.4%	108.3%	106.7%	107.4%	105.9%	103.6%	105.3%
Std. Dev.	21.5	18.9	15.3	19.4	15.7	11.0	9.1	8.0

Table 2.4: Results of experiment 3 (1) using a network colony structure.

Colony No	1	2	3	4	5	6	Best
Average	703.6	688.6	686.9	685.9	696.1	692.9	677.5
Quality	109.5%	107.2%	106.9%	106.8%	108.3%	107.8%	105.5%
Std. Dev.	21.8	16.6	12.8	14.2	13.1	17.8	11.4

Table 2.5: Results of experiment 3 (2) using a network colony structure based on the positive pheromone effects.

these TSPs. In particular, the ant agents in both algorithms generated tours that included many cross-subtours in each trial. For example, this weakness in TSPs could be broken to regard solutions that are generated by the ant agents as genotype solutions and then these genotypes could be translated into phenotype solutions in such a way as to apply operations such as 2-opt to each cross-subtour point; however, this idea was not used in this study for only comparisons between the proposed algorithm and the original one.

2.4 Discussion

Generally, approximate methods with hand-crafted heuristics tend to have weak points that search regions would be specified by properties of used heuristics but these heuristics often make search efficient. The original ant algorithm also has this weakness, and I should therefore handle heuristics with conscious of problem features. We consider that the

Colony No	1	2	3	4	5	6	Best
Average	698.5	684.2	689.0	686.8	695.9	691.8	673.4
Quality	108.7%	106.5%	107.3%	106.9%	108.3%	107.7%	104.8%
Std. Dev.	17.5	10.8	13.6	10.7	11.8	21.1	9.1

Table 2.6: Results of experiment 3 (3) using a network colony structure based on no effects.

Problem		Multiple Ant Colonies Algorithm			
Name	Dimension	Best Avg.	Quality	Std. Dev.	Time Avg.
att48	48	34972.1	104.3%	222.2	139.6 sec
eil51	51	448.3	104.3%	4.8	196.6 sec
st70	70	710.9	104.8%	8.3	319.2 sec
gr96	96	543.9	106.2%	2.9	956.5 sec
ch130	130	6348.6	103.9%	43.8	1398.4 sec

Problem		Ant Algorithm			
Name	Dimension	Best Avg.	Quality	Std. Dev.	Time Avg.
att48	48	35169.2	104.9%	232.5	56.1 sec
eil51	51	457.7	106.5%	2.6	146.7 sec
st70	70	727.9	107.3%	15.9	341.1 sec
gr96	96	550.3	107.4%	14.1	848.8 sec
ch130	130	6423.1	105.1%	81.6	1757.2 sec

Table 2.7: Results for other TSPs.

proposed multiple ant colonies algorithm can overcome this weakness by the introduction of colony-level interactions without requiring exquisite heuristics. The negative pheromone effects enable the maintenance in the search process, and colonies can easily exchange good schemata with each other due to the positive pheromone effects. In our proposals, I didn't dare to include well tuned up ant agent strategies for the sake to extend a basic performance of the ant algorithm, and the performance will be still better with nice strategies using in the ant algorithm families.

Moreover, the proposed algorithm can be applied to a parallel computing, such as cluster computing, without the need to consider synchronization of operations, and it would have better performance than that of the original ant algorithm simply running parallel. However, further investigation of the construction of colony structures and determination of interactions between colonies is needed. It is desirable for the algorithm to autonomously, adaptively, and dynamically organize its interactions in accordance with states, although I decided as hand-crafted in the experiments, and the interactions worked effectively in a network colony structure. One idea for the control of interactions is to introduce a super ant agent, i.e., the queen ant agent, that observes the behavior of the ant agents and manages the colony-level interactions. The queen ant

agent could be designed to focus on uni-path behavior, i.e., the situation in which all ant agents make the same tour [5]. It will be effective that the queen ant agent operates the changing mechanisms of the parameters according to the whole behaviors in its colony.

2.5 Conclusions

In this chapter, I have extended the original ant algorithm to a multiple ant colonies algorithm with introducing the colony-level interactions such as positive and the negative pheromone effects. The negative pheromone effects enable maintenance of variation in the search process while maintaining quality in the solutions, and colonies are able to exchange good schemata for solving a problem due to the positive pheromone effects. Through computer experiments, I showed that the proposed algorithm has better performance than the original ant algorithm with almost the same agent strategies. In a future study, I will attempt to apply the proposed algorithm to cluster computing with well-defined heuristics or strategies and also to huge combinatorial optimization problems such as a TSP problem involving several thousand cities.

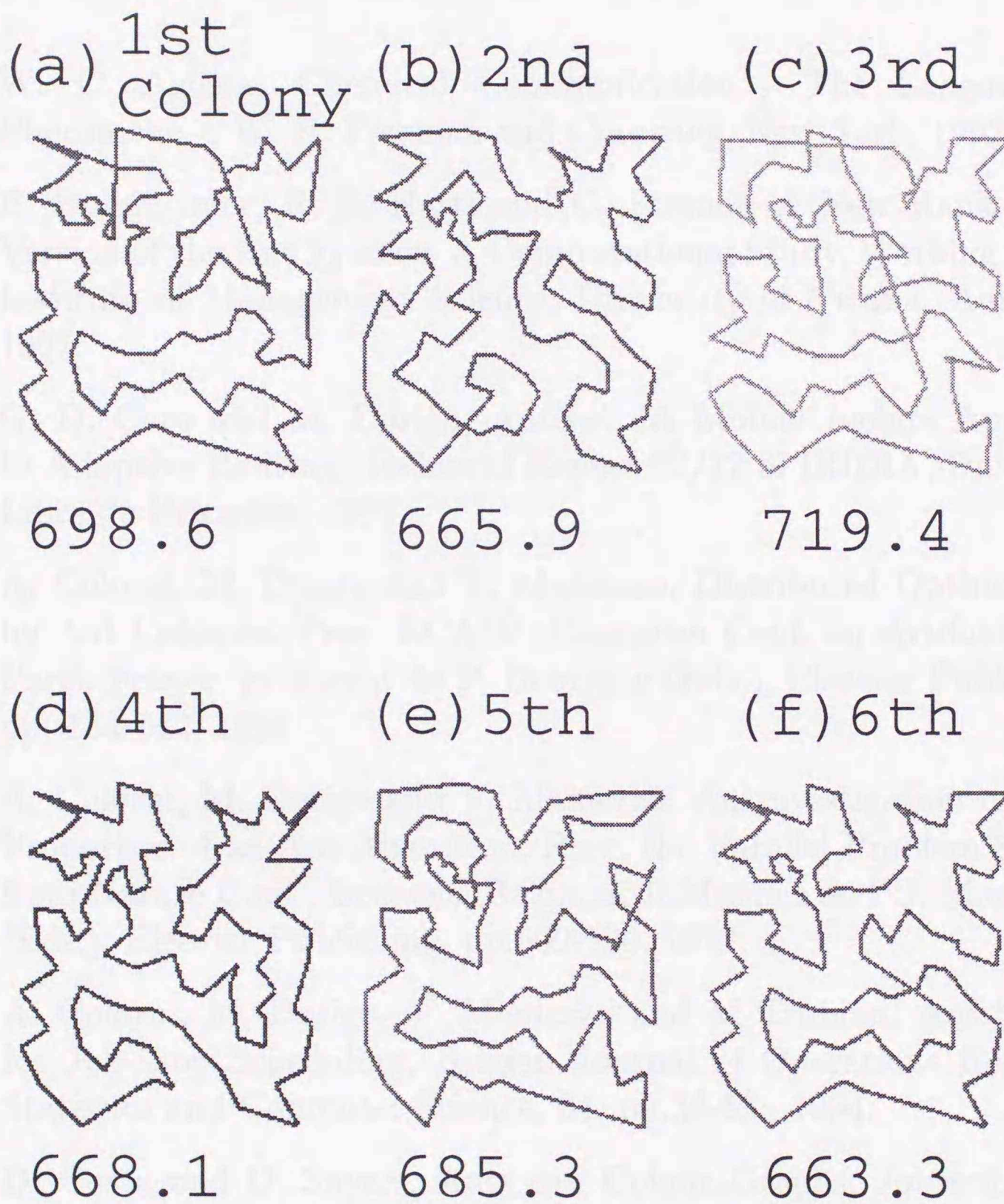


Figure 2.11: Examples of best solutions obtained by each colony in experiment 3 (1).

Bibliography

- [1] W. C. Agosta, Chemical Communication - The Language of Pheromone -, W. H. Freeman and Company, New York, 1992.
- [2] B. Bullnheimer, R. F. Hartl and C. Strauss, A New Rank Based Version of the Ant System: A Computational Study, Working paper, Institute of Management Science, University of Vienna, Australia, 1997.
- [3] G. D. Caro and M. Dorigo, AntNet: A Mobile Agents Approach to Adaptive Routing, Technical Report 97/12 of IRIDIA, Université Libre de Bruxelles, 1997.
- [4] A. Coloni, M. Dorigo and V. Maniezzo, Distributed Optimization by Ant Colonies, Proc. ECAL91-European Conf. on Artificial Life, Paris, France, F. Varela and P. Bourgine (Eds.), Elsevier Publishing, pp. 134-142, 1991.
- [5] A. Coloni, M. Dorigo and V. Maniezzo, An Investigation of Some Properties of an Ant Algorithm, Proc. the Parallel Problem Solving from Nature Conf., Brussels, Belgium, R. Manner and B. Manderick (Eds.), Elsevier Publishing, pp. 509-520, 1992.
- [6] A. Coloni, M. Dorigo, V. Maniezzo and M. Trubian, Ant System for Job-shop Scheduling, Belgian Journal of Operations Research, Statistics and Computer Science, 34, pp. 39-53., 1994.
- [7] D. Costa and D. Snyers, Ants can Colour Graphs, Journal of the Operational Research Society, 48, pp. 295-305, 1997.
- [8] M. Dorigo, V. Maniezzo and A. Coloni, The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13, 1996.
- [9] M. Dorigo and L. M. Gambardella, Ant Colonies for the Travelling Salesman Problem, BioSystems 43, Elsevier, pp. 73-81, 1997.

- [10] L. M. Gambardella and M. Dorigo, Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem, Proc. ML-95, Twelfth International Conf. on Machine Learning, pp. 252-260, 1995.
- [11] H. Kawamura, M. Yamamoto, T. Mitamura, K. Suzuki and A. Ohuchi, Cooperative Search Based on Pheromone Communication for Vehicle Routing Problems, Publication in the IEICE Transactions, Fundamentals, Vol. E81-A, No. 6, pp. 1089-1096, June, 1998.
- [12] P. Kuntz, P. Layzell and D. Snyers, A Colony of Ant-like Agents for Partitioning in VLSI Technology, Proc. of Forth European Conf. on Artificial Life, Phil Husbands and Inman Howey (Eds.), pp.417-424.
- [13] V. Maniezzo, A. Coloni and M. Dorigo, The Ant System Applied to the Quadratic Assignment Problem, Technical Report 94/28 of IRIDIA, Université Libre de Bruxelles, 1994.
- [14] T. Mikami, S. Mikami and M. Wada, Ant-like Collective Sorting for Massive Data Analysis - A Quantitative Study -, Proc. Intelligent Autonomous Systems (IAS-5), Y. Kakazu, M. Wada and T. Sato (Eds.), pp. 686-693, 1998.
- [15] H. Mühlenbein, M. Schomisch and J. Born, The Parallel Genetic Algorithm as Function Optimizer, Proc. the Fourth International Conf. on Genetic Algorithm, Richard K. Belew and Lashon B. Booker (Eds.), pp.271-278, 1991.
- [16] M. Nakamura and K. Kurumatani, Formation Mechanism of Pheromone Pattern and Control of Foraging Behavior in an Ant Colony Model, Proc. the Fifth International Workshop on the Synthesis and Simulation of Living Systems, Christopher G. Langton and Katsunori Shimohara (Eds.), The MIT Press, pp.67-74, 1997.
- [17] G. Reinelt, The Traveling Salesman - Computational Solutions for TSP Applications, Lecture Notes in Computer Science 840, G. Goos and J. Hartmanis (Eds.), Springer-Verlag, 1994.
- [18] T. Stützle and H. Hoos, The Max-Min Ant System and Local Search for the Traveling Salesman Problem, Proc. IEEE International Conf. of Evolutionary Computation (ICEC'97), pp. 308-313, 1997.
- [19] D. Subramanian, P. Drushcel and J. Chen, Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks, Proc. the

Fifteenth International Joint Conf. on Artificial Intelligence (IJCAI-97), pp.832-838, 1997.

- [20] K. Suzuki and A. Ohuchi, Reorganizations of Agents with Pheromone Style Communication in Multiple Monkey Banana Problem, Proc. Intelligent Autonomous Systems (IAS-5), Y. Kakazu, M. Wada and T. Sato (Eds.), pp. 615-622, 1998.
- [21] M. Yamamoto, H. Kawamura, K. Suzuki and A. Ohuchi, Collective Approach for Combinatorial Optimization Problems, Proc. The International Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC98), pp.1479-1482, 1998.

Chapter 3

Distributed Optimization Algorithm Based on Pheromone Style Communication for Vehicle Routing Problem

3.1 Introduction

Collective approaches are expected to perform the flexibility and adaptability for solving the problems. Several collective approaches are studied in the area of multi-agent systems, evolutionary computations and artificial life. I think that these approaches are expectant of solving larger and more complicated combinatorial optimization problems. The major issue in these approaches is how to construct the cooperative relation between the distributed autonomous operators as the problem solvers. To construct the cooperative relation, the designs of the communication between the operators occupy important factor. Because the design of the communication dominates the abilities of the operators and the total performance of the problem solving systems, the design of the communication is expected to have the adaptability according to the problem domains and the behaviors of the operators.

In this chapter, I focus on the communication methods of living creatures which are used to emerge the structured behaviors. Especially, the pheromones that are the chemical substances are widely used by several creatures. For example, if the real ants catch a certain stimulus, e.g., they encounter enemies, discover feed, and so on, the ants reflectively generate the specific type of pheromone into the environment. According to the types of generated pheromones, the ants who receiving it reflectively take the specific types of actions, e.g., attacking the enemies, correcting the feed, and so on. These specific actions caused by the pheromones will

continue until the pheromones are disappeared in the environment [1][9]. It is considered that real ants make the complex community using several types of pheromones as communication.

Studies on mechanisms of ant colonies and pheromone communications are worthy of notice for development of artificial intelligence. The characteristic behaviors of real ants have been widely researched in various fields. Nakamura, Suzuki and Mikami studied on multiple agent systems based on behaviors of real ants and the pheromone style communication for these organization abilities [24][3][28][22]. For optimizations, Subramanian proposed distributed routing algorithms with simple biological ants for routing in packet-switched communication networks [4][27]. Caro also proposed mobile agents approach based on the AntNet for adaptive communications networks routing [11]. Kawamura and Yamamoto constructed optimization algorithms using the pheromone style communications for Vehicle Routing Problems and Nurse Scheduling Problems respectively [19][29]. The ant algorithm, originally proposed by Colormi et al in 1991 for Traveling Salesman Problems (TSPs) [5][12][13][14][16][17], is also one of optimization algorithm inspired from the analogy of foraging behaviors and interactions in real ant colonies.

Therefore, I propose the framework of the distributed scheduling algorithm based on multi-agent system using pheromone style communication for solving the Vehicle Routing Problems (VRPs). In proposed algorithm, each agent autonomously and distributively plan its partial schedule using pheromone style communication to search the given problem itself. Because agents don't have to search complete problem space, it is expected for agents to decrease their searching loads. Moreover, pheromone style communication effectively works to emerge the cooperative relation among agents for solving the problem.

3.2 Vehicle Routing Problem

The VRP is an abstraction of a class of problems that has been studied both as a theoretical problem and for its practical relevance to applications [7][6]. There are a number of extensions to the VRP which includes time window constraints, vehicle capacity constraints, and so on, however the basic problem consists of a set of customers to be served by a fleet of vehicles from a single depot (see Fig. 3.1).

In this chapter, I focus on a simple VRP as a base for constructing the algorithm. Let $V = \{1, 2, \dots, m\}$ be a set of vehicles, and $C =$

The objective is to find the minimum distance routes which visit all customer nodes.

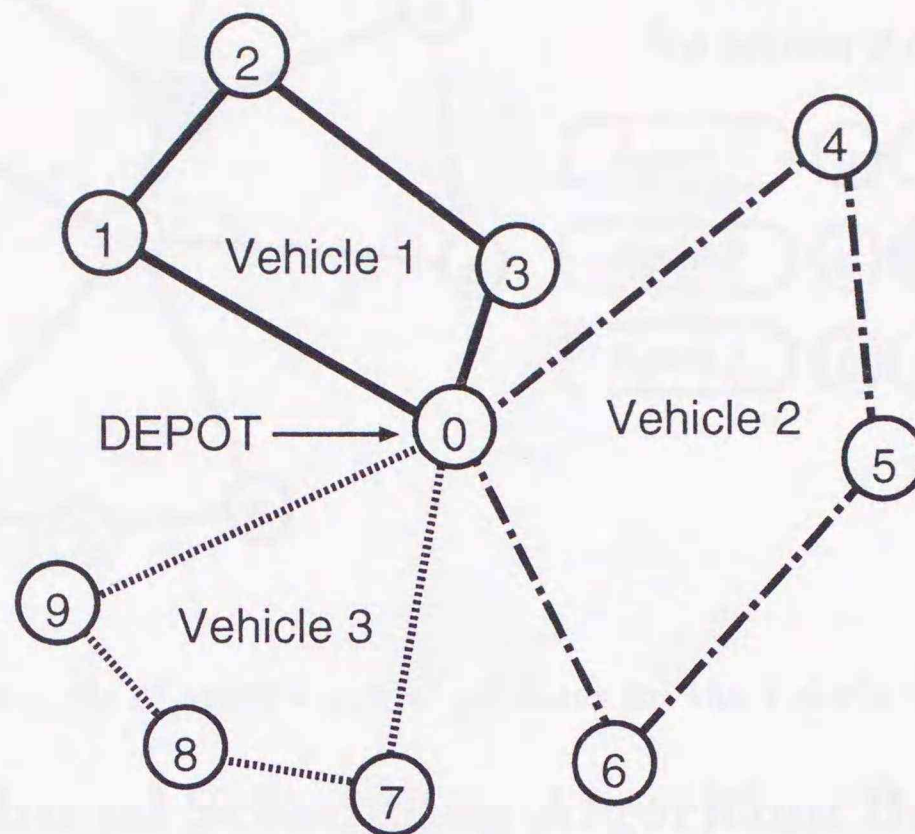


Figure 3.1: The example of a solution for VRP including three vehicle routes.

$\{1, 2, \dots, n\}$ be a set of customers. A set of customer nodes C plus the node 0 (which represents depot node) is denoted as N . Each element of N is placed on the 2 dimensional space, and the position of node i is denoted as (x_i, y_i) . The Euclidean distance is used as distance $d_{(x_i, y_i), (x_j, y_j)}$ for each pair of nodes. Each vehicle has a route which firstly starts from node 0, visits the number of n/m nodes in turn, and finally returns to node 0. As constraints, the customer nodes must be visited just once by any vehicles. The objective of this problem is to find a set of routes which minimize the total distance of delivering goods to the customers. The VRP is denoted as follows.

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{(x_i, y_i), (x_j, y_j)} x_{ijk} \quad (3.1)$$

$$\text{s.t. } \sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \forall i \in C \quad (3.2)$$

$$\sum_{i \in C} \sum_{j \in N} x_{ijk} = n/m, \forall k \in V \quad (3.3)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \forall h \in N, \forall k \in V \quad (3.4)$$

$$x_{ijk} \in \{0, 1\}, \forall i \in N, \forall j \in N, \forall k \in V \quad (3.5)$$

where, k th vehicle has a sub route from i to j if x_{ijk} takes the value of 1, and doesn't have a sub route if it takes 0.

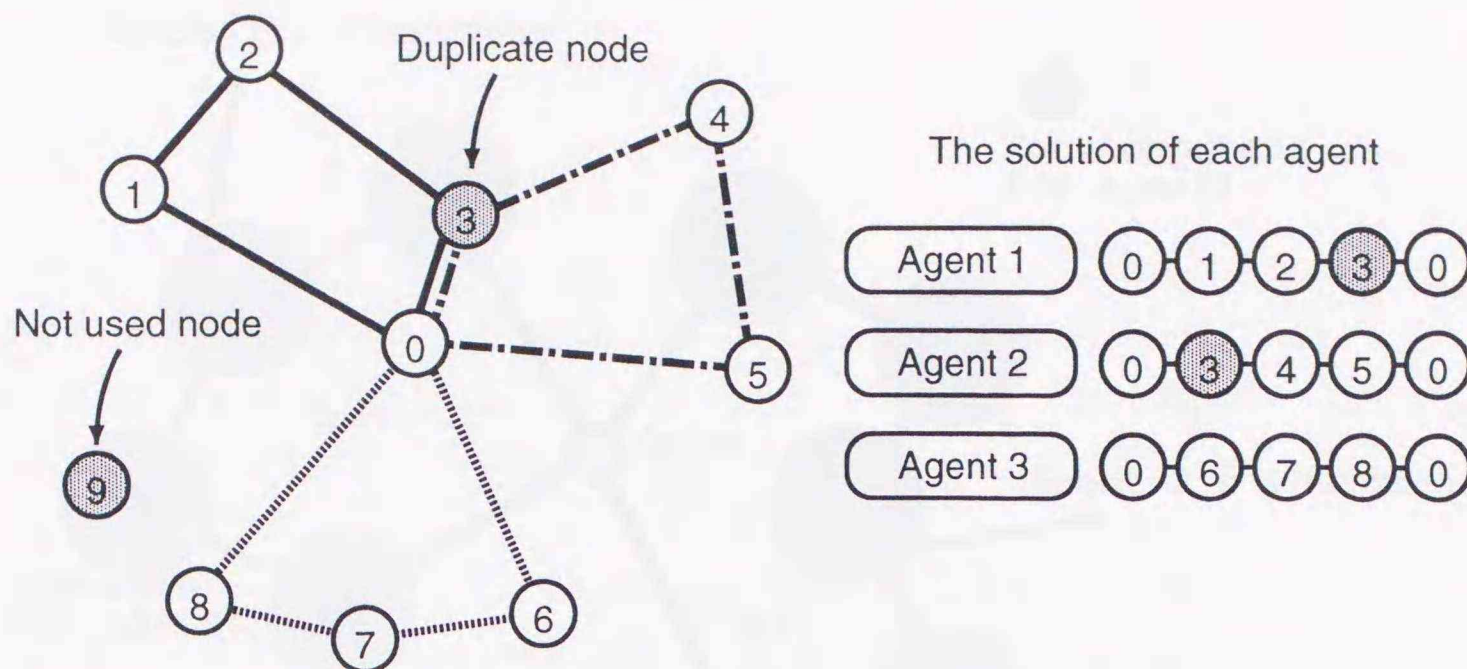


Figure 3.2: The example of agent's partial solutions for the Vehicle Routing Problem.

3.3 Distributed Scheduling Algorithm Based on Multi-agents with Pheromone Style Communication

The proposed distributed scheduling algorithm is based on the multi-agent system. It consists of some vehicle agents which have the ability to autonomously and distributively generate single vehicle schedule for the given problem, i.e., each vehicle route is independently searched by an agent. It is expected that agents decrease their search loads because each agent doesn't have to search complete given problem space. But it raises as a difficult problem how to harmonize whole behaviors of agents. So, I adopt the pheromone style communication, which is derived from the analogy of social insects, as the only communication method for agents to realize the state of other agents and a group of agents. Because the environment manages the appearance and disappearance of pheromone information, the agents have no necessity for management of messages, moreover, it is expected that agents can organize themselves and cooperate each other for solving given problem as social insects.

According to the above idea, the formalization of the multi-agent system based on pheromone style communication is considered here. It is supposed that there are m vehicle agent, and the formalization of vehicle agents are given as follows.

$$A = \langle u, v, x, y, z, f, g, h \rangle, \quad (3.6)$$

where,

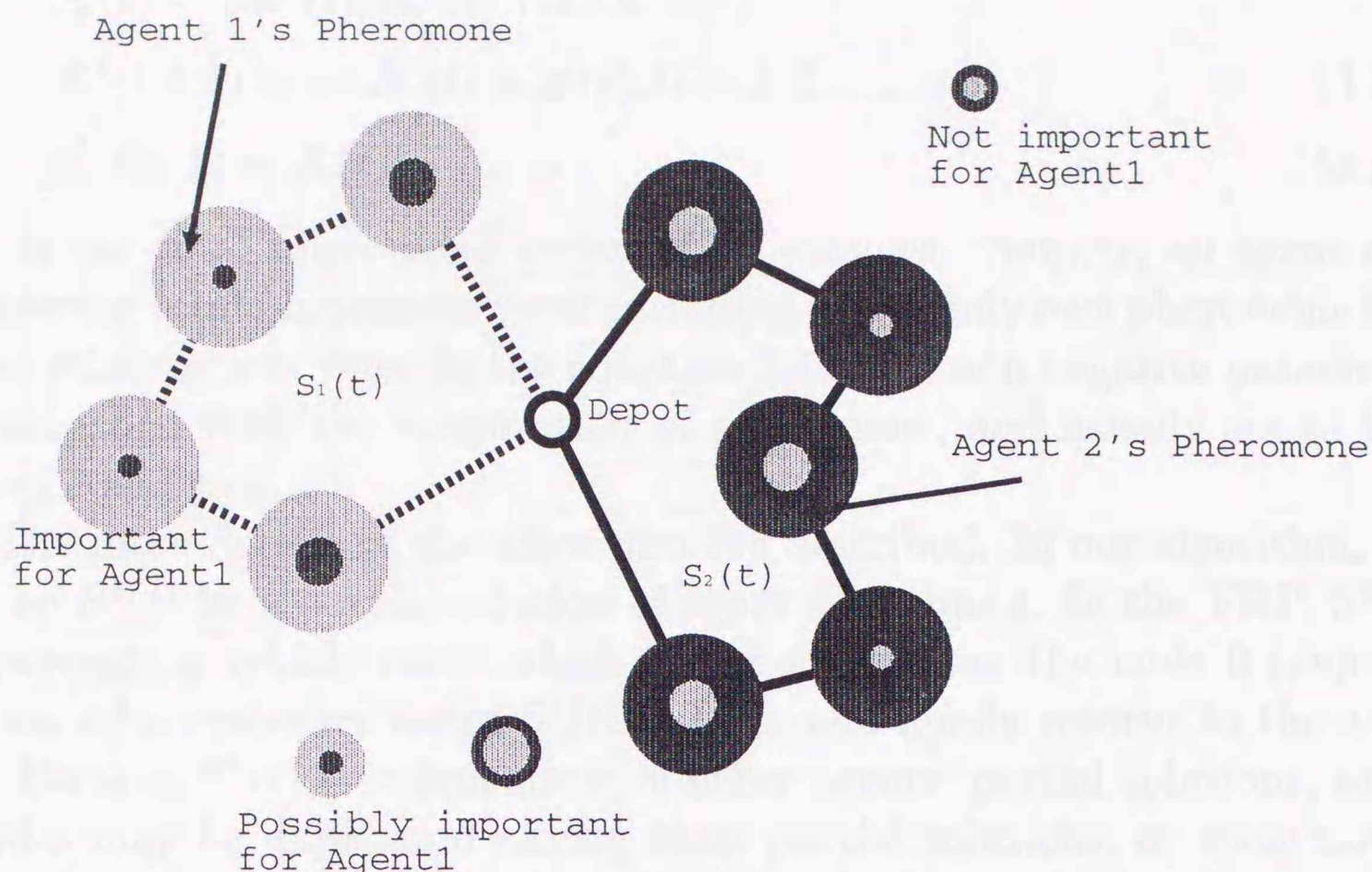


Figure 3.3: The image of meanings of pheromones in the algorithm.

- $u(t)$: Input from the environment
(i.e., the problem domain of given VRP) at time t
- $v(t)$: Input vector from the pheromone space at time t
- $x(t)$: Internal State at time t
- $y(t)$: Output to the environment
(i.e., the partial solution for given VRP) at time t
- $z(t)$: Output vector to the pheromone space at time t

$$x(t+1) = f(x(t), u(t), v(t)) \quad (3.7)$$

$$y(t+1) = g(x(t), u(t), v(t)) \quad (3.8)$$

$$z(t+1) = h(x(t), u(t), v(t)) \quad (3.9)$$

In the above formalizations, the function f is to update its internal state according to inputs from the environment, pheromone space and an internal state of mine. The function g is to decide its partial solution, and h is to decide the output vector of pheromone. It is supposed that $v^i(t)$ and $z^i(t)$ are the input and output vector of j -th vehicle agent respectively.

$$P = \langle v, X, z \rangle \quad (3.10)$$

where,

$$\mathbf{X}(t) = \{\mathbf{X}^1(t), \mathbf{X}^2(t), \dots, \mathbf{X}^m(t)\} \quad (3.11)$$

$$\mathbf{X}^i(t+1) = \rho \cdot \mathbf{X}^i(t) + \mathbf{x}^i(t), (i = 1, 2, \dots, m) \quad (3.12)$$

$$\mathbf{v}^i(t+1) = \mathbf{X}(t) \quad (3.13)$$

$\mathbf{X}$ is the total pheromone vector of all colonies. Namely, all agent can sense the total pheromone vector including in not only own pheromone but also other agents' one. In the equation 3.12, ρ is non negative parameter concerned with the evaporation of pheromone, and usually set to less than equal to 1.

The more details of the algorithm are described. In our algorithm, let m be $S^k(t)$ be a partial solution of agent k at time t . In the VRP, $S^k(t)$ represents a vehicle route which firstly starts from the node 0 (depot), visits n/m customer nodes $S^k(t)$ in turn, and finally returns to the node 0. Because $S^k(t)$ is independent of other agents' partial solutions, some nodes may be duplicated among some partial solutions, or some nodes may be not used by any agents (Fig.3.2). In this case, the solution synthesized from all partial solutions is not feasible for given problem but it is permitted in our algorithm for agents to behave without strongly considering the relations between other agents. It is resolved through pheromone style communication (Fig.3.4).

Initially $S^k(0)$ is generated with n/m nodes randomly chosen from customer nodes. At time t , each agent generates the candidate new partial solution based on its last partial solution and pheromone information as the input from environment. The candidate is decided which to be accepted or not to be accepted according to the acceptance probability calculated from the evaluation of it. After updating the partial solution, the agents adjust their policies concerning evaluation function and output the modifications of pheromone information to environment. When all agents have completed their operations at time t , the pheromone information on environment is updated according to each modifications. The operations of agents and updating of the pheromone information will be repeated until the stop time t_{max} comes. The following subsections describe the details of the operations of agents and the management of pheromone information on environment.

3.3.1 The Operations of Agents

At each time t , agent k alters its partial solution $S^k(t)$, then adjusts its parameters as policy, and decides the modification of pheromone information according to following steps.

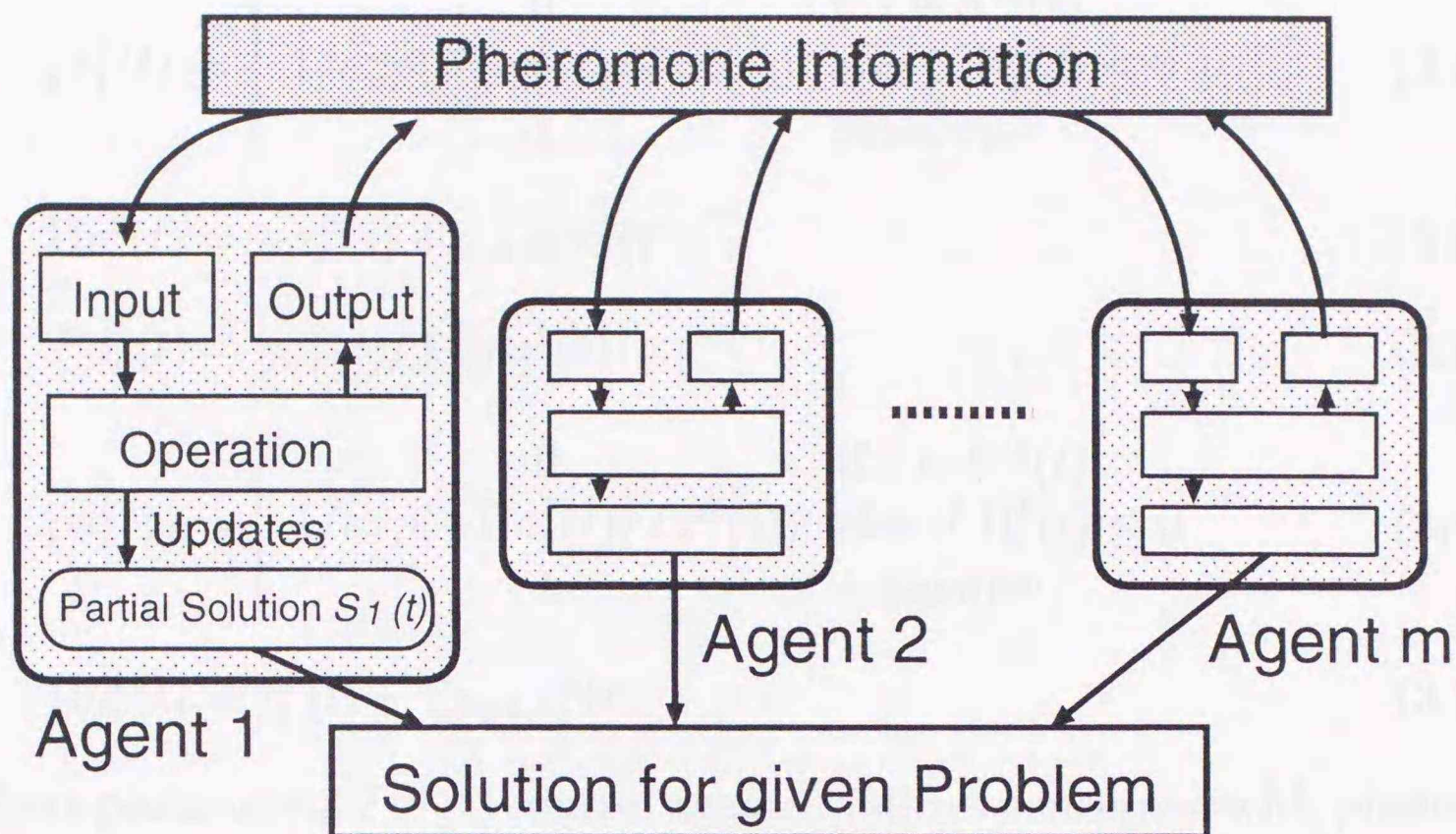


Figure 3.4: The multi-agent system and the architecture of agents for distributed scheduling.

Step 1: $S^k(t)$ is copied to $S^{*k}(t)$ as a pattern of candidate partial solution.

Step 2: One node i belonging to $S^{*k}(t)$ is selected and removed from $S^{*k}(t)$ according to the removing probability $Pr_i^k(t)$.

Step 3: One node i not belonging to $S^{*k}(t)$ is also selected and added to $S^{*k}(t)$ according to the adding probability $Pa_i^k(t)$.

Step 4: $S^{*k}(t)$ is re-composed by the further insertion algorithm.

Step 5: $S^{*k}(t)$ is copied to $S^k(t+1)$ according to the acceptance probability $R^k(t)$. If $S^{*k}(t)$ is not accepted, $S^k(t)$ is simply copied to $S^k(t+1)$.

Step 6: The parameters $TE^k(t)$ are adjusted by the result of step 5.

Step 7: The modifications of pheromone information by agent k is calculated according to $S^k(t+1)$ decided in step 5.

In step 2 and 3, the removing probability $Pr_i^k(t)$ and the adding probability $Pa_i^k(t)$ are defined as follows.

$$Pr_i^k(t) = pr_i^k(t) / \sum_j pr_j^k(t) \quad (3.14)$$

$$pr_i^k(t) = \begin{cases} 0 & \text{if } i \notin S^{*k}(t) \\ 1/\exp(-\Pi r_i^k(t)/T\tau^k(t)) & \text{else if } \Pi_i^k(t) \leq 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.15)$$

$$\Pi r_i^k(t) = -\tau_i^k(t) + \sum_{h \neq k} \tau_i^h(t) \quad (3.16)$$

$$Pa_i^k(t) = pa_i^k(t) / \sum_j pa_j^k(t) \quad (3.17)$$

$$pa_i^k(t) = \begin{cases} 0 & \text{if } i \in S^{*k}(t) \\ 1/\exp(-\Pi a_i^k(t)/T\tau^k(t)) & \text{else if } \Pi_i^k(t) \leq 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.18)$$

$$\Pi a_i^k(t) = \tau_i^k(t) - \sum_{h \neq k} \tau_i^h(t) \quad (3.19)$$

Where parameters $T\tau^k(t)$ control agents' policies concerned with pheromone information. The agent k attaches importance to pheromone information with small value of $T\tau^k(t)$, or the agent k don't attach with large value. Namely the pheromone information with small value of $T\tau^k(t)$ strongly affects the behavior of choosing nodes. In our experiments these parameters are fixed as same small values through running. Pheromone information $\tau_i^k(t)$, which has been formed by the output of the agent k through search process, is related the state between agent k and node i . If the intensity of $\tau_i^k(t)$ is larger value, agent k probably attaches importance to node i and vice versa. The agents can construct the cooperative relation between others by using the pheromone information as probabilities.

In step 4, the further insertion algorithm is known as one of good heuristic algorithm to quickly compose a route of TSP [8]. It is to start with routes on one node and then extended these routes by inserting the remaining node that has the shortest distance to a route node. The example flow of generating new partial schedule is represented in Fig. 3.5.

In step 5, the acceptance probability $R^k(t)$ of new partial solution is defined as

$$R^k(t) = \begin{cases} 1 & \text{if } \Delta E^k(t) \geq 0 \\ 1/\exp(-\Delta E^k(t)/TE^k(t)) & \text{otherwise} \end{cases} \quad (3.20)$$

$$\Delta E^k(t) = -\{E(S^{*k}(t)) - E(S^k(t))\} \quad (3.21)$$

Where $E(S^k(t))$ is the evaluation function of $S^k(t)$ and it corresponds to the total distance of a route $S_k(t)$. $\Delta E_k(t)$ is the change value of evaluations between the candidate partial solution and the last one. $TE^k(t)$ represents the agent's policy concerned with the evaluation of a partial

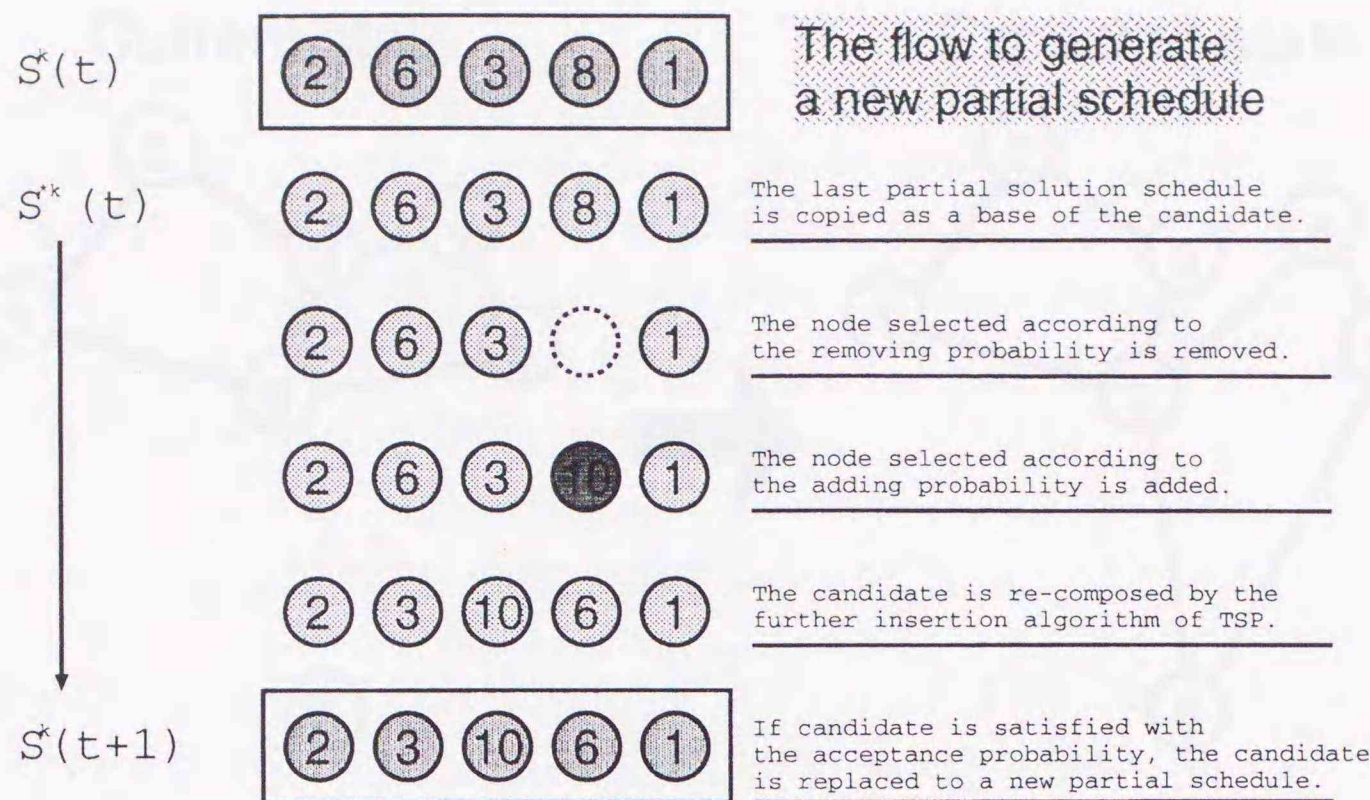


Figure 3.5: The example flow of operations generating a new partial schedule.

solution. If $TE^k(t)$ is small value, agent k tends to strongly prefer partial solutions having good evaluations with breaking the relationship between other agents. It indicates that agents can't adapt to changeable relations between other agents' searches and the behavior of agents converge quickly. Especially, there are some cases that the changes to decrease the evaluation are not expected like shown in Fig. 3.6. So, the management of $TE^k(t)$ is very important for agents behaving appropriately.

In step 6, I adopt the following management mechanism of $TE^k(t)$ to keep the probability $R^k(t)$ proper value.

$$TE^k(t+1) = \begin{cases} TE^k(t) \times TE_{down} & \text{if agent } k \text{ updates its partial solution} \\ TE^k(t) \times TE_{up} & \text{otherwise} \end{cases} \quad (3.22)$$

Where TE_{down} and TE_{up} are the parameters to control $TE^k(t)$. TE_{down} is ordinary set to the value less than 1 and TE_{up} is more than 1. If agent k has scarcely generated the acceptable partial solutions, $TE^k(t)$ is made larger by the influence of TE_{up} . Opposite if agent has vainly generated the partial solutions with poor evaluations, $TE^k(t)$ is smaller by TE_{down} . In the case of $TE_{up} = 1/TE_{down}$, agents would keep the expectancy of probability $R^k(t)$ the value of 0.5. The expectancy tends to take a value more than 0.5 with $TE_{up} > 1/TE_{down}$ and less than 0.5

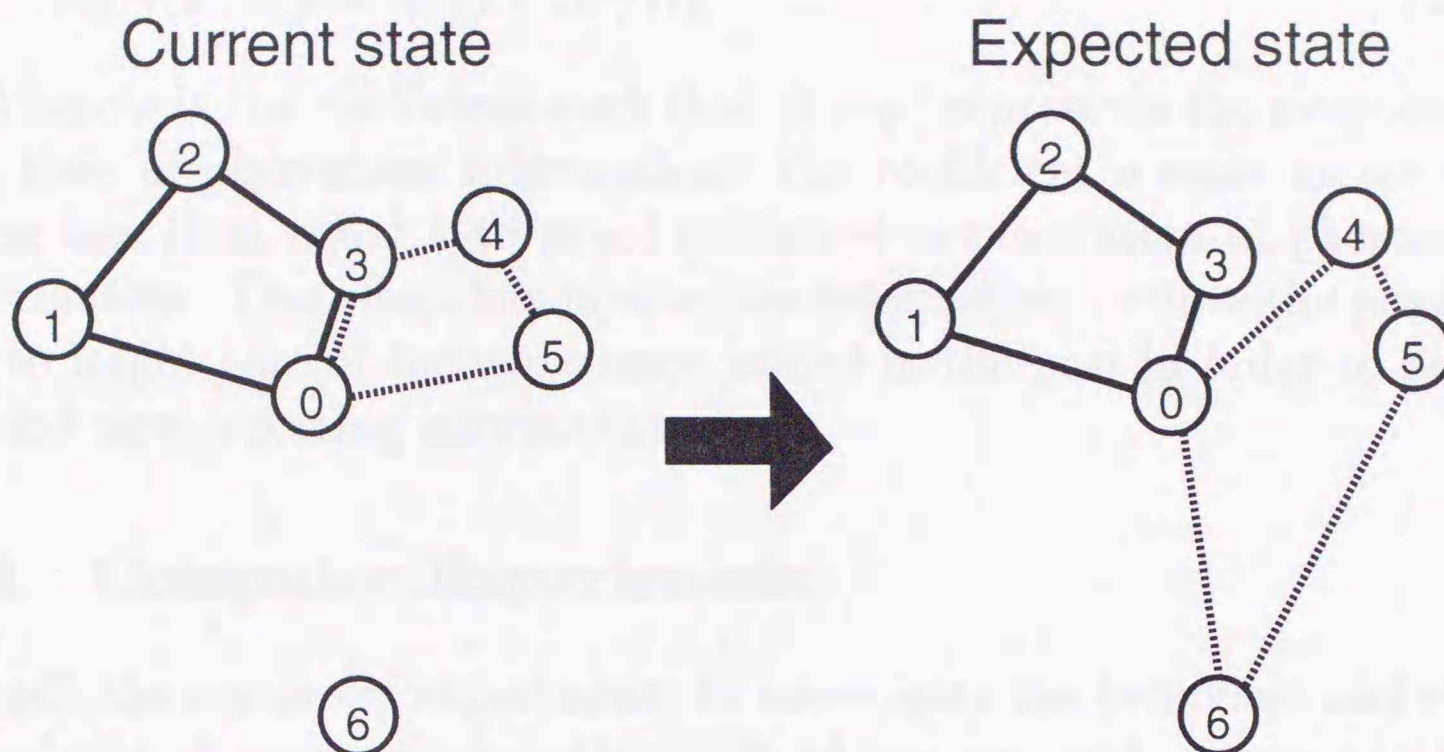


Figure 3.6: The example of agent's partial solutions for the Vehicle Routing Problem.

with $TE_{up} < 1/TE_{down}$. This mechanism enables agents to adjust their evaluation functions to dynamic relations on dealing the nodes.

In step7, the modification of pheromone information by agent k is calculated according to $S^k(t+1)$ as outputs to environment. Agent k simply puts a certain intensity of pheromone on the nodes considered as important according to following formula.

$$\Delta\tau_i^k(t) = \begin{cases} Q_1 & \text{if node } i \in S_k(t+1) \\ Q_2/(1 + d_{(x_i, y_i), (cx^k, cy^k)}) & \text{otherwise} \end{cases} \quad (3.23)$$

Where Q_1 and Q_2 are constant values, and (cx^k, cy^k) represents the average position of nodes contained in $S^k(t)$. The first value of above equation must be larger value than the second because the agent puts a higher amount of pheromone on the nodes used in its partial solution to indicate its importance for other agents. In the second, agent k put a larger intensity of pheromone to nodes considered as important but not used in its partial solution, that is, nodes placed near to its route.

3.3.2 The Management of Pheromone Information on the Environment

The intensity of pheromone information are non-negative real value and set to 0 at initial time of algorithm. Pheromone information is updated after any agents has completed their operations according to the following formula

$$\tau_i^k(t+1) = \rho \times \tau_i^k(t) + \Delta\tau_i^k(t) \quad (3.24)$$

Where ρ is the coefficient such that $(1 - \rho)$ represents the evaporation per time of pheromone information. The coefficient ρ must be set to a value less than equal 1 to avoid unlimited accumulation of pheromone information. This effect lets pheromone information to have the possibility to forget part of the experience gained in the past in order to better exploit new incoming information.

3.4 Computer Experiments

I made the computer experiments to investigate the behaviors and characteristics of proposed algorithm with pheromone style communication. The test problem consists of 60 customer nodes randomly placed in the plane and the node 0 (representing the depot) placed in the center position of plane. The parameter settings used in this experiment are shown in Tab. 3.1. I decided these parameter settings based on the preliminary experiments. The changes of agent routes are shown in Fig. 3.7. In (a) of the figure, the initial routes of 6 vehicle agents generated randomly are shown. At time $t = 1000$, the agents almost specialized own search regions, and searched near these regions in time progress. In this algorithm, each agent doesn't have the ability to know the conditions of other agents except pheromone style communication. The fact that the feasible solutions were obtained by autonomous partitioning of nodes is considered as the evidence that pheromone style communication effectively worked.

Next, the change of pheromone through this experiment are shown in Fig. 3.8. In this figure, the horizontal line indicates the agent number, and the vertical line indicates the node number. That is, the number of 6×60 pheromone intensities are shown. In addition, the radius of each position corresponds to the intensity. The intensities were initially set to zero as (a), and the process of changes can be observed according to time progress. The intensities of some customer nodes are changed, and it is occurred to adjust partitioning the nodes. In the graph of Fig 3.9, the evaluation value of each agent is represented. The evaluation values change dynamically before about $t = 1000$ because search regions of agents have not been specialized yet. After $t = 1000$, the values are almost stable. Especially, each agent couldn't observe other agent's evaluations, but the agents could generate about same quality routes except the agent 2 had little evaluation route than others. It indicates that it

m	ρ	$T\tau^k(0)$	$TE^k(0)$	TE_{up}	TE_{down}	Q_1	Q_2	t_{max}
6	0.65	4.0	40.0	1.0001	0.999	10.0	100.0	5000

Table 3.1: The values of parameters.

is possible for the algorithm to search the problems consisting of scheduled elements having different evaluation values. In Fig. 3.10, the values of TE^k at each t is showed. The values were autonomously controlled by the control mechanism of agents, and it is considered that control mechanism properly worked because agents succeeded partitioning the customer nodes.

Through some experiments, I prepared the appropriate settings of the parameters. As the same time, I found the effects of some parameters to the performance of the algorithm. In the case that ρ was closed to 0, the agents were able to make no feasible solutions for given problem because the pheromone information had been evaporated before its effect worked. When ρ was closed to 1, agents could make feasible solutions but the variety of solutions took place little by reason of long remaining waste information gained in past. ρ around 0.65 showed good performance. $TE^k(0)$ are not sensitive because it has been adjusted by the control mechanism. The values of Q_1 and Q_2 were appropriate in this experiment. There is much probability of discovering the good solution with enough large t_{max} because the agents have generated various solutions without the convergence of behaviors until the stop of algorithm comes.

3.5 Conclusions

In this chapter, I propose the framework of distributed scheduling algorithm for the VRP based on multi agent system with pheromone style communication derived from the analogy of social insects. In proposed algorithm, each agent searches a partial problem generated from original problem with pheromone style communication. Because agents don't have to search complete problem space, they decrease their search loads. Throughout the computer experiment, it is confirmed that multi-agents adaptively cooperate and self-organize to search the given problem.

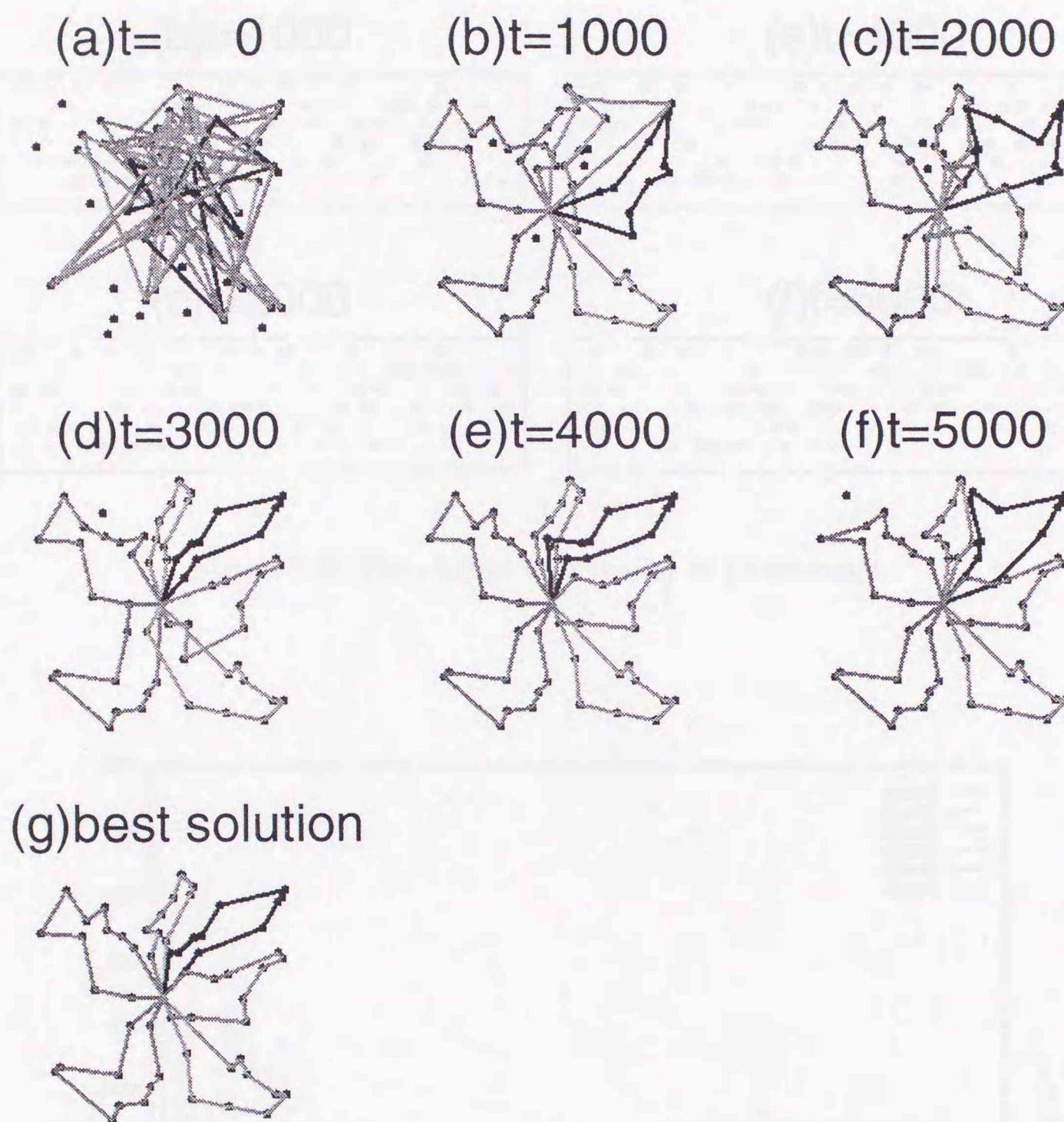


Figure 3.7: The changes of agent routes and the best solution obtained through the experiment.

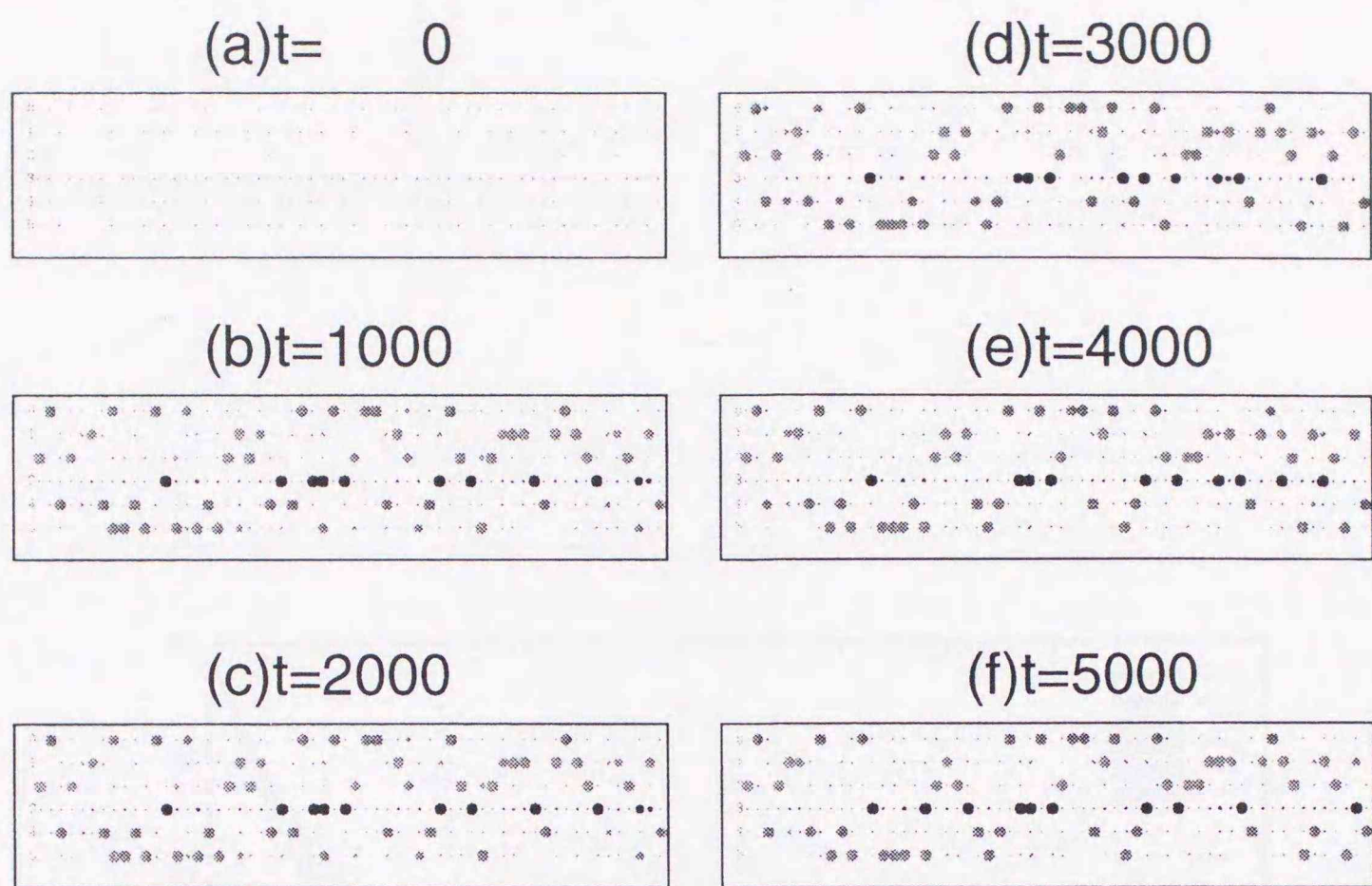


Figure 3.8: The change of intensity of pheromone.

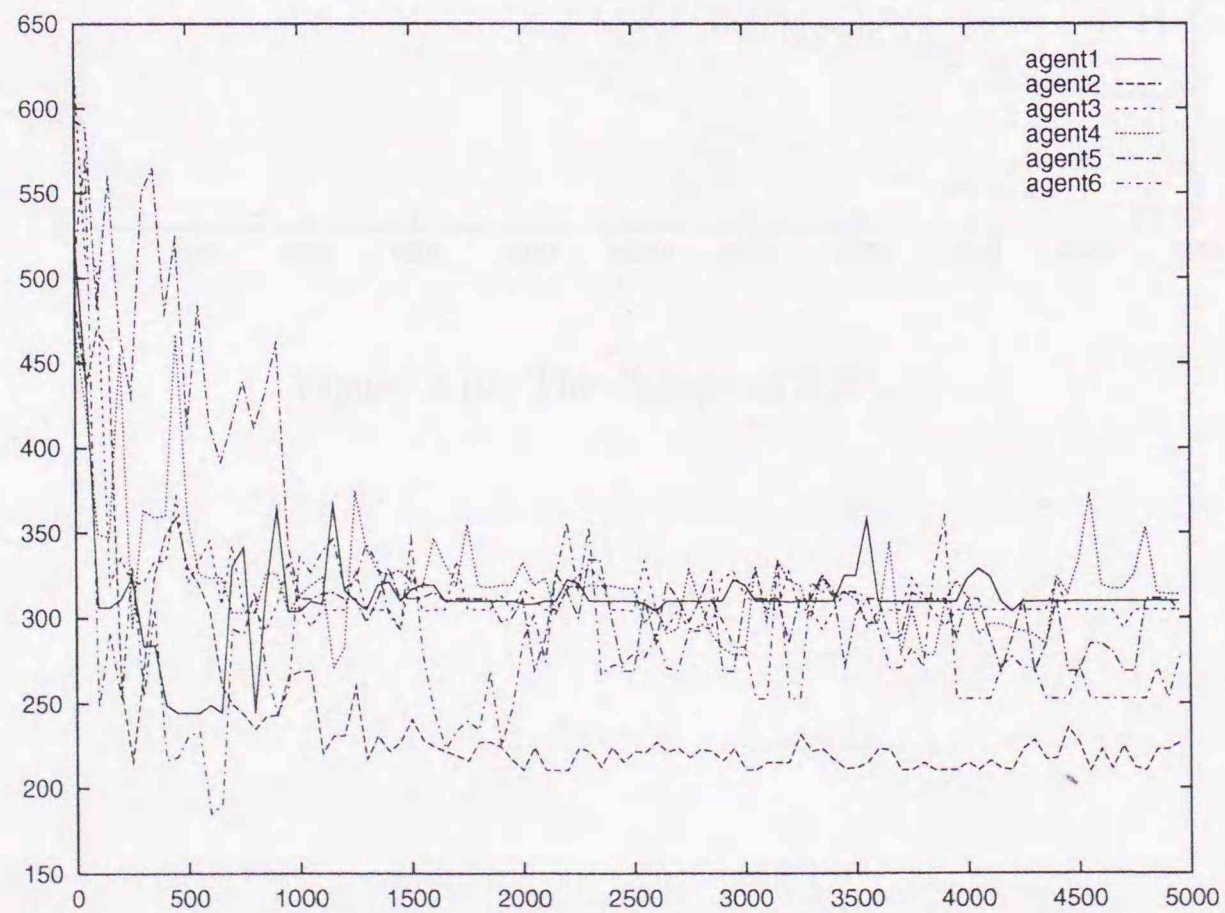


Figure 3.9: The change of evaluation values.

Bibliography

[1] K. Yashima, *Phenomena of Japanese*, in *Kokyo-shugyokyo-shuppan-shi*, The Home of the Biology, Japan, 1976. (in Japanese)

[2] Yoshiko Kurohara, Ikuo Shigenaga, Etsuo Suyama, and Hirofumi Mitsu, A Robust and Behavior like a Self-organizing Model in the Phenomenon of the Human Brain, in *Proceedings of the 1997 Workshop on Symbolic and Simulation of Living Systems*, pp. 370-375, 1997.

[3] Kenji Saito and Hiroshi Ohuchi, Emergent Organization with Communication in a Multi-Agent System, in *Proceedings of the International Conference on Artificial Intelligence Systems*, pp. 173-182, 1998.

[4] David A. Brumby, *Human-Computer Interaction and Learning*, in *Proceedings of the 1997 Workshop on Symbolic and Simulation of Living Systems*, pp. 370-375, 1997.

[5] Masao Ohtsuka, *Vehicle Control in a Multi-Agent System*, in *Proceedings of the 1997 Workshop on Symbolic and Simulation of Living Systems*, pp. 370-375, 1997.

[6] H. Kodaira, H. Gotoh, A. Aoki, and M. Hara, Routing and Scheduling of Vehicles and Crews, *Computers and Operations Research*, Vol. 10, pp. 137-151, 1983.

[7] L. R. Goffman, *Mathematical Simulation: Modeling and Data Search Algorithms for the Vehicle Routing Problem*, *Annals of Operations Research*, Vol. 1, pp. 211-216, 1983.

[8] G. H. G. H. The Traveling Salesman - Computational Solutions for the TSP, *Applications of the Traveling Salesman Problem in Computer Science*, Springer-Verlag, Berlin Heidelberg, 1994.

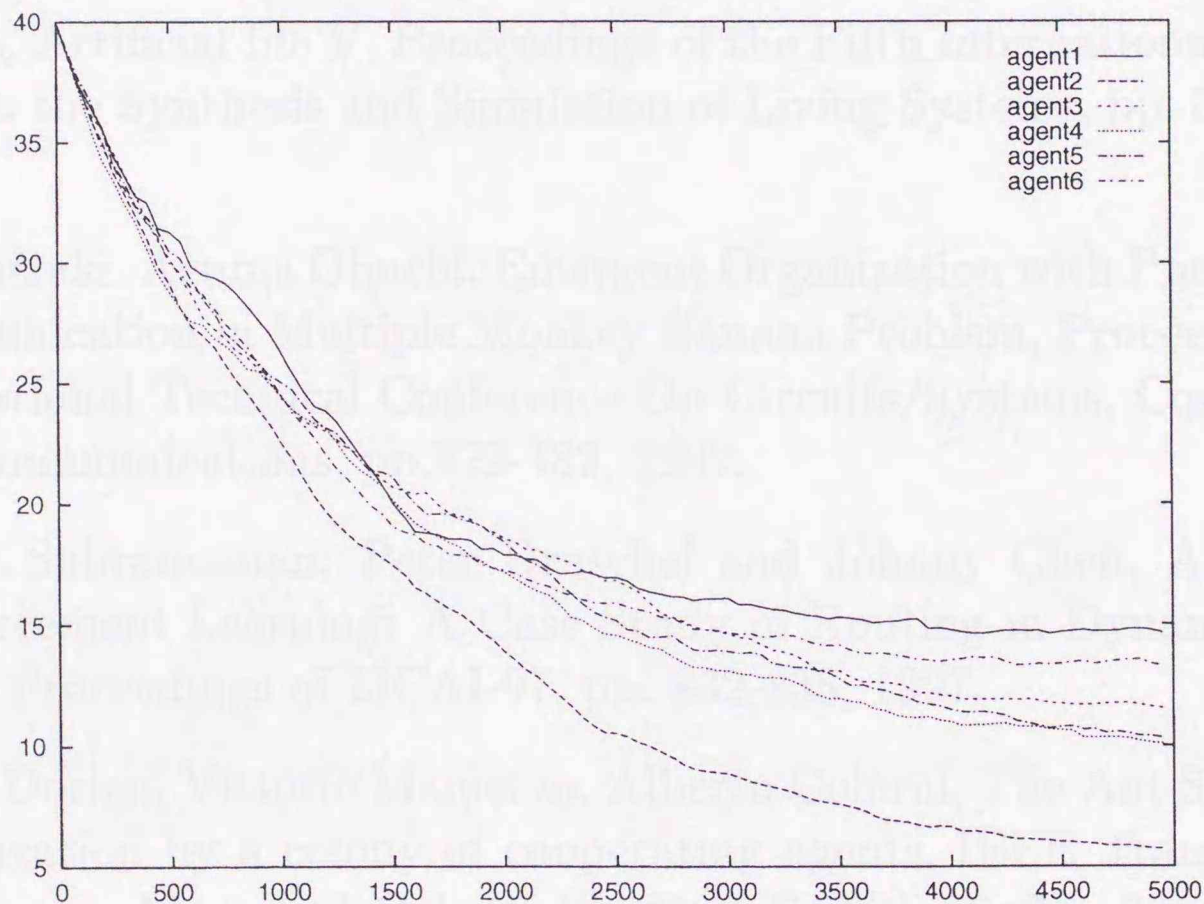


Figure 3.10: The change of TE^k .

Bibliography

- [1] K. Yushima, Pheromones of Insects. In Tokyo-daigaku-syuppankai, The Series of Up Biology, Japan, 1976. (in Japanese)
- [2] Yoshihiro Kuwana, Isao Shimoyama, Yushi Sayama and Hirofumi Miura, A Robot that Behaves like a Silkworm Moth in the Pheromone Stream, Artificial life V, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems, pp. 370-376, 1997.
- [3] Keiji Suzuki, Azuma Ohuchi, Emergent Organization with Pheromone Communication in Multiple Monkey Banana Problem, Proceedings of International Technical Conference On Circuits/Systems, Computers And Communications, pp.479-482, 1997.
- [4] Devika Subramanian, Peter Druschel and Johnny Chen, Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks, Proceedings of IJCAI-97, pp. 832-838, 1997.
- [5] Marco Dorigo, Vittorio Maniezzo, Alberto Coloni, The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13, 1996.
- [6] L. Bodin, B. Golden, A. Assad, and M. Ball, Routing and Scheduling of Vehicles and Crews, Computers and Operations Research, Vol.10, No.2, pp. 117-148, 1983.
- [7] I. H. Osman, Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. Annals of Operations Research 41, pp. 421-451, 1993.
- [8] G. Reinelt, The Traveling Salesman - Computational Solutions for TSP Applications. Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1994.

- [9] W. C. Agosta, Chemical Communication - The Language of Pheromone -, W. H. Freeman and Company, New York, 1992.
- [10] B. Bullnheimer, R. F. Hartl and C. Strauss, A New Rank Based Version of the Ant System: A Computational Study, Working paper, Institute of Management Science, University of Vienna, Australia, 1997.
- [11] G. D. Caro, and M. Dorigo, AntNet: A Mobile Agents Approach to Adaptive Routing. Technical Report 97/12 of IRIDIA, Université Libre de Bruxelles, 1997.
- [12] A. Coloni, M. Dorigo, and V. Maniezzo, Distributed Optimization by Ant Colonies. In Elsevier Publishing, F. Varela and P. Bourgine (eds.), Proceedings of ECAL91-European Conference on Artificial Life, France, Paris, pp. 134-142, 1991.
- [13] A. Coloni, M. Dorigo, and V. Maniezzo, An Investigation of Some Properties of an Ant Algorithm. In Elsevier Publishing, R. Manner and B. Manderick (eds.), Proceedings the Parallel Problem Solving from Nature Conference, Brussels, Belgium, pp. 509-520, 1992.
- [14] A. Coloni, M. Dorigo, V. Maniezzo and M. Trubian, Ant System for Job-shop Scheduling, Belgian Journal of Operations Research, Statistics and Computer Science, 34, pp. 39-53, 1994.
- [15] D. Costa and D. Snyers, Ants can Colour Graphs, Journal of the Operational Research Society, 48, pp. 295-305, 1997.
- [16] M. Dorigo, V. Maniezzo, and A. Coloni, The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13, 1996.
- [17] M. Dorigo, and L. M. Gambardella, Ant Colonies for the Travelling Salesman Problem. In Elsevier, BioSystems 43, pp. 73-81, 1997.
- [18] L. M. Gambardella and M. Dorigo, Ant-Q: A Reinforcement Learning approach to the traveling salesman problem, Proc. ML-95, Twelfth International Conf. on Machine Learning, pp. 252-260, 1995.
- [19] H. Kawamura, M. Yamamoto, T. Mitamura, K. Suzuki, and A. Ohuchi, Cooperative Search Based on Pheromone Communication for Vehicle Routing Problems, IEICE Transactions, Fundamentals, Vol. E81-A, No. 6, pp. 1089-1096, 1998.

- [20] P. Kuntz, P. Layzell and D. Snyers, A Colony of Ant-like Agents for Partitioning in VLSI Technology, Proc. of Forth European Conf. on Artificial Life, Phil Husbands and Inman Howey (Eds.), pp. 417-424.
- [21] V. Maniezzo, A. Coloni and M. Dorigo, The Ant System Applied to the Quadratic Assignment Problem, Technical Report 94/28 of IRIDIA, Université Libre de Bruxelles, 1994.
- [22] T. Mikami, S. Mikami, M. and Wada, Ant-like Collective Sorting for Massive Data Analysis - A Quantitative Study -. In IOS Press, Ohmsha, Y. Kakazu, M. Wada and T. Sato (eds.). Proceedings of Intelligent Autonomous Systems (IAS-5), pp. 686-693, Japan, 1998.
- [23] H. Mühlenbein, M. Schomisch and J. Born, The Parallel Genetic Algorithm as Function Optimizer, Proc. the Fourth International Conf. on Genetic Algorithm, Richard K. Belew and Lashon B. Booker (Eds.), pp. 271-278, 1991.
- [24] M. Nakamura, and K. Kurumatani, Formation Mechanism of Pheromone Pattern and Control of Foraging Behavior in an Ant Colony Model. In MIT Press, Christopher G. Langton and Katsunori Shimohara (eds.), Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems, pp. 67-74, 1997.
- [25] G. Reinelt, The Traveling Salesman - Computational Solutions for TSP Applications, Lecture Notes in Computer Science 840, G. Goos and J. Hartmanis (Eds.), Springer-Verlag, 1994.
- [26] T. Stützle and H. Hoos, The Max-Min Ant System and Local Search for the Traveling Salesman Problem, Proc. IEEE International Conf. of Evolutionary Computation (ICEC'97), pp. 308-313, 1997.
- [27] D. Subramanian, P. Druschel, and J. Chen, Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks, Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), pp. 832-838, 1997.
- [28] K. Suzuki, and A. Ohuchi, Reorganizations of Agents with Pheromone Style Communication in Multiple Monkey Banana Problem, In IOS Press, Ohmsha, Y. Kakazu, M. Wada and T. Sato (eds.). Proceedings Intelligent Autonomous Systems (IAS-5), pp. 615-622. 1998.

- [29] M. Yamamoto, H. Kawamura, K. Suzuki, and A. Ohuchi, Collective Approach for Combinatorial Optimization Problems, Proceedings of The International Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC98), pp. 1479-1482, 1998.

Chapter 4

Ants War with Pheromone Style Communication

4.1 Introduction

In nature, there are many kinds of interesting organisms which make highly structured group behaviors to adapt for their environment, and they have various communication ways to interact with each other for their lives. Typical examples of such organizations are termites, honeybees, fish, and so on. In particular, ants and bees, which are called social insects, are known to communicate with each other through pheromones for keeping their organizations [1].

Pheromones are chemical substances put on the ground or emitted in the air by each insect, and the information applications are remarkable in the organizations of organisms. There are various signals (such as sexual signals) according to their types. Ants and bees can communicate with each other by using the characteristics of pheromones themselves. For instance, in real ant colonies, ants finding the food items lay the release way to their nest with the specific type of pheromone after eating them and the ants which follow the pheromone trail and they still find the way to the food. When all food items are completed to be gained, the pheromone trail evaporates from the environment, and all ants immediately return to the nest and automatically disappear. Real ants use many types of pheromones and keep their complex organizations for their lives.

Several researchers study interesting style communication in various applications. For example, Colantoni proposed the ant colony algorithm to solve combinatorial optimization problems [2]. In the ant colony algorithm, the ants move randomly on the graph and they find the shortest path from the source to the destination through pheromone trails. The algorithm is based on the behavior of ants in nature, and it is very effective for solving combinatorial optimization problems.

Chapter 4

Ants War with Pheromone Style Communication

4.1 Introduction

In nature, there are many kinds of interesting creatures which exhibit highly structured group behaviors to adapt for their environment, and they take various communication ways to interact with each other for their lives. Typical examples of such communication ways include sounds, lights, gestures, and so on. In particular, real ants and bees, which are called social insects, are known to communicate with each other through pheromones for keeping their organizations [1].

Pheromones are chemical substances put on the ground or emitted to the air by such insects, and the evaporation and diffusion are remarkable as the characteristics of pheromones. These are message signals having special meanings according to these types. Real ants and bees can communicate with each other by using of the characteristics of pheromones very well. For instance in real ant colonies, ants finding the food items lay the return way to their nest with the specific type of pheromone attracting other ants, and the ants sensing it follow the pheromone trail and they efficiently find the way to the foods. When all food items are completed to be gained, the pheromone trail evaporates from the environment, and already unnecessary information are automatically disappeared. Real ants use many types of pheromones and keep their complex organizations for their lives.

Several researchers study pheromone style communications in various applications. For optimizing, Coloni proposed the ant algorithm for traveling salesman problem with the analogy of foraging behaviors of real ants [3]. For observing group behaviors with pheromone style communications, Nakamura studied on foraging behaviors of artificial ant agents

[5]. However, most studies are based on hand-coded pheromone style communications, and they have never touched the autonomous emergence of pheromone communications. On the other hand, Suzuki focused on the evolutionary acquisition of pheromone style communication on the Monkey Banana Problem of multiple agent, but the pheromones in his work have designed special meanings for execution of tasks [6]. So, it is expected for pheromone style communications to emerge autonomously out of no meanings according to solved tasks.

The aims of this chapter are to confirm whether it is possible or not to acquire pheromone style communications out of no meanings through the evolutionary process and to discuss about communications of multi-agent systems. For the evolutionary acquisition, it is difficult to answer the question which is earlier to emerge the meanings of the pheromones or to acquire the reactions to pheromones, although if one is acquired, then the other is easily decided. For such aims, the ants war of artificial ant agents is introduced. The agents have the limited sensor which can detect only neighbor grids, and pheromone style communications are only media for communications. Therefore, it is necessary for agents to cooperate and communicate with each other in order to win this war and survive in the environment. The evolutionary process of communications is observed through the ants war world.

This chapter is composed as follows. In section 2, I introduce the ants war of artificial ant agents and the agent architecture with the neural network. Section 3 includes the ant agents based on the continuous pheromone sensor, and section 4 concerns with the agents based on the discrete pheromone sensors. Each sections includes the computer experiments and discussions. Finally I conclude this chapter in section 5.

4.2 Ants War

In this section, the ants war of artificial ant agents is introduced for the sake of the evolutionary acquisition of pheromone style communications. As shown in Fig. 4.1, the stage of this war is constructed on the $X \times Y$ grids, and there are two kinds of agent teams, "White ants" and "Black ants". The objective of each team is to carry back and gain more food items which are initially put on the center line of the world, but they have only limited sensing ability to perceive neighbor grids. Moreover, the food items are too heavy to carry by only individual power, so the collaboration is very important to win this war. The characteristics of

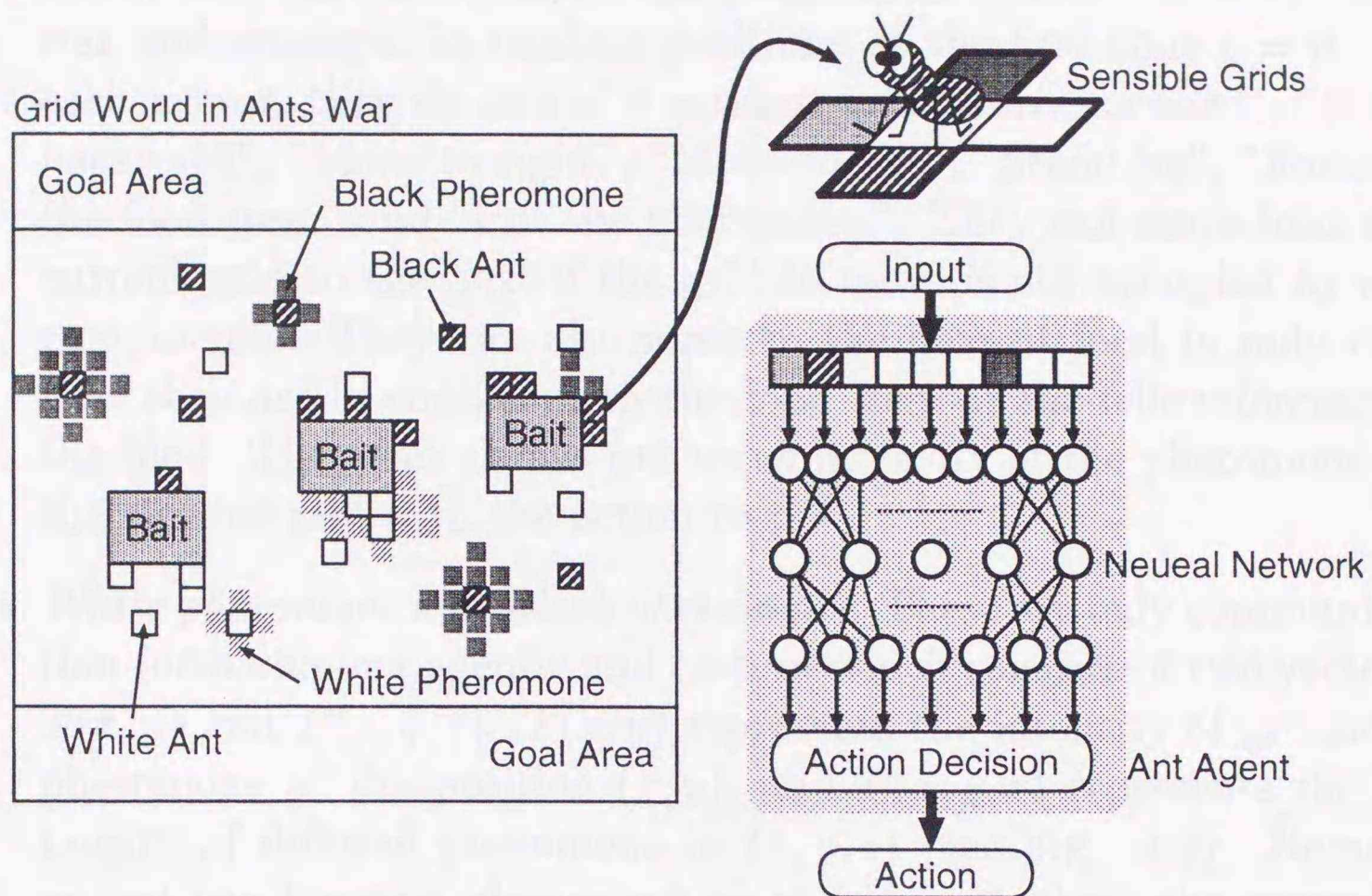


Figure 4.1: Ants war world and architecture of the ant agent.

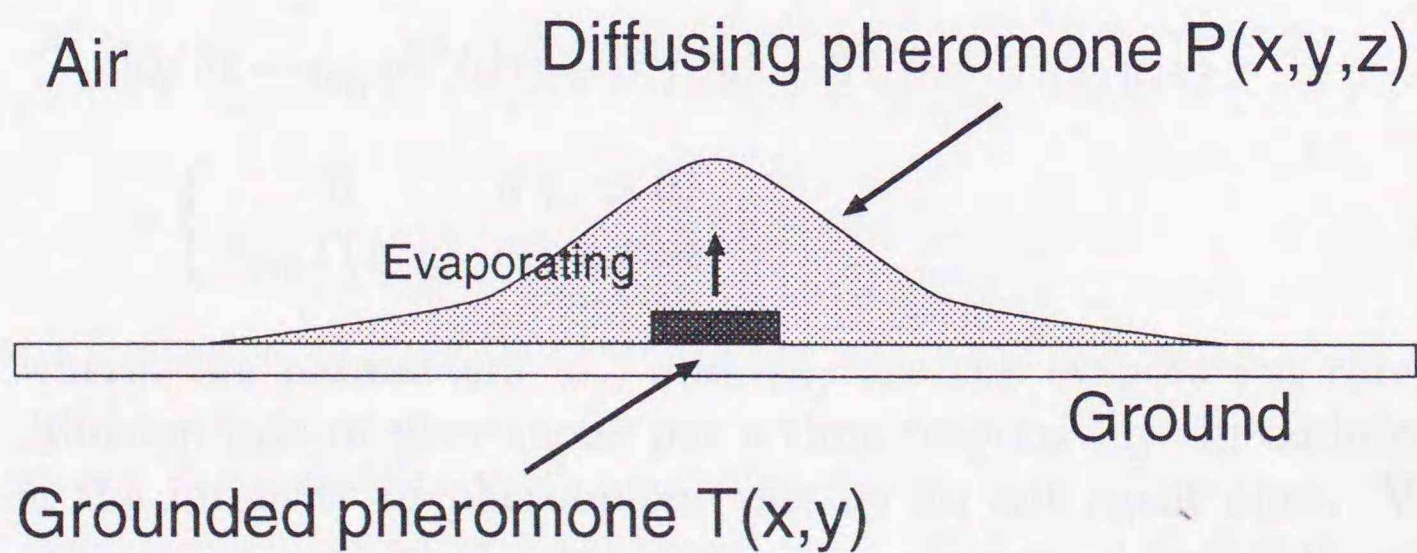


Figure 4.2: The image of evaporation and diffusion of pheromone.

each component are as follows.

- *White ants* and *Black ants*: They are main characters of the ants war and arranged in random positions at the first time $t = 0$. At each time t , they do one of 9 actions, i.e., "Move forward", "Move backward", "Move to right", "Move to left", "Stand by", "Force to the food item" and "Put the pheromone". They can move from the current grid to the next if the grid to move is not occupied by any components. They can also force to the nearest food in only case that they are in contact with the food item or the fellow forcing to the food. They can always put some intensity of the pheromone on the current grid with the action to put.
- *White pheromone* and *Black pheromone*: These are only communication media among agents, and each of which consists of two vectors, $T(x, y)$ and $P(x, y, z)$. $T(x, y)$ represents the intensity of grounded pheromone at the position (x, y) , and $P(x, y, z)$ represents the intensity of diffused pheromone in (x, y, z) (see Fig. 4.2). Namely, an ant can lay own pheromone on the ground, then, the grounded pheromone diffuses to the air according to time progress. The diffusional equations are defined as following partial differential ones.

$$[\partial/\partial t + \gamma_{eva}]T(x, y) = \Delta T^k(x, y) \quad (4.1)$$

$$\Delta T^k(x, y) = \begin{cases} Q & \text{if agent } k \text{ on the grid } (x, y) \\ & \text{puts the pheromone} \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

$$\begin{aligned} & [\partial/\partial t - \gamma_{dif}(\partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2)]P(x, y, z) \\ &= \begin{cases} 0 & \text{if } (z > 0) \\ \gamma_{eva}T(x, y) & \text{otherwise} \end{cases} \end{aligned} \quad (4.3)$$

where, the parameters γ_{eva} and γ_{dif} are the evaporation rate and diffusion rate of pheromone per a time respectively. In addition, Q is the intensity of pheromone put by an ant agent once. White pheromone and black pheromone are evaporated and diffused according to the above equations.

- *Food items*: These are the objects carried by ants, and can be moved up and down on the grids. Each food has the parameter which represents the degree to be forced. The forced parameters are increased

by the white forcing ants and decreased by the black forcing ants. When the absolute value of parameter is achieved to the constant value, the food item is moved by one grid to the direction indicated by the sign of parameter. The food would be moved quickly by lots of ants but slowly by a few. After move, the parameter is cleared to zero. If the food item is over each goal line, it is completed to be gained and disappeared from the grid world. The winner of the war is the team which gets over half number of food items in halfway of time or gets more foods in stop time t_{max} coming. If both teams get no food items, the team carrying the foods for more distances is winner.

As the characteristics of the ants war, I should notice

- how they find the food in the stage of the war with the limited sensing ability.
- how they percept to be required the forcing action in no contact with the food item.
- how they quickly switch their strategies in the situation that they finish to gain the food item and it is disappeared from the environment.
- how they select the carried food item from several ones for the appropriate resource allocation.

Namely, the interactions among agents are very important to resolve the above, and I consider that the ants war is suitable to observe the evolutionary process of the pheromone style communications. Using this game settings, the two types of pheromone sensors are introduced, namely, the continuous pheromone sensor and the discrete pheromone sensor. The ant agent with continuous pheromone sensor is used for preliminary experiments to investigate the possibility of evolutionary emergence of pheromone style communication, and the ant agent with discrete pheromone sensor is for detail discussion concerning with the evolutionary emergence of pheromone style communication. The agent architectures including some sensors, the computer experiments, and the discussion will be appeared in following sections.

4.3 The Formalization of Ant Agents and Pheromone Communication

The formalization of ant agents on the ants war is given as follows.

$$A = \langle \mathbf{u}, \mathbf{v}, \mathbf{y}, \mathbf{z}, g, h \rangle \quad (4.4)$$

where,

$\mathbf{u}(t)$: Input from the environment at time t

$\mathbf{v}(t)$: Input vector from the pheromone space at time t

$\mathbf{y}(t)$: Output to the environment at time t

$\mathbf{z}(t)$: Output vector to the pheromone space at time t

Compared with other agent formalizations, the ant agent on the ants war doesn't have the internal state $\mathbf{x}(t)$ because the ant agent is constructed as a reactive one here. Because the pheromone space is the vector space, an input $\mathbf{v}(t)$ and output $\mathbf{z}(t)$ concerning with the pheromone space are vectors.

$$\mathbf{y}(t+1) = g(\mathbf{u}(t), \mathbf{v}(t)) \quad (4.5)$$

$$\mathbf{z}(t+1) = h(\mathbf{u}(t), \mathbf{v}(t)) \quad (4.6)$$

In the above equations, the function g is to decide its action to the environment, and h is to decide the output vector of pheromone. Namely, the functions g and h are evolutionally designed based on the genetic algorithm.

Let $\mathbf{v}_i(t)$ and $\mathbf{z}_i(t)$ be an input and output of i -th agent at time t , and the pheromone space is formalized as follows.

$$P = \langle \mathbf{v}, \mathbf{X}^G, \mathbf{X}^A, \mathbf{z}, F^G, F^A \rangle \quad (4.7)$$

where,

$\mathbf{X}^G(t)$: Internal state vector of pheromone space
on the ground at time t

$\mathbf{X}^A(t)$: internal state vector of pheromone space
in the air at time t

$$\mathbf{X}^G(t+1) = \mathbf{F}^G(\mathbf{X}^G(t)) + \sum_i \mathbf{z}_i(t) \quad (4.8)$$

$$\mathbf{X}^A(t+1) = \mathbf{F}^A(\mathbf{X}^G(t), \mathbf{X}^A(t)) \quad (4.9)$$

$$\mathbf{v}_i(t+1) = \mathbf{X}^A(t) \quad (4.10)$$

where, the functions of $\mathbf{F}^G$ and $\mathbf{F}^A$ represent the evaporation function of the pheromone and the diffusion function of the pheromone respectively. Namely, the agent can sense the pheromone which is diffused from the ground.

4.4 The Ant Agent with Continuous Pheromone Sensor

In this section, I describe about the architecture of agents with continuous pheromone sensor. The continuous pheromone sensor is introduced for preliminary experiments to investigate the possibility of evolutionary emergence of pheromone style communication. It is easy to design the pheromone sensor by that the inputs of pheromone sensor take direct intensities of sensing pheromone. The details of agent architecture are as follows.

4.4.1 The Agent Architecture

The ant agents in the team are constructed with same regular neural network having 9 input units, 7 hidden units, and 7 output units. The reason that all ants in the team have same network is motivated from the fact that real social insects possess a lot of genes in their colonies [7], of course including the handiness of implementation. Let the value of k -th input unit be noted as i_k , and the values of inputs are decided from the sensor as

$$i_1 = \begin{cases} 1 & \text{if the agent touches the food item} \\ 0 & \text{otherwise} \end{cases}$$

$$i_{2-5} = \begin{cases} 1 & \text{if the agent is in contact with the fellow agent in the} \\ & \text{\{front, back, right, left\} grid} \\ 0 & \text{otherwise} \end{cases}$$

$$i_{6-9} = \text{the intensity of own team pheromone in the} \\ \text{\{front, back, right, left\} grid (continuous values)}$$

$$i_{10} = -1 \text{ (bias)}$$

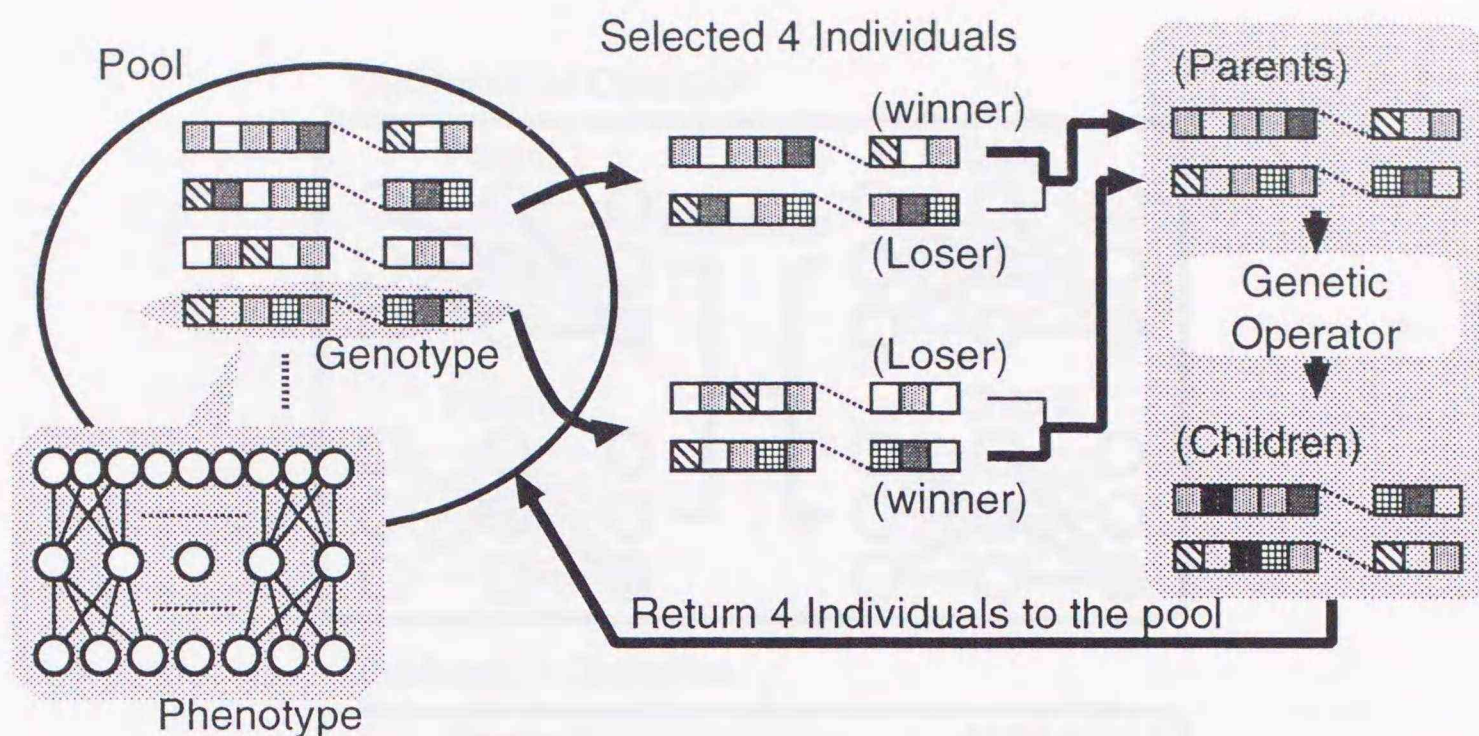


Figure 4.3: A series of GA operations in a GA time.

The outputs of 7 units are naturally calculated through the hidden units with the sigmoid function.

$$h_i = f\left(\sum_{j=1}^{10} w_{ij}i_j\right) \quad (4.11)$$

In addition, h_8 is set to the value of -1 as bias, and

$$o_i = f\left(\sum_{j=1}^7 v_{ij}h_j\right) \quad (4.12)$$

$$f(x) = 1/(1 + \exp(-x)) \quad (4.13)$$

where, h_i is the output of i -th hidden unit, and o_i is the output of i -th output unit. Both of w_{ij} and v_{ij} are weights connecting i -th pre-layer unit to j -th unit respectively. The function f is sigmoid one.

The values of 7 outputs, $o_1, o_2, \dots, o_7$, correspond to the degree to be selected as the agent action. The agent regard such degrees as the probabilities to select the action, and decides it based on roulette wheel selection. These very simple agents are adopted for the ants war. Namely, each action is selected according to following probabilities.

$$\text{Probability to select the action } i = o_i / \sum_j o_j \quad (4.14)$$

4.4.2 Evolutionary Strategy

The GA is applied to the ant agents for the evolution. In the GA, the phenotypes of the agent strategies, that is, the weight sets of the neural

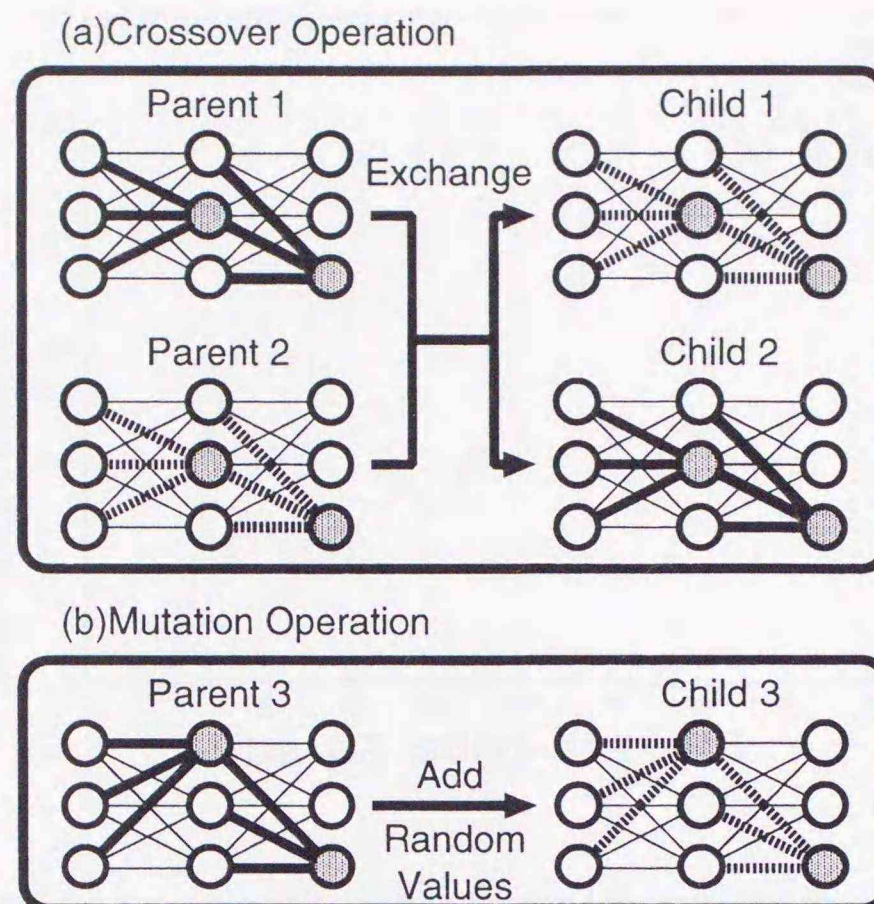


Figure 4.4: The crossover and mutation operations used in the GA.

networks, are required to be encoded to the genotypes. The weight set as the phenotype are directly encoded to the genotype, so the genotype is a vector of weight values [4]. The fitness function is generally designed for the selection, but it is not defined here. If I can design the fitness function for the evolution, I can picture the expected behaviors of agents, and surviving individuals merely have high fitness values as the results to optimize it. Our interest is what strategy is produced through the evolutionary process based on only the results of the war and whether it can be or not be observed the emergence of communications out of nothing. Therefore, the selection operator used here selects some individuals from the pool with even probabilities, and the winner individuals simply survive and produce their children as the results of wars (Fig. 4.3). Namely, it can be said that the fitness landscape would be dynamically variable according to individuals contained in the pool. The agent strategies obtained through the GA process only represent that they have survived, so it is interesting.

The number of N individuals are prepared in the pool, and these individuals are initialized with randomized weight values. At the GA step g , the following operations are applied to the individuals until stop criteria g_{max} comes.

1. Two pairs of individuals are randomly selected from the pool as

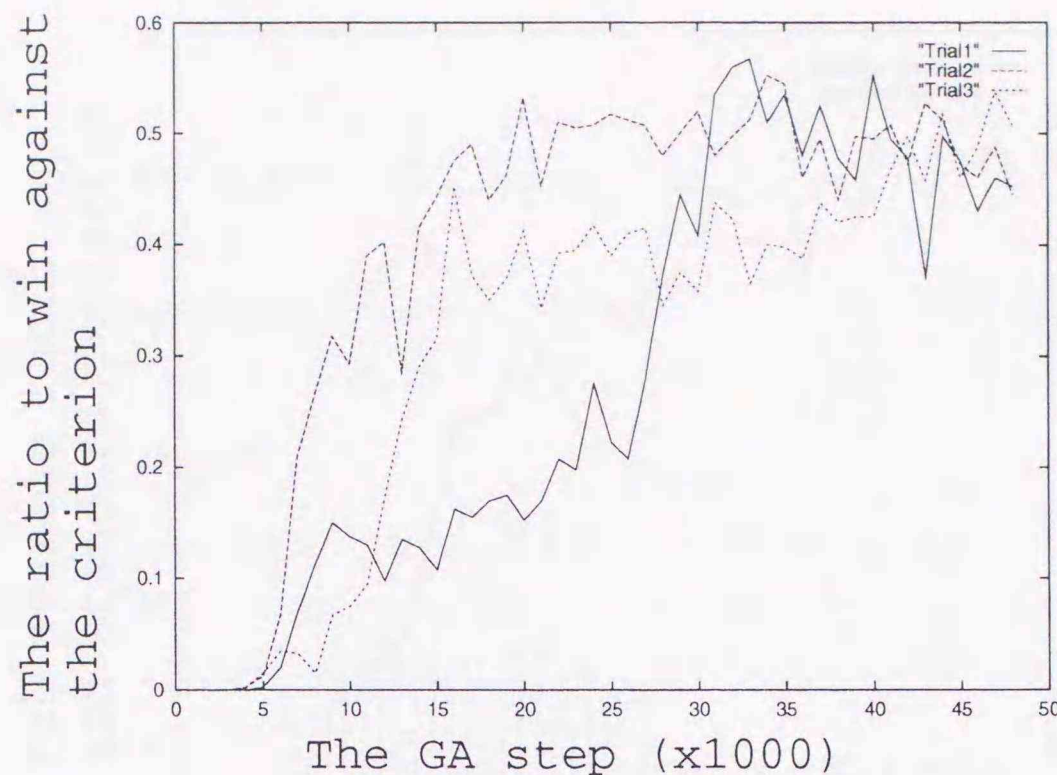


Figure 4.5: The graph representing the ratios to win against the criterion final population.

candidates of parents.

2. Two individuals of each pair are fought in the ants war. Two winners become the parents, and two losers are erased.
3. Two parents are copied to two children as the bases.
4. The crossover operator is applied to two children according to the crossover probability P_c .
5. The mutation operator is applied to two children according to the mutation probability P_m .
6. Two parents and new two children are returned to the pool. The number of individuals in the pool are always kept the number of N .

Applying the crossover operator, several input nodes are selected according to the probability P_{ch} , and the weight values connecting to these nodes are exchanged among two parents (Fig. 4.4(a)). In the mutation operator, several nodes are selected with same probability P_{ch} , and the weights are added $[-0.5, +0.5]$ random values (Fig. 4.4(b)). The operations applied in the GA are very simple and don't include any special techniques for expected evolution, so the nature of the evolutionary process would be cleared.

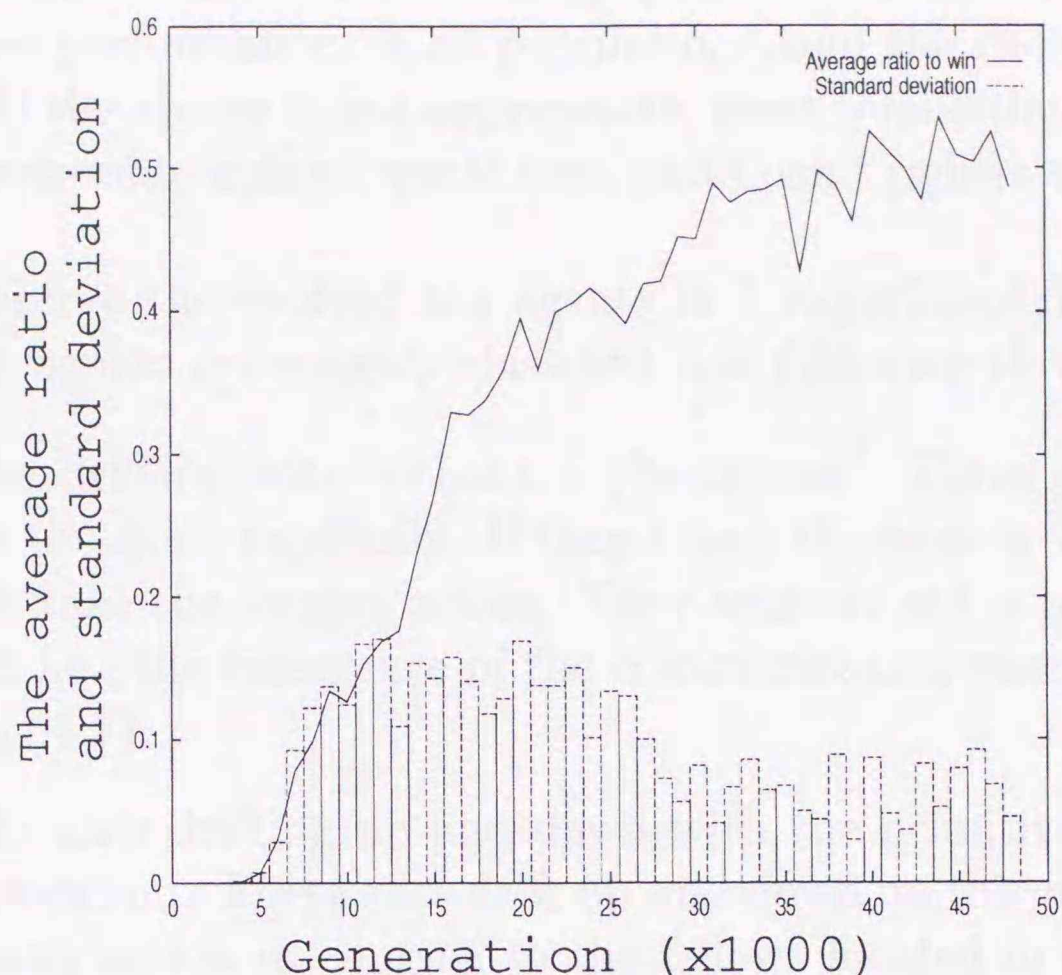


Figure 4.6: The graph representing the average ratios to win against the criterion in 7 trials.

4.4.3 Experimental Results

Several computer experiments were made with the teams consisting of 50 ant agents. These agents had 7 hidden units neural networks, and genetic parameters for 20 individuals in the pool were as $P_c = 0.4$, $P_m = 0.6$, $P_{ch} = 0.1$ and $g_{max} = 50000$.

At first, it is necessary to show whether the evolution properly occurred or didn't, but it is difficult because the selection pressure of the evolution is based on only results of the wars, and the GA doesn't have any global fitness functions to evaluate populations. So, I measure the progress of the evolution by regarding the final population as the criterion to evaluate and fighting every individual in the final population and every one in each genetic time [2], i.e., $400(N \times N)$ matches are made for evaluating one population. If the progress is occurred favorably, the ratios to win against the criterion would be gradually increased from the first population to the final, and be close to 0.5 finally. The graph in Fig. 4.5 represents the transitions of the ratios in typical 3 trials, and the graph in Fig. 4.6 represents the average ratio in 7 experimental trials. The horizontal axis and vertical one indicate the GA time and the ratios respectively. I can confirm the evolution progressed in each trial and the selection pressure

certainly worked. Moreover, the final populations as the criteria almost show the best performances in all populations, and the choice of criteria is suitable. If the choice is not appropriate, most populations would win against the criterion or most would lose, and I can't rightly evaluate each population.

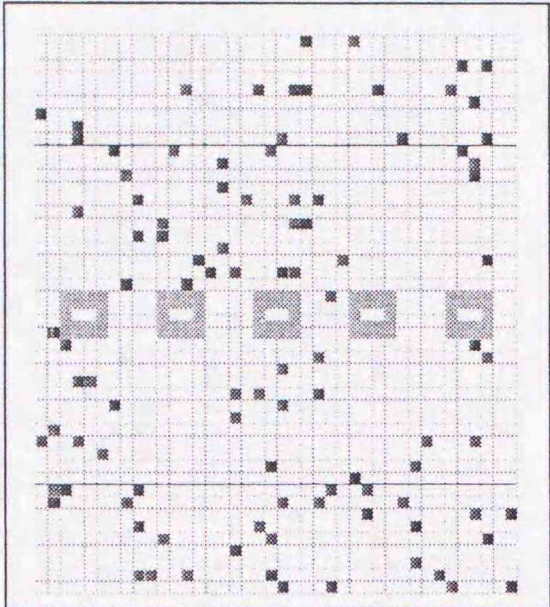
Next, I observe the evolved ant agents in 7 experimental trials. The obtained ant agents are roughly classified into following three types.

Type 1: The drifting ants without a pheromone. These ants tend to drift on the grids randomly. If they touch the food or fellow agents in front, take the forcing action. They scarcely put a pheromone in the war, i.e., the emergence of the communication was not occurred (see Fig. 4.7).

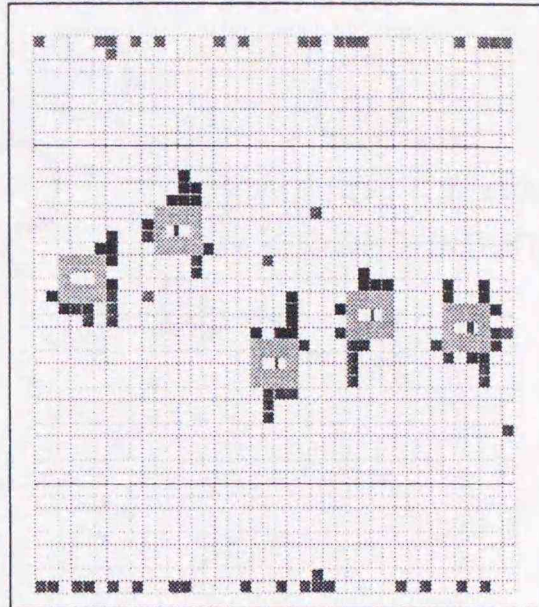
Type 2: The ants drifting up and down with the primitive pheromone communication. These ants drift up and down on the grids and take the forcing action in clinging to the fellows located in the left side, so these ants often take the formation like a bar. In particular, when the leftmost ant in the bar formation doesn't touch the food item, this ant puts the pheromone. Then, other ants sensing the pheromone move from current positions, and the deadlock situation that any ants don't touch the food item is resolved. I can say that this is the primitive pheromone style communication (see Fig. 4.8).

Type 3: The marching ants with the pheromone. These ants move backward with little intensities of the pheromone on neighbor grids and take forcing action in contact with the food item or the fellow in front. In addition, these ants sometimes put the pheromone indiscriminately. The ants which are initially placed at upside half of grids reach near food items efficiently. While, the remainder collect in the downside of the grids, so the intensities of the pheromone in the downside are greater by indiscriminate putting action. The ants sensing strong pheromone begin to march forward with putting the pheromone, and they can reach the food item. Especially in the situation that the food item is obtained and disappeared, these ants quickly begin to march forward and reach the other food item again later. The screen shots of the war with this type of ants are shown in Fig. 4.9. In the war, most ants are observed to take part in forcing the food items, and these ants highly cooperate with each other to win the war by using the pheromone style communication.

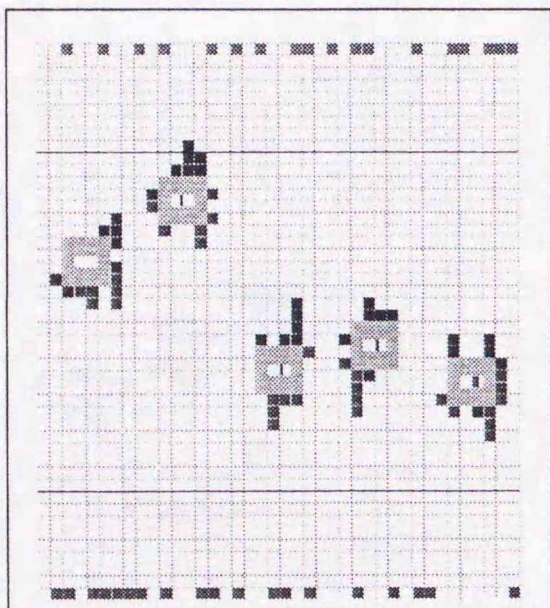
(a) Initial state



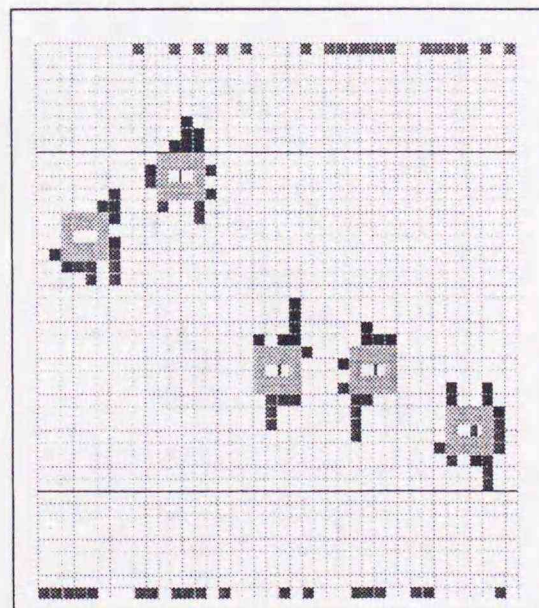
(b) $t=100$



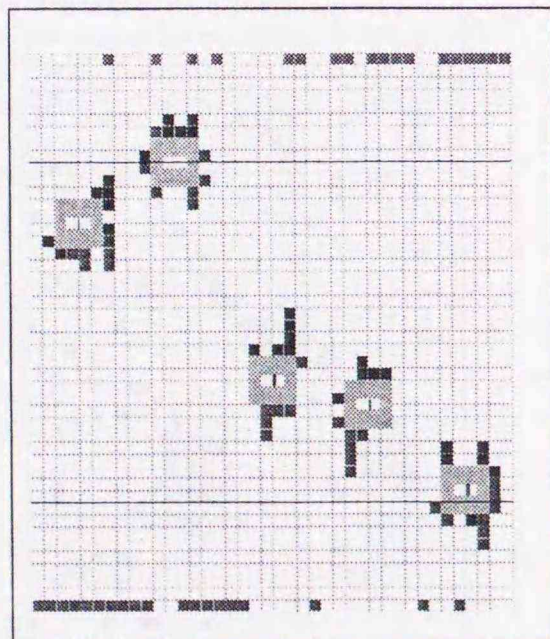
(c) $t=200$



(d) $t=300$



(e) $t=400$



(f) $t=500$

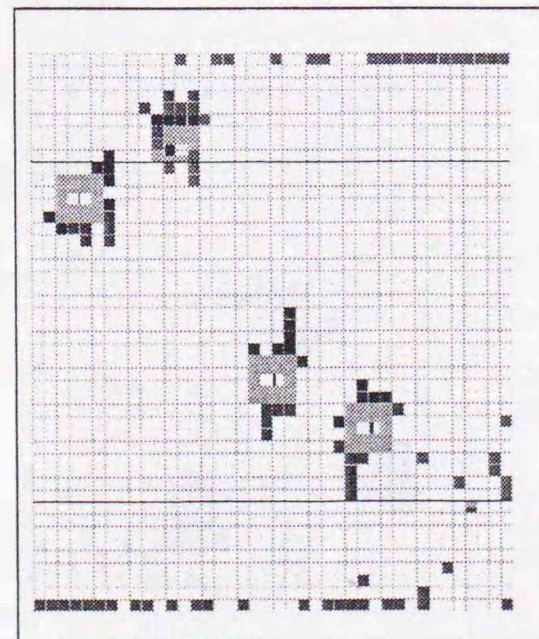
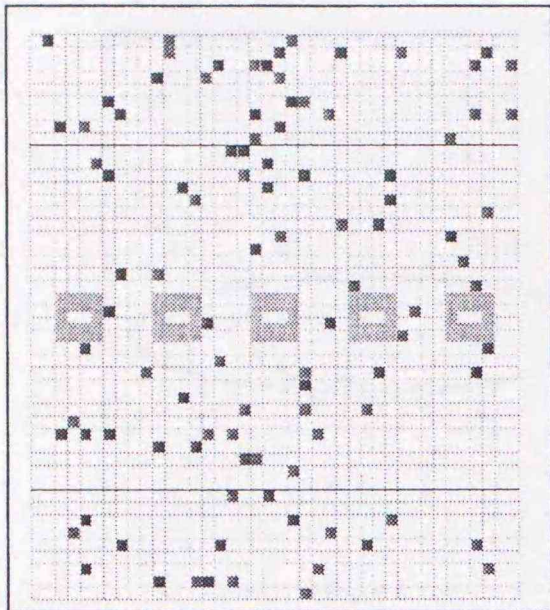
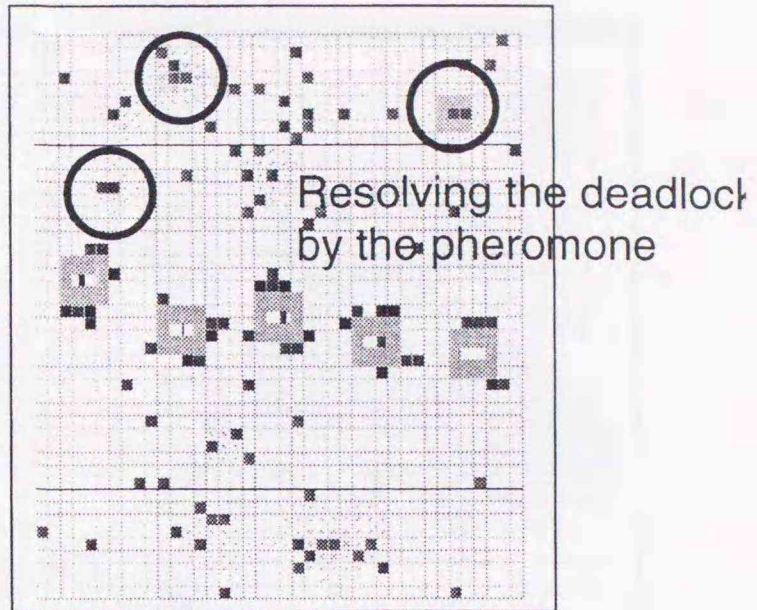


Figure 4.7: The screen shots in the ants war of type 1 agents.

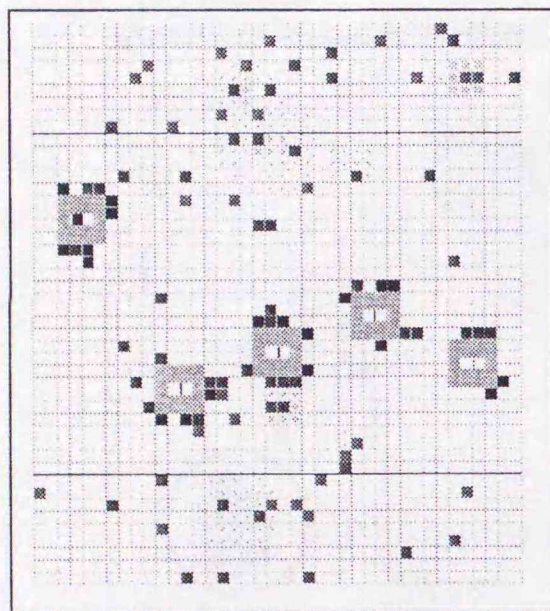
(a) Initial State



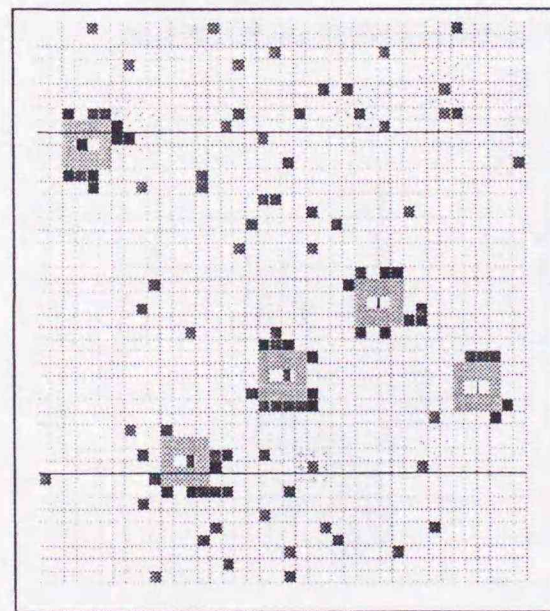
(b) $t=100$



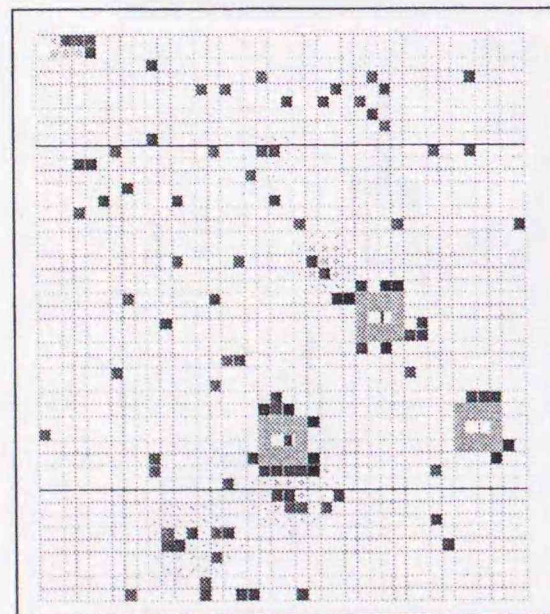
(c) $t=200$



(d) $t=300$



(e) $t=400$



(f) $t=500$

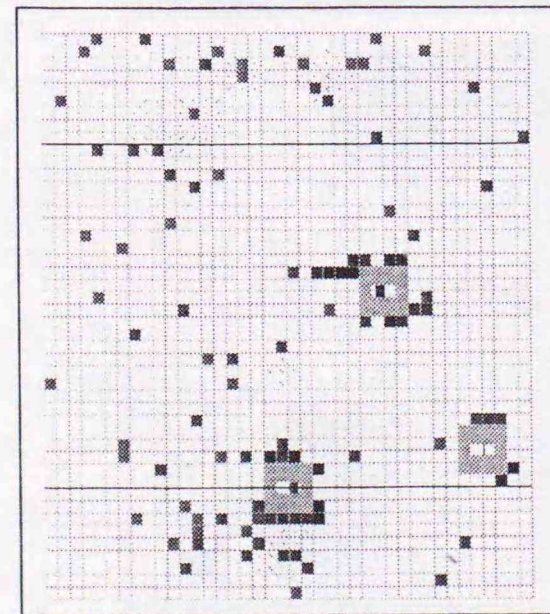
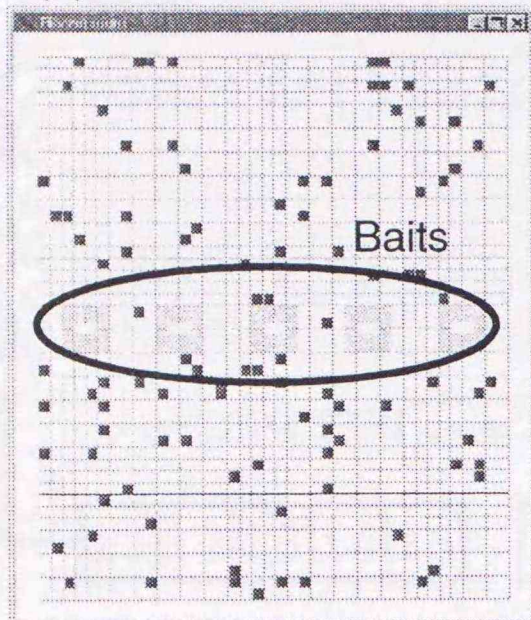
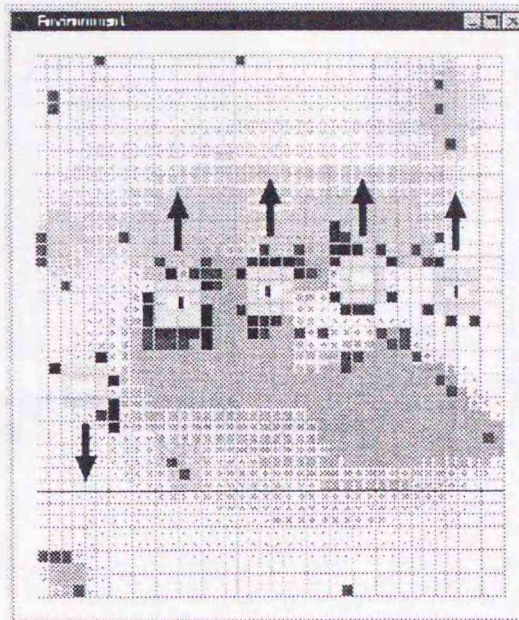


Figure 4.8: The screen shots in the ants war of type 2 agents.

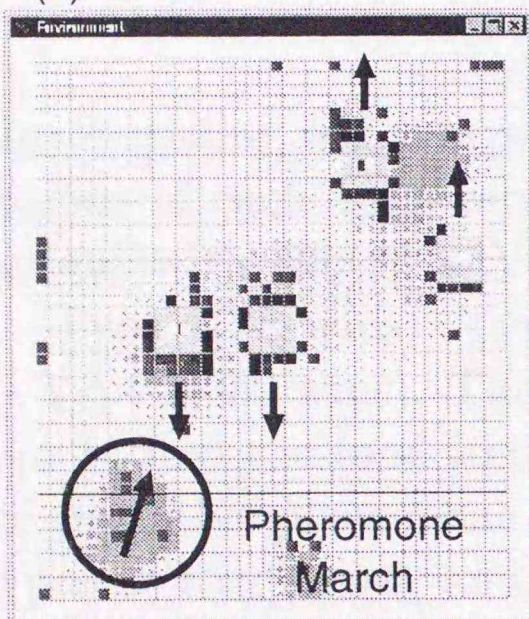
(a) Initial state



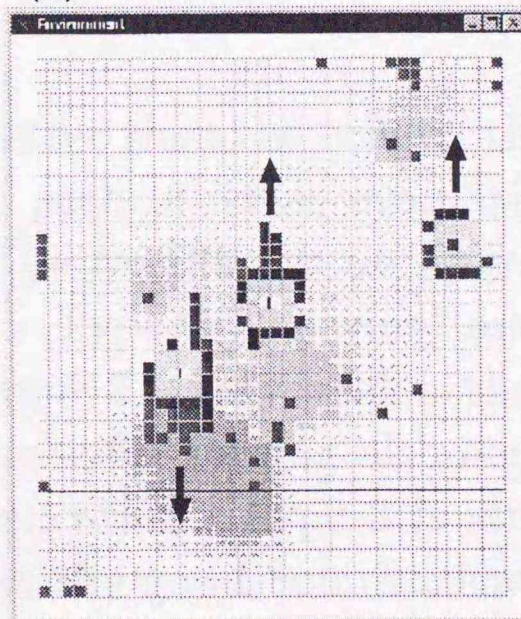
(b) $t=100$



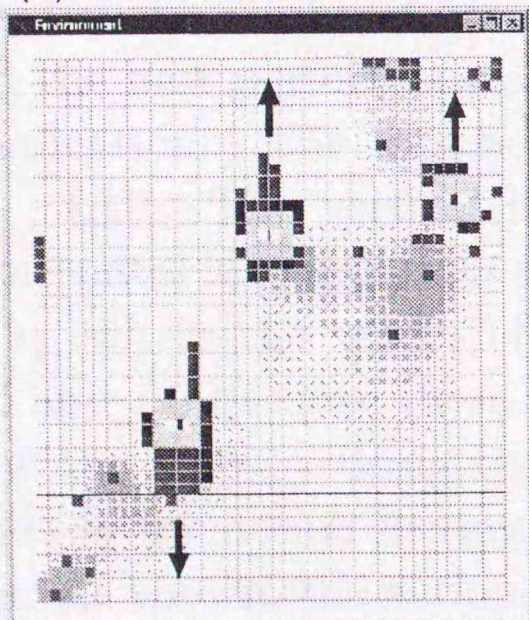
(c) $t=200$



(d) $t=300$



(e) $t=400$



(f) Final state

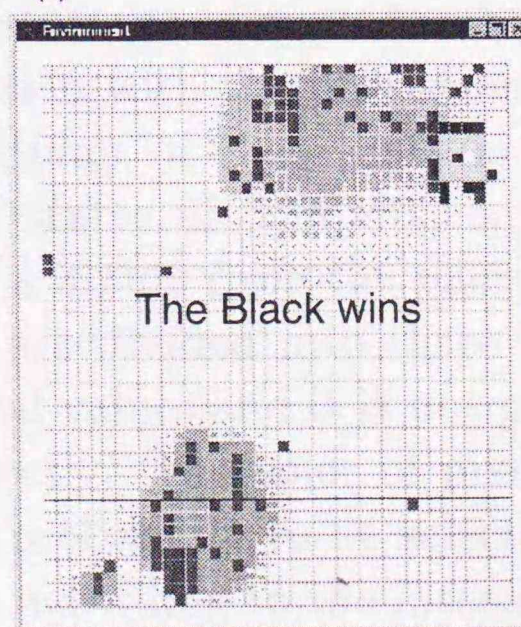


Figure 4.9: The screen shots in the ants war of type3 agents.

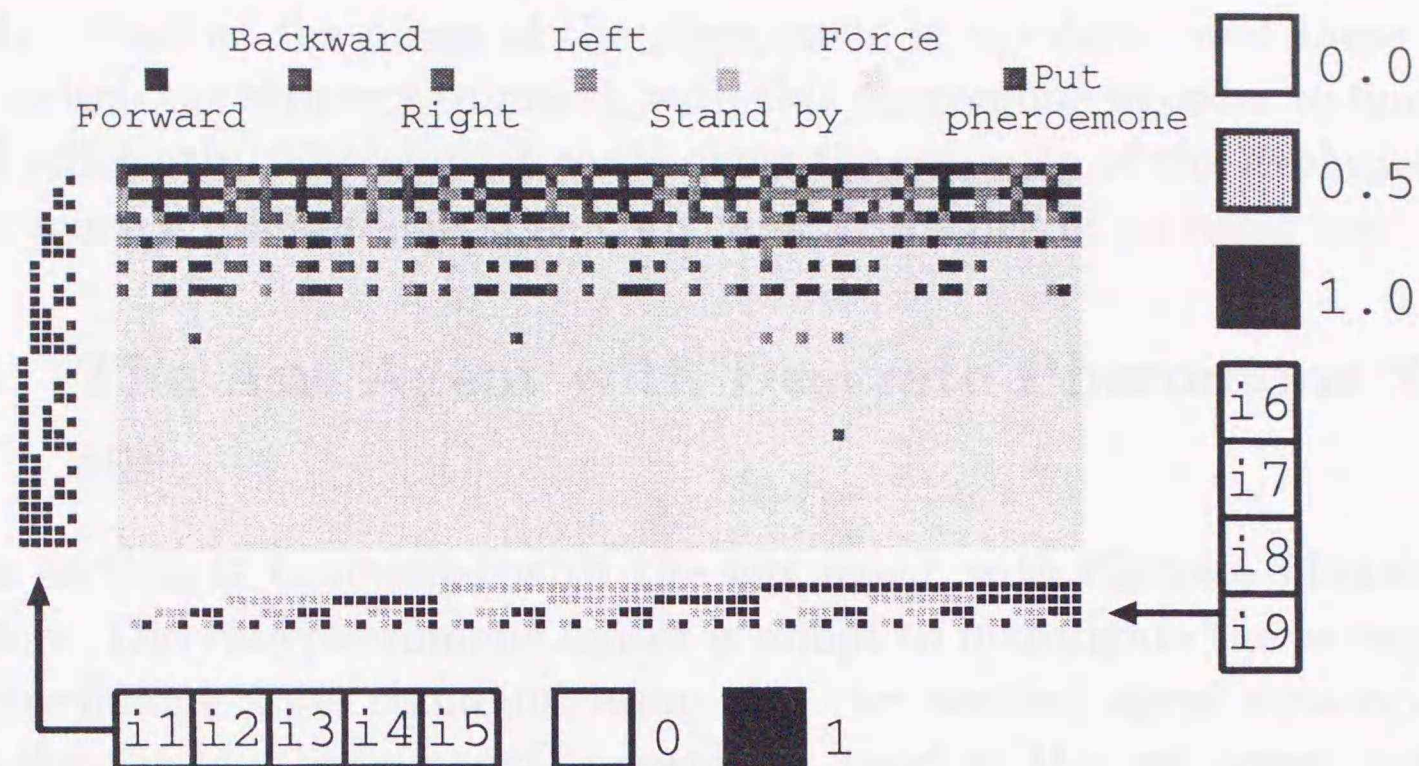


Figure 4.10: The inputs-actions bitmap.

For investigating which types of ants is better than the others, the final populations of each ants were fought with each other as evaluating the populations with $400(N \times N)$ matches. The results are as Type 1(372 wins) - Type 2(28 wins), Type 2(8 wins) - Type 3(392 wins) and 1(20 wins) - Type 3(380 wins). The reason that type 2 is very weak is that they are poor to find the food items and cooperate for forcing although they use the primitive pheromone style communication. Type 3 is strongest in 3 types as I expected, and it is said that the good communication as type 3 is effective for the cooperation to win.

Here, I make an investigation into the strategy of type 3 ants which showed the best performance and cooperation with the pheromone style communication. The inputs - actions bitmap of these ants is shown in Fig. 4.10. The vertical indicates the states of 5 inputs, $i_1, \dots, i_5$. The horizontal also indicates the states of 4 inputs from the pheromone, $i_6, \dots, i_9$, where the values of the inputs are divided into three values, 0.0, 0.5 and 1.0 because these values are real values and it is not possible to show the map by using real values directly. The color of pixels corresponds to the action having the highest probability to be selected in 9 actions. As shown, the actions are various according to the intensities of the pheromone. Especially, it is confirmed that this type of ants tend to take the forcing action in the situation that they touch the food item or the fellows in right and left, the action to move backward in that they sense little intensities of the pheromone in right and left, and the action

to move forward in sensing some intensities of the pheromone respectively. That is, the mean of the pheromone is repulsive, and these ants can switch the strategy to march with this pheromone in order to find the food efficiently. Therefore, I could show the one case of the evolutionary emergence of pheromone style communications out of no meanings.

4.5 The Ant Agent with Discrete Pheromone Sensor

This section is concerned with the ant agent with discrete pheromone sensors. Discrete pheromone sensor is adopted to investigate the emergence of pheromone style communication and the evolved agent strategy. In addition, a two layer neural network is used in the ant agent instead of a three layer in the above subsection. This simplification is adopted for investigating about the evolution of strategy and communication, although it is possible to make the ant agent evolve with complicated sensors and architectures by using of evolutionary computation or machine learning techniques. This section contains the alterations from continuous pheromone sensors to discrete pheromone sensors, and the interesting experimental results.

4.5.1 The Agent Architecture

Some components of the agent architecture are different from the continuous pheromone sensor. First, the sensors of the ant agent is described. The sensory inputs of the agent on position (x, y) are defined as follows.

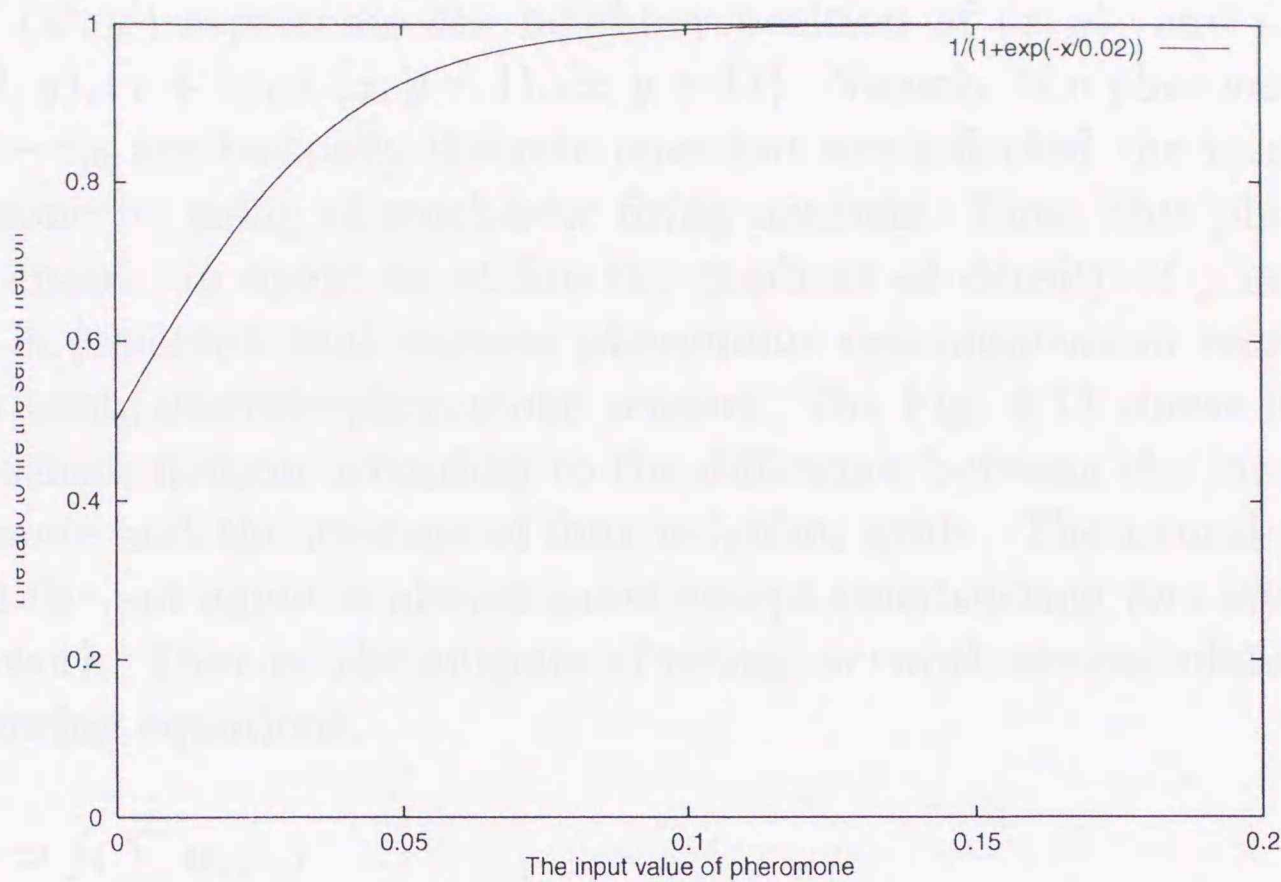


Figure 4.11: The ratio to fire the sensor neuron according to the input value of pheromone.

$$i_1 = \begin{cases} 1 & \text{if the agent touches the food item} \\ 0 & \text{otherwise} \end{cases}$$

$$i_{2-5} = \begin{cases} 1 & \text{if the agent is in contact with the fellow agent in the} \\ & \text{\{front, back, right, left\} grid} \\ 0 & \text{otherwise} \end{cases}$$

$$i_6 = \begin{cases} 1 & \text{if the agent is in contact with the fellow forcing agent} \\ 0 & \text{otherwise} \end{cases}$$

$$i_{7-10} = \begin{cases} 1 & \text{if the probability } Prob(x', y') \text{ in \{front, back, right, left\}} \\ & \text{is satisfied.} \\ 0 & \text{otherwise} \end{cases}$$

$$i_{11} = -1 \text{ (bias)}$$

where, $Prob(x', y')$ is the probability to fire the sensor neuron based on the intensity of pheromone, and is defined as follows.

$$Prob(x', y') = 1/(1 + \exp(-(P(x', y', 0) - P^*(x, y, 0))/T)) \quad (4.15)$$

$$P^*(x, y, 0) = (P(x-1, y, 0) + P(x+1, y, 0) + P(x, y-1, 0) + P(x, y+1, 0))/4 \quad (4.16)$$

where, (x', y') represents the neighbor position of (x, y) , and $(x', y') \in \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$. Namely, the pheromone sensors $i_7 - i_{10}$ are basically discrete ones but are reflected the intensity of pheromone by using of stochastic firing neurons. Thus, this pheromone sensor enable an agent to utilize the gradient of density of pheromone, and it is expected that various pheromone communication emerges although using discrete pheromone sensors. The Fig. 4.11 shows the ratio to fire sensor neuron according to the difference between the intensity of pheromone and the average of four neighbor grids. The neural network used in the ant agent is almost same except constructing two layers neural network. That is, the outputs of neural network are calculated based on following equations.

$$o_i = f\left(\sum_{j=1}^{11} w_{ij} i_j\right) \quad (4.17)$$

$$f(x) = 1/(1 + \exp(-x)) \quad (4.18)$$

This simplification is for the convenience investigation of evolved neural network through evolutionary process. The action is decided according to the same roulette wheel strategy used in the continuous pheromone sensor architecture. The used genetic algorithm is not changed.

4.5.2 Computer Experiments

Several computer experiments were made according to the above settings. Throughout experiments, about two types of pheromone style communication emerged, namely attractive pheromone and repulsive pheromone. Attractive pheromone is diffused by the ant touching the food item, and an ant sensing attractive pheromone moves towards the direction to strong pheromone according to the gradient of density. Thus, many ants can efficiently gather near the food item. On the other hand, repulsive pheromone is randomly diffused by every ant, and an ant tends to avoid strong pheromone. Therefore, an ant can reach the food items more efficiently than without repulsive pheromone because an ant can avoid once searched area of the grids by the effect of repulsive pheromone. The details of both experiments are described following subsections.

4.5.2.1 Result 1 about Attractive Pheromone

This subsection describes the experimental results about pheromone style communication which was emerged through evolutionary process. As

result 1, pheromone style communication based on attractive pheromone is represented. Fig. 4.12 and 4.13 shows the screen shots of the ants war with agents in the final generation of genetic algorithm. The figure (a) represents the initial state of ants war, in which two kinds of ant agents are placed randomly, and five food items are placed in the center of arena. In each screen shot from (b) to (i), it is observed that ants touching the food item diffuse attractive pheromone, and an ant sensing it attracts to the food items based on the gradient of density. Especially, figure (f) shows the situation that the food items have been carried out and disappeared from the grid world. The attractive pheromone around near disappeared food item evaporated, and already redundant pheromone automatically disappeared.

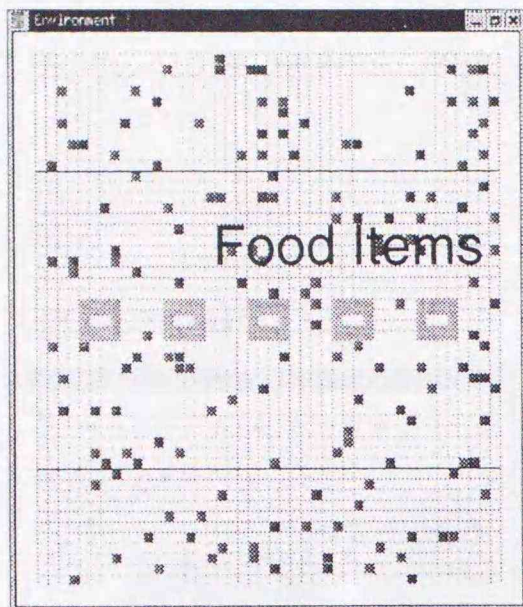
Next, it is considered whether the evolution of ant agents rightly occurs or not. The measurement of degree of evolution is difficult because implemented genetic algorithm has no global fitness function or evaluation function. Therefore, I supposed that ant agents in final generation are strongest in all generation agents, and adopted the idea to measure the degree of evolution by that ant agents in each GA generation are fought to agents in final GA generation. Namely, the degree of evolution is indicated by the ratio to win against ant agents in final generation. The graph in 4.14 represents the ratio to win against final generation, and x-axis indicates the number of GA generation. The ratio to win is increased according to the progress of generation, and it can be considered that the evolution based on implemented simple genetic algorithm works effectively. In final generation, the ratio takes the value of 0.5, that means the strength of ant agents are even.

Next, the evolution is investigated in another viewpoint. It is difficult to investigate the emergence of communication because it is also difficult to define how communication is. The emergence of intent of pheromone is considered. To evaluate the emergence of intent of pheromone is easier than to do the emergence of communication because it can be achieved by measuring the effectiveness of pheromone in selecting actions. Thus, the shannon's entropy and the information value are introduced. Let $S_i (i = 1, 2, \dots, 7)$ be a selected action, and $P(S_i)$ be a probability to select action i through a game of ants war. The entropy of outputs $H(S)$ is represented as follows.

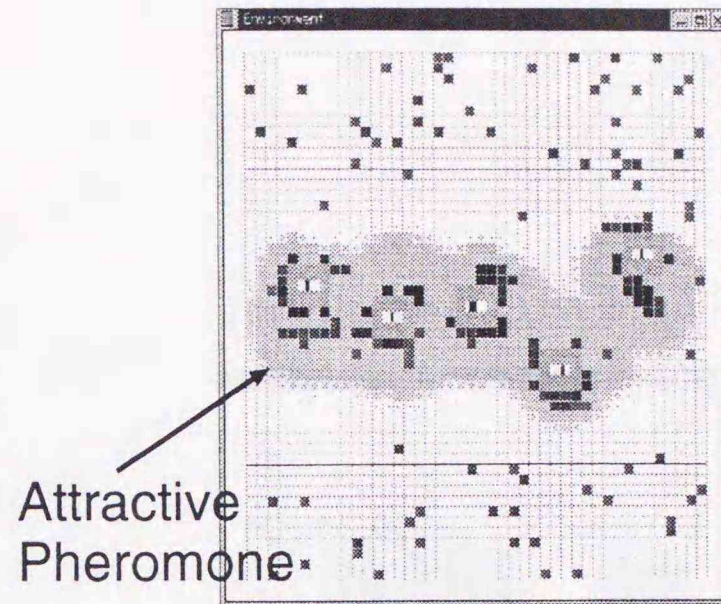
$$H(S) = - \sum_i P(S_i) \log P(S_i) \quad (4.19)$$

The patterns of sensor inputs are classified according to the sensor inputs from the environment ($i_1 - i_6$) or pheromone ($i_7 - i_{10}$). Concerning with

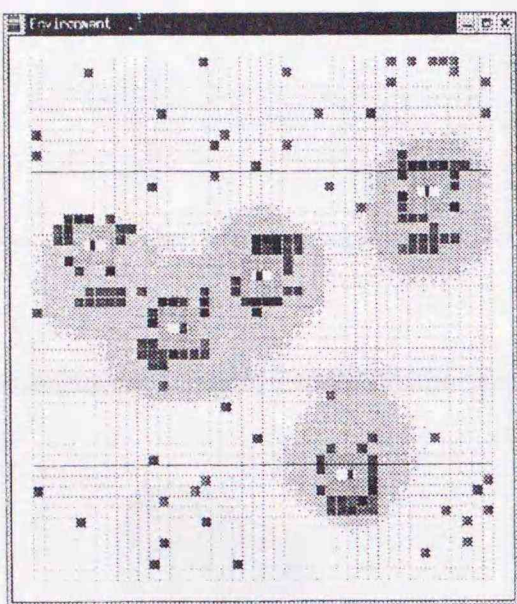
(a) Initial state



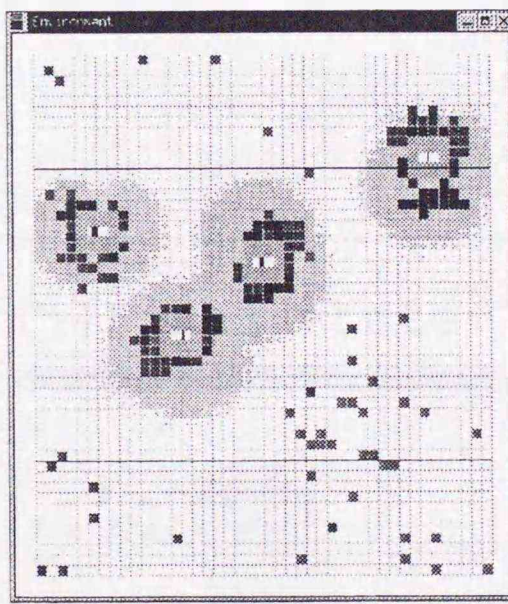
(b) $t=100$



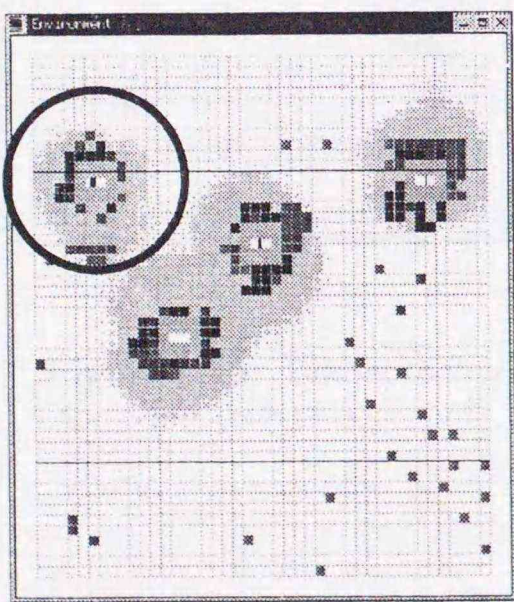
(c) $t=200$



(d) $t=300$



(e) $t=400$



(f) $t=500$

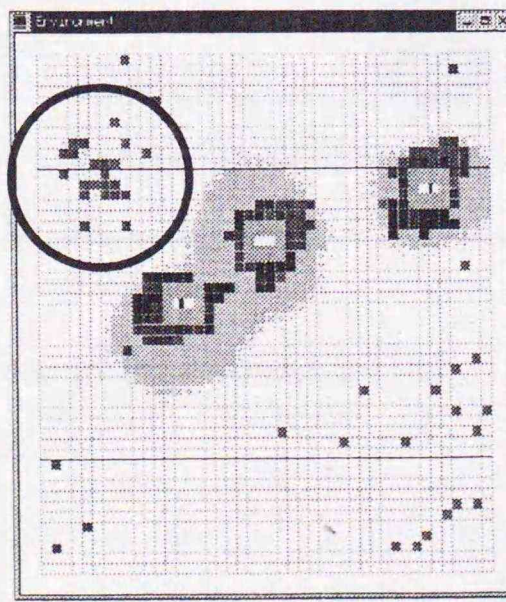
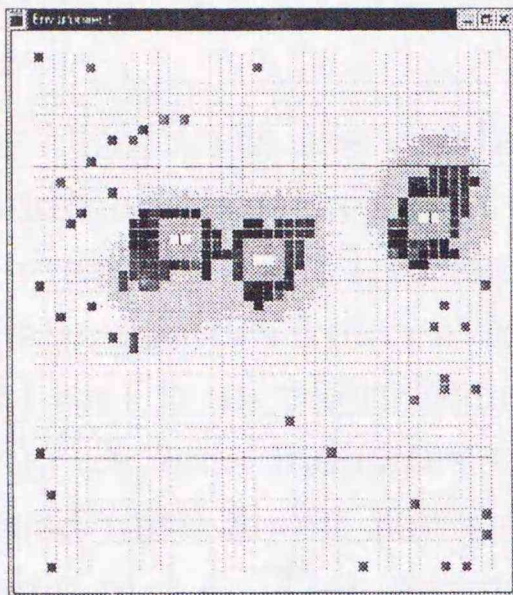
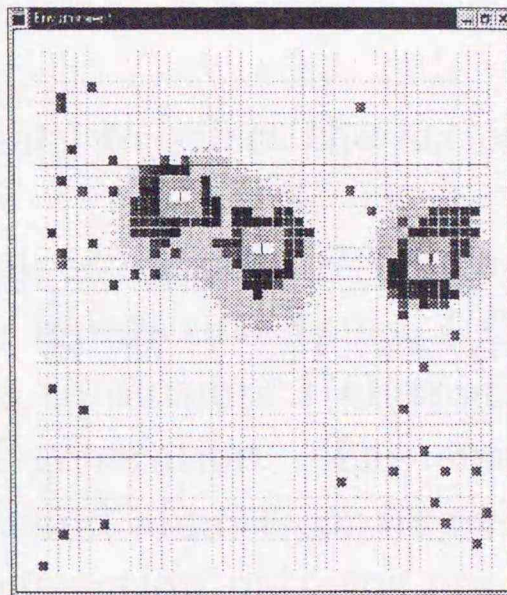


Figure 4.12: The screen shots of the ants war in experiment 2-1(a).

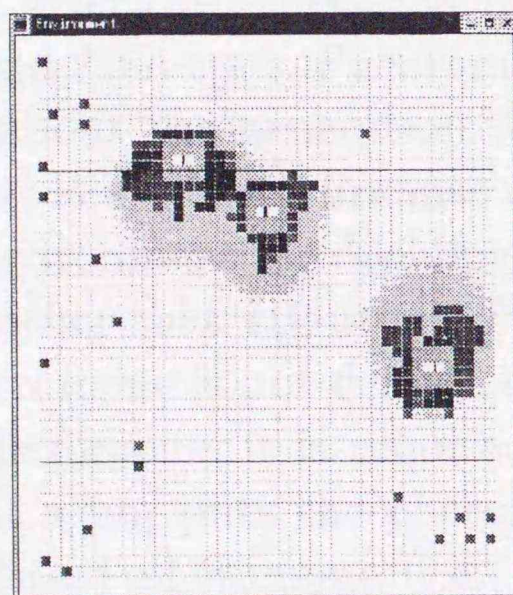
(g) $t=600$



(h) $t=700$



(i) $t=800$



(j) Final state

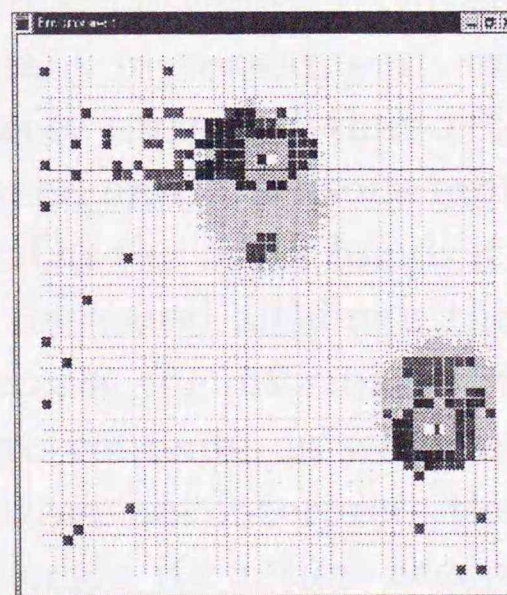


Figure 4.13: The screen shots of the ants war in experiment 2-1(b).

the sensor inputs from the environment (i_1-i_6), there are 64 patterns (2^6), and there are 16 patterns (2^4) with the sensor inputs from the pheromone. Let $\hat{S}_j$ be the j -th pattern of sensor inputs, and the information value of $\hat{S}$, $I(S, \hat{S})$ is defined as next.

$$I(S, \hat{S}) = H(S) - H(S|\hat{S}) \quad (4.20)$$

$$H(S|\hat{S}) = \sum_j P(\hat{S}_j) H(S|\hat{S}_j) \quad (4.21)$$

$$H(S|\hat{S}_j) = - \sum_i P(S_i|\hat{S}_j) \log P(S_i|\hat{S}_j) \quad (4.22)$$

where, $P(S_i|\hat{S}_j)$ is the conditional probability to select the action S_i in the condition that the input pattern is $\hat{S}_j$.

The graph in Fig. 4.15 shows the entropy of outputs $H(\text{Output})$, the information values from environment sensor inputs to outputs, $I(\text{Output}, \text{Sensor})$ and from pheromone sensor inputs to outputs $I(\text{Output}, \text{Pheromone})$. X axis indicates the number of generation. The entropy is decreased until about the generation 7000, and almost converged about the value of 1.4. This means that the input-action patterns are almost formed before the generation 7000, and the patterns are minutely adjusted after the generation 7000. The information values $I(\text{Output}, \text{Sensor})$ and $I(\text{Output}, \text{Pheromone})$ are gradually increased, and converged after about the generation 30000. It is easily imagined that $I(\text{Output}, \text{Sensor})$ is increased because ant agents depend on the sensory input from the environment for deciding their actions. On the other hand, it is very interesting that $I(\text{Output}, \text{Pheromone})$ is increased although the intent of pheromone is not defined at first. To increase the information value of the pheromone, it is required to specify two contents, in what situation an ant agent puts the pheromone, and how an ant interprets the intent of diffused pheromone (i.e., how an ant decide its action according to sensing pheromone). Moreover, if the pheromone have no clear intent, diffused pheromone works as noise for agents. Both of these are unclear at first stage of evolutionary process, then, may be gradually formed by the affection of bias of initial neural networks. According to this result, the pheromone affect the behavior of ant agents, and it can be said that the pheromone have a kind of "intent". Of course the intent is attractive in this experiment. The graph in Fig. 4.16 represents the transition of average ratio to select the action to put the pheromone. At first generation, the ratio takes high value because the neural networks are randomly

generated, and ant agents select random actions. After few thousand generations, the ratio is converged about the range of 0.03 – 0.05.

Following the above results, the emergence of intent of pheromone is confirmed, but the emergence of communication is not done. To investigate the emergence of communication, the additional experiments were made. At first, two kinds of ant agents were prepared. The first ant agents were normal ones, and the second agents were also normal except forbidden to put the pheromone. Using these two kinds of agents, several trials of ants war were done in changing the number of agents from 5 to 35. Trials of each number were done 30 times for averaging. The graph in Fig. 4.17 shows the average ratio to win against forbidden agents, and x axis is the number of generation. At the number of 1, the ratio represents about 0.7 and not 0.5. The reason the ratio didn't take 0.5 is that the attractive pheromone put by only single ant agent has the effect to clip the ant agent to the position on which the pheromone was put. Namely, the ant which puts the pheromone near the food item tends to be near the food, and can recovery to find the food again if loses sight of the food. Then, the ratio to win is increased according to the number of agents, and converges the value of 1 exceeding 30. This result shows the pheromone have more effectiveness in larger number than in small number, namely it is enough to say that the pheromone style communication emerged in the evolutionary process. If the pheromone is used by an ant agent for only itself, the ratio to win doesn't change according to the number of agents.

The emergence of pheromone style communication was confirmed, therefore the evolved strategy are investigated as next. It is easy to reveal the evolved strategy because the agent architecture in this section is based on two layers neural networks. Thus, I can confirm the effects from sensory inputs to outputted actions by checking the weight sets from input layers to output layers. It is simply considered that if the weight from the input i to the output j is positive value, the input i excites selecting the output j for the ant agent, oppositely, if the weight is negative value, the input i inhibits selecting the output j . Moreover, the probability to select the output j is decided according to the sigmoid function whose input is calculated as the sum of connected weights, and the effect of each input is easily confirmed. If the agent architecture is complicated, for example, using three layers neural networks, Q-learning method, etc, to investigate obtained strategy is difficult.

Fig. 4.18 - Fig. 4.21 represent the weights from each input to 7 outputs

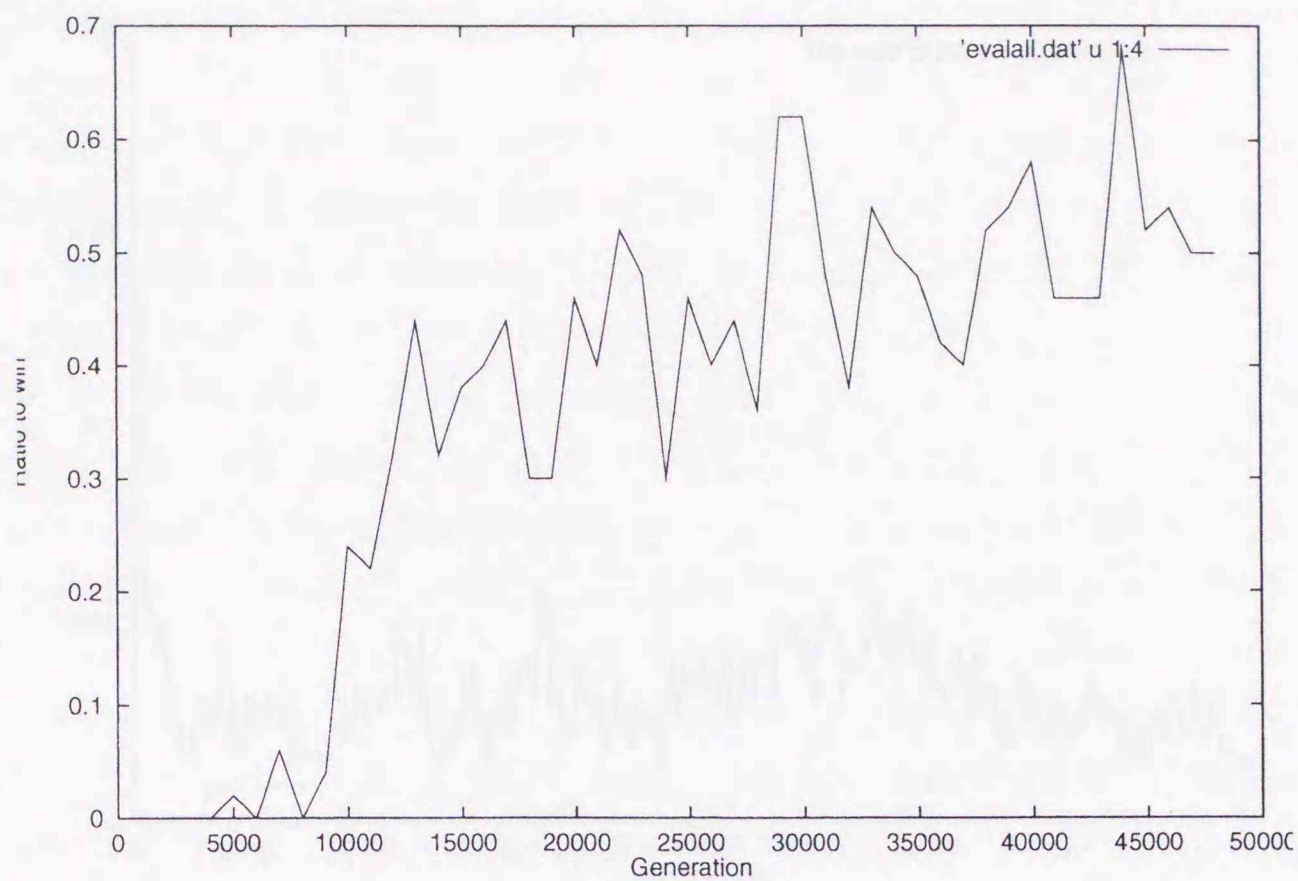


Figure 4.14: The ratio to win against the final population in the experiment 2-1.

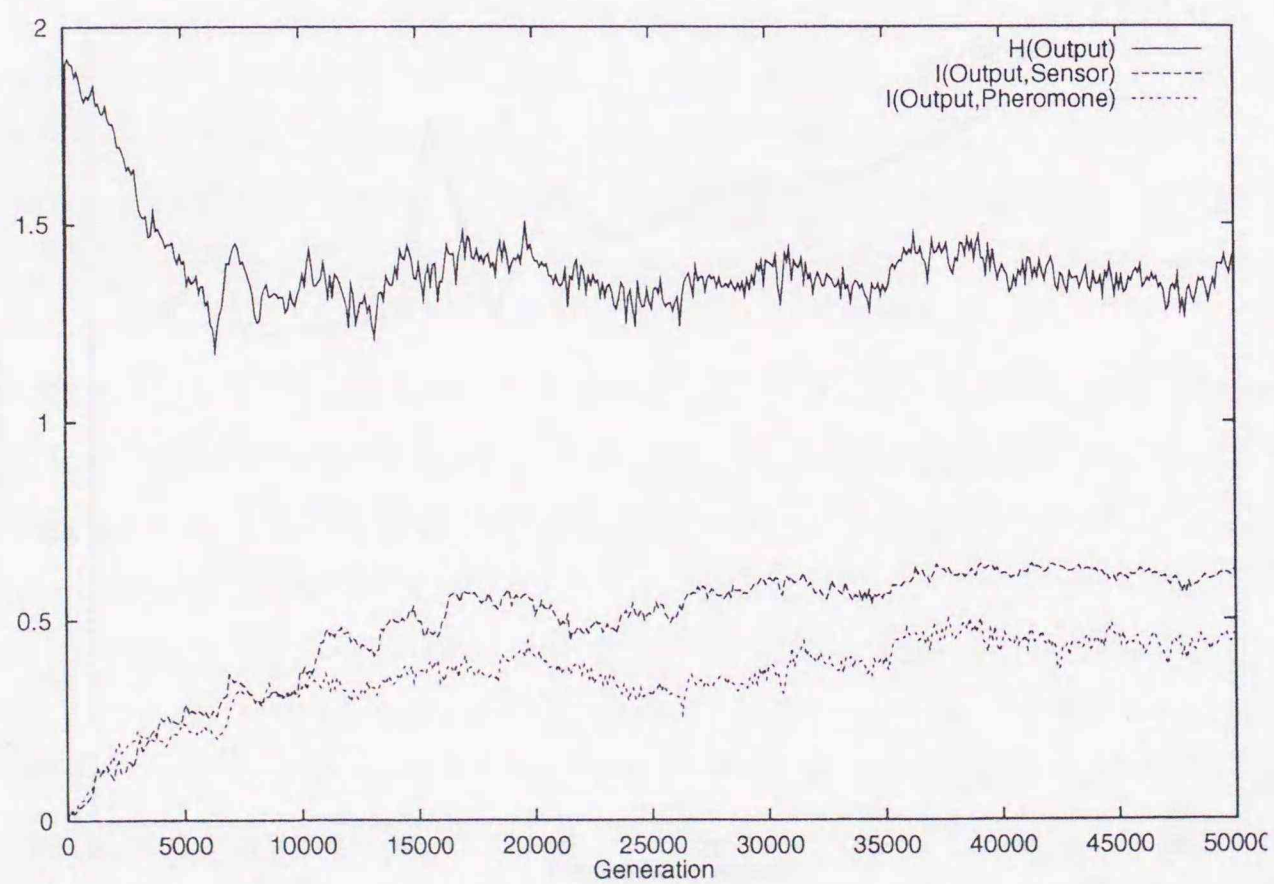


Figure 4.15: The transitions of $H(\text{Output})$, $I(\text{Output}, \text{Sensor})$, and $I(\text{Output}, \text{Pheromone})$ in experiment 2-1.

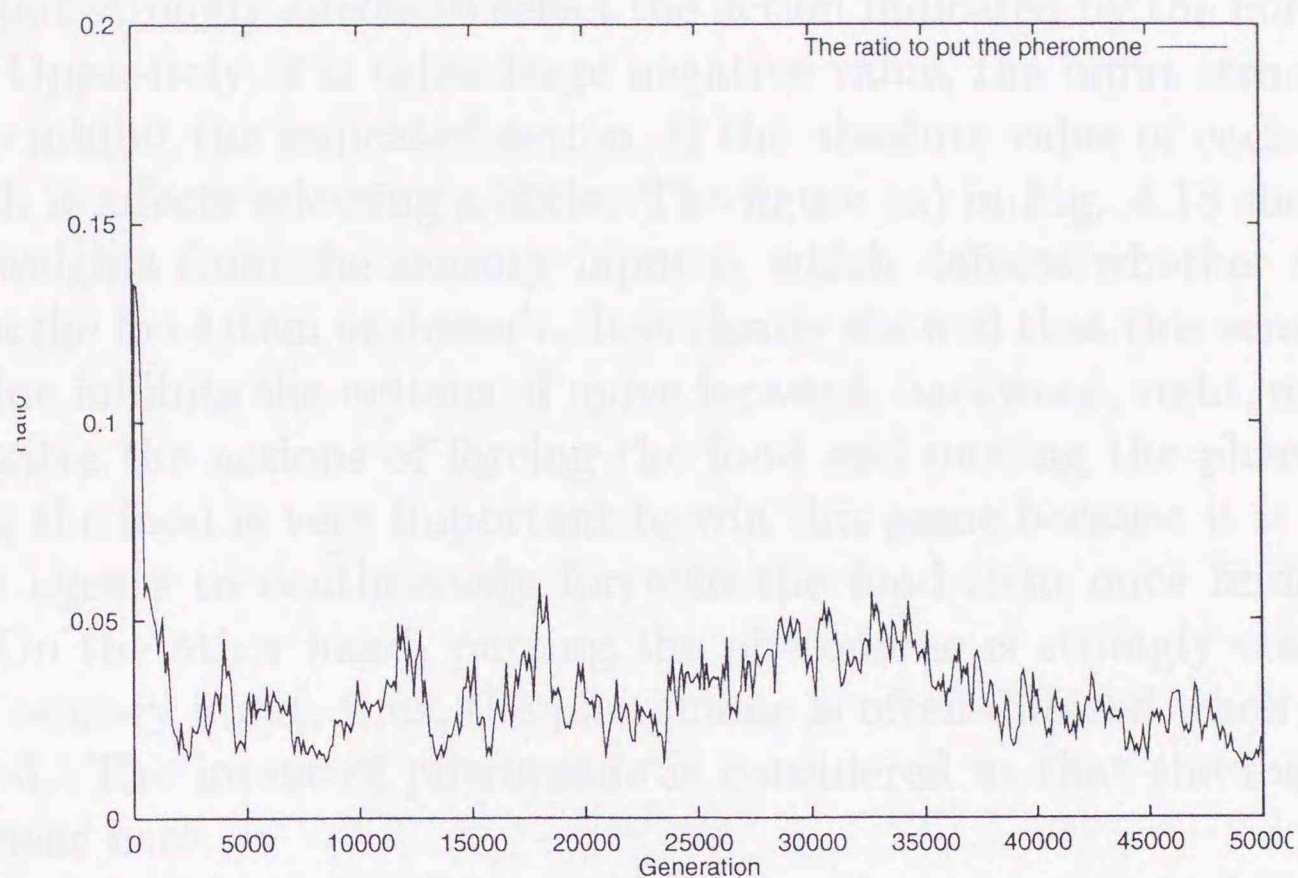


Figure 4.16: The transition of the ratio to put the pheromone in the experiment 2-1.

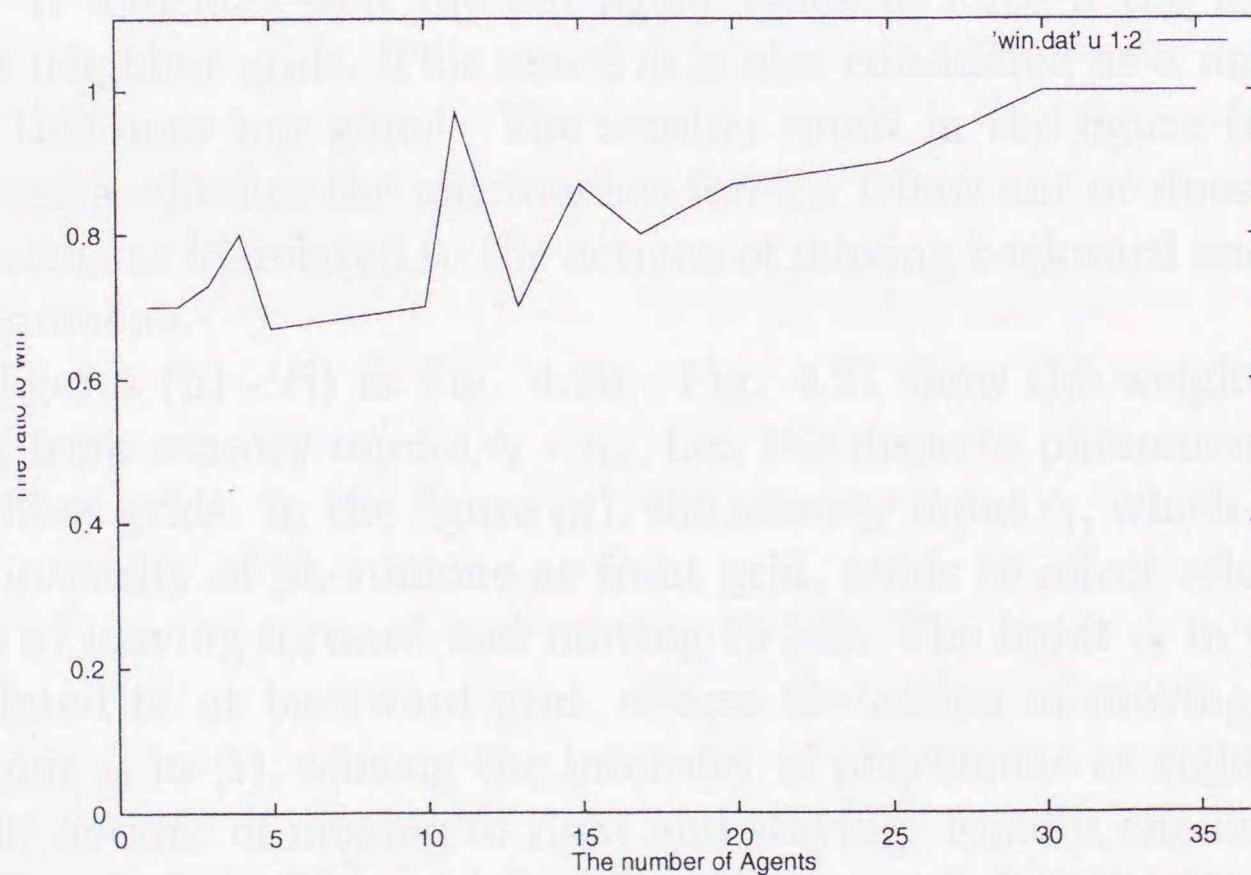


Figure 4.17: The ratio to win against the ant agents which were forbidden to put the pheromone in the experiment 2-1.

as selected actions. The horizontal axis in each figure indicates the number of actions (see Tab. 4.1), and the vertical axis indicates the value of weights. If the value of each weight takes large positive value, the sensory input strongly affects to select the action indicated by the horizontal index. Oppositely, if it takes large negative value, the input strongly affects to inhibit the indicated action. If the absolute value of each weight is small, it affects selecting a little. The figure (a) in Fig. 4.18 shows the set of weights from the sensory input i_1 which detects whether the ant touches the food item or doesn't. It is clearly showed that this sensory input value inhibits the actions of move forward, backward, right, and left, and excites the actions of forcing the food and putting the pheromone. Forcing the food is very important to win this game because it is needed for ant agents to continuously force to the food item once finding the food. On the other hand, putting the pheromone is strongly connected to this sensory input, thus, the pheromone is often diffused when finding the food. The intent of pheromone is considered as that the food item exists near here.

The figures (b) - (e) in Fig. 4.18 - Fig. 4.19 concern with the sensory inputs of $i_2 - i_5$, that is, whether fellow ant agent exists at neighbor grids or doesn't. In each figure, the weights to the forcing action take largest values. It indicates that the ant agent tends to force if the fellow ants exist at neighbor grids. This reaction is also considered as a natural one to win this ants war game. The sensory input in the figure (f) in Fig. 4.19 detects whether the ant touches forcing fellow ant or doesn't. This input seems to be related to the actions of moving backward and putting the pheromone.

The figures (h) - (j) in Fig. 4.20 - Fig. 4.21 show the weight set connecting from sensory inputs $i_7 - i_{10}$, i.e., the discrete pheromone sensors at neighbor grids. In the figure (g), the sensory input i_7 , which is related to the intensity of pheromone at front grid, tends to affect selecting the actions of moving forward and moving to left. The input i_8 in the figure (h), related to at backward grid, affects the action of moving to right. The input i_9 in (i), sensing the intensity of pheromone at right grid, affects the actions of moving to right and staying. Finally, the input i_{10} in (i) of Fig. 4.21, sensing at left grid, affects to select the action of moving to left. Resultingly, this ant agent tends to prefer moving to strong pheromone. These analysis of weights correspond to the results observed in the screen shots.

The graphs in Fig. 4.22 and Fig. 4.23 represent the intensities of two

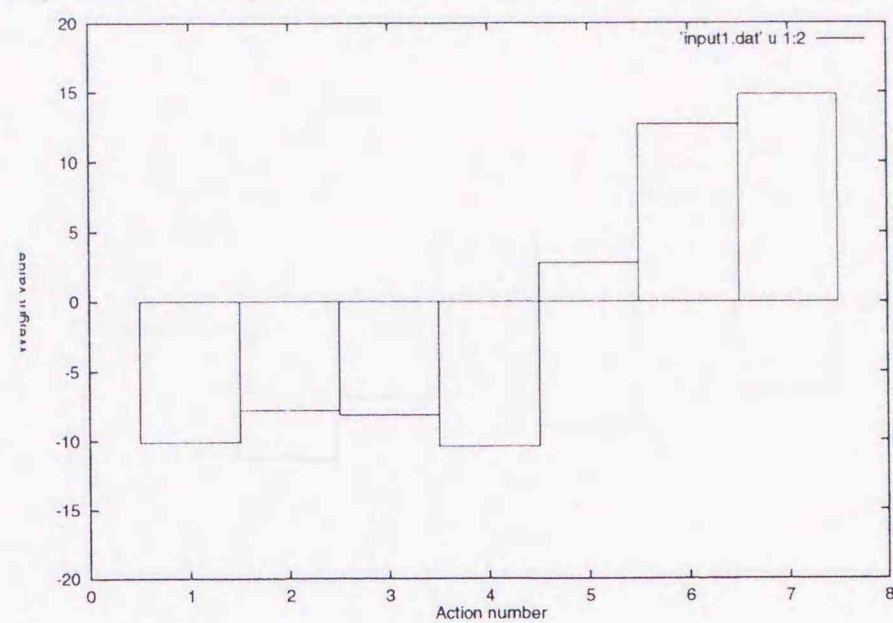
Action no.	Action
1	Move forward
2	Move backward
3	Move to right
4	Move to left
5	Stay at the current grid
6	Force to the food item or the neighbor forcing ant
7	Put the pheromone on the ground of current grid

Table 4.1: The Action and action number on ants war.

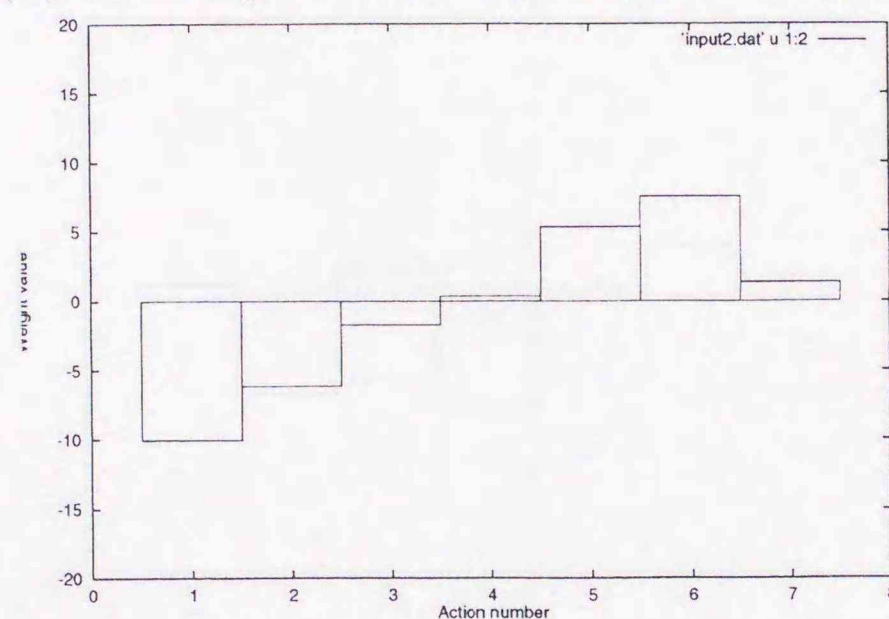
agents' pheromones, $P(x, y, 0)$, and the position of food item in sectional views in the trial of ants war. The vertical axis indicates the intensity of pheromone, and the horizontal one represents the position number of grids. The small squares represent the food item. In the initial state (a), there is no pheromone on grids. After $t = 40$, it is observed that there is strong pheromone near the food item, thus, the gradient of density of pheromone is formed from the food item toward the borders of grids. In addition, Fig. 4.24 represents the contour line of the intensities of pheromone, $P(x, y, 0)$ at time $t = 100$ in the trial, and Fig. 4.25 uses the log scale of Z axis for same time data. The intensities of pheromone take large values pointedly near the food items. From these results, it seems that using the gradient of density of pheromone is valid to communicate each other where the food item is.

The graph in Fig. 4.26 represents the transitions of the ratios of forcing ants and the number of food items in the trial. The horizontal axis indicates the number of game time, and the left vertical axis indicates the value of ratio. In addition, the right vertical axis indicates the number of food items in the grid world. From start time 0 to the time about 230 that one food item is disappeared, the ratio of forcing ants in each agent team is quickly increased. This means that the ants finding food items diffuse the attractive pheromone, and ants sensing the pheromone gather to the pheromone. Especially, it is observed that each ant team could quickly change the target food item after food item disappearing, and the number of forcing ants are recovered in the progress of time.

(a) The weights from input i_1 to 7 outputs



(b) The weights from input i_2 to 7 outputs



(c) The weights from input i_3 to 7 outputs

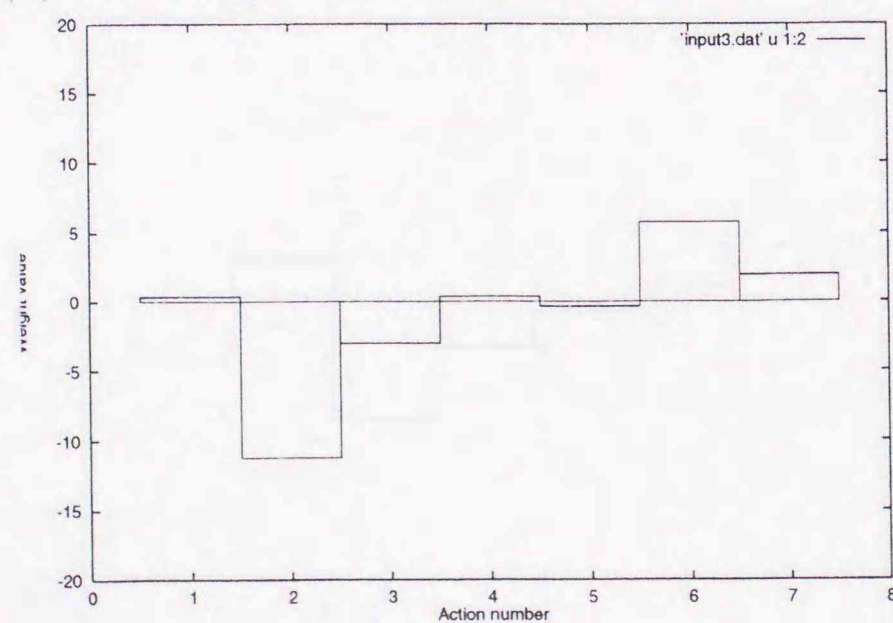
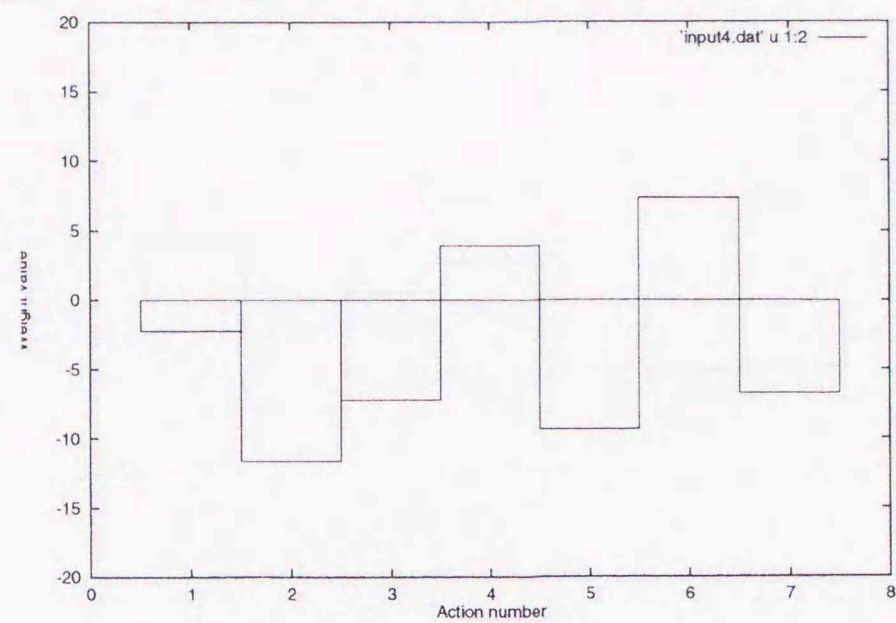
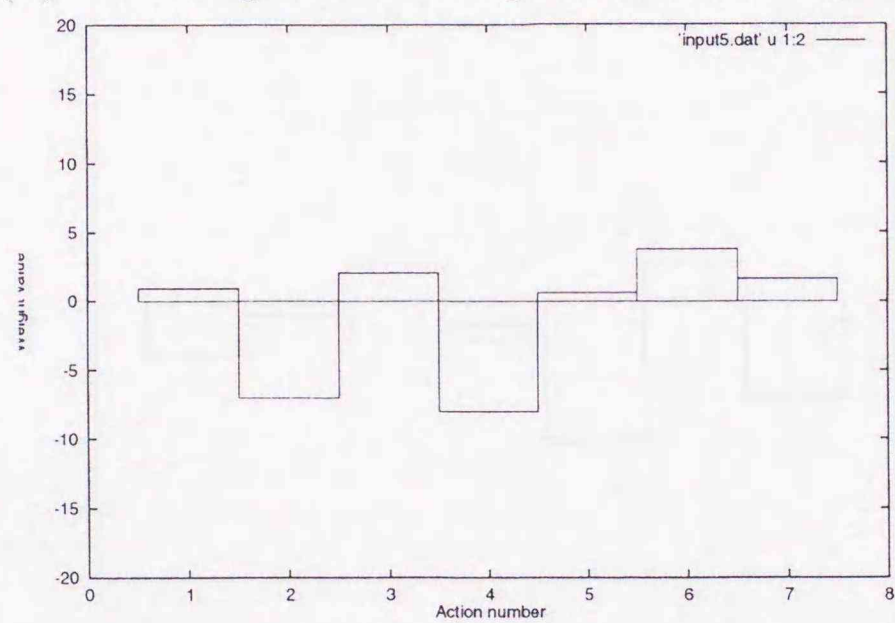


Figure 4.18: The weights from input $i_1 - i_3$ to outputs in the experiment 2-1 (A).

(d) The weights from input i_4 to 7 outputs



(e) The weights from input i_5 to 7 outputs



(f) The weights from input i_6 to 7 outputs

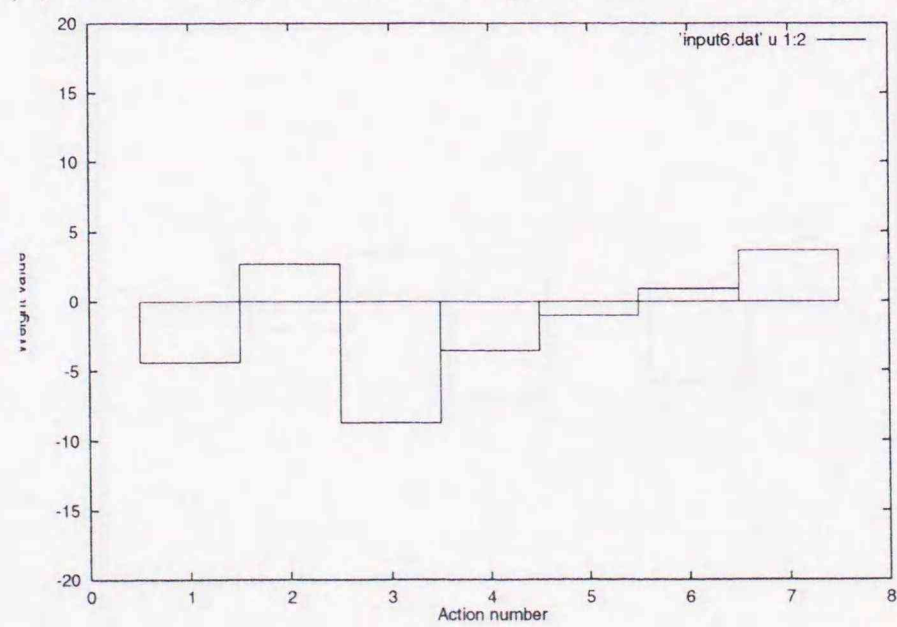
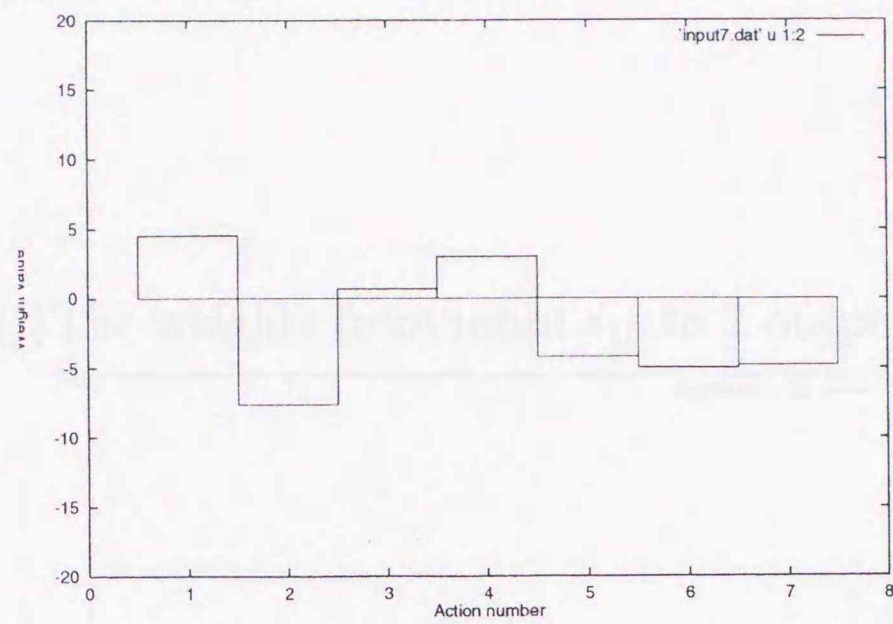
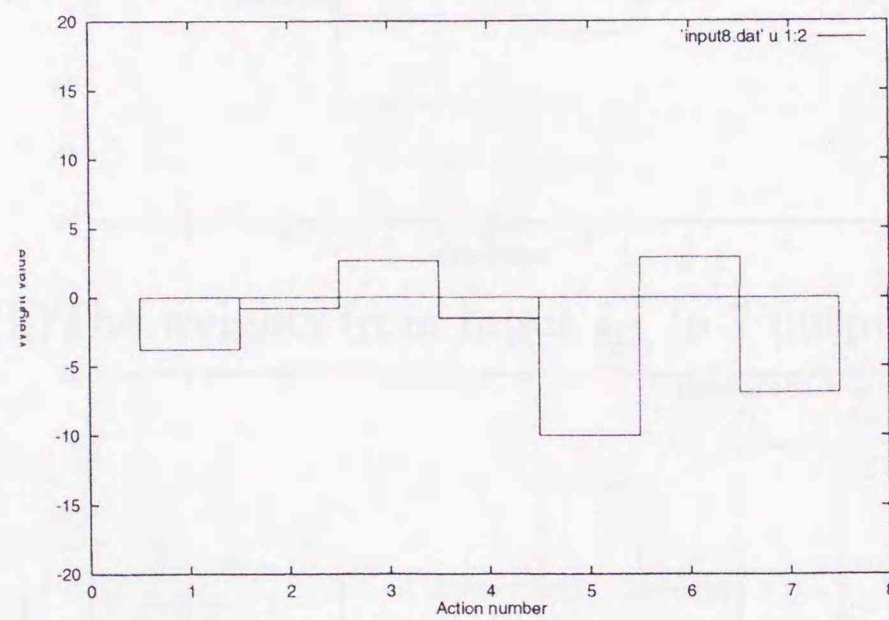


Figure 4.19: The weights from input $i_4 - i_6$ to outputs in the experiment 2-1 (B).

(g) The weights from input i_7 to 7 outputs



(h) The weights from input i_8 to 7 outputs



(i) The weights from input i_9 to 7 outputs

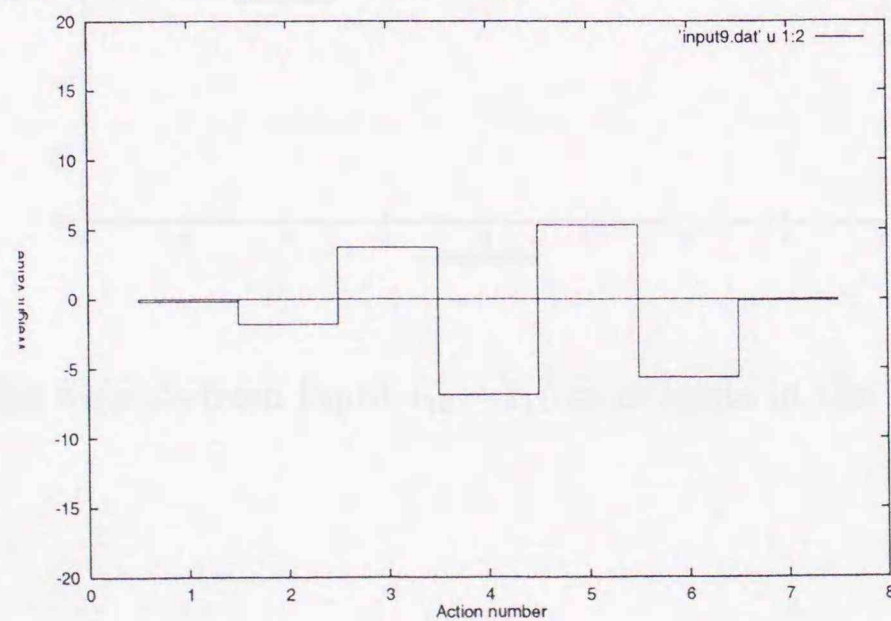
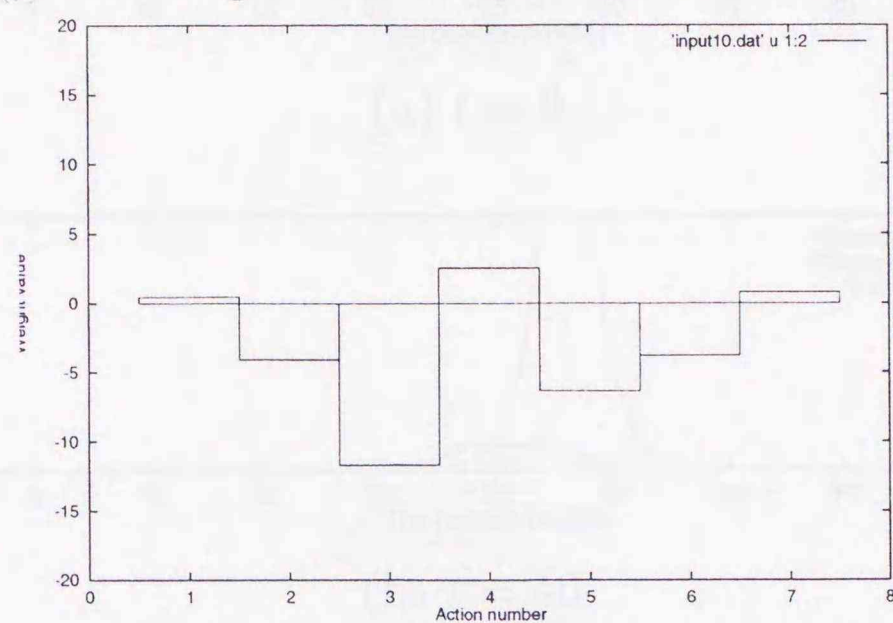


Figure 4.20: The weights from input $i_7 - i_9$ to outputs in the experiment 2-1 (C).

(j) The weights from input i_{10} to 7 outputs



(k) The weights from input i_{11} to 7 outputs

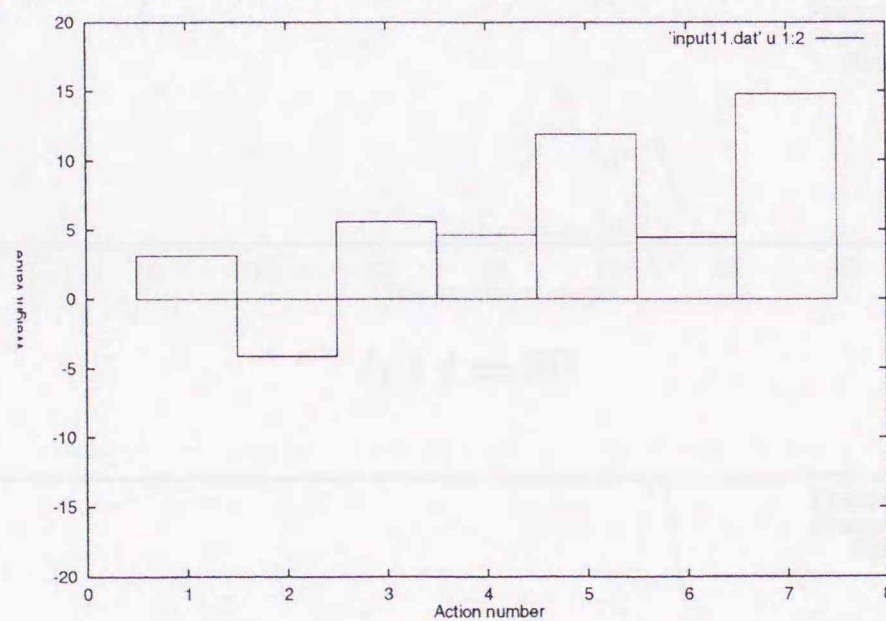
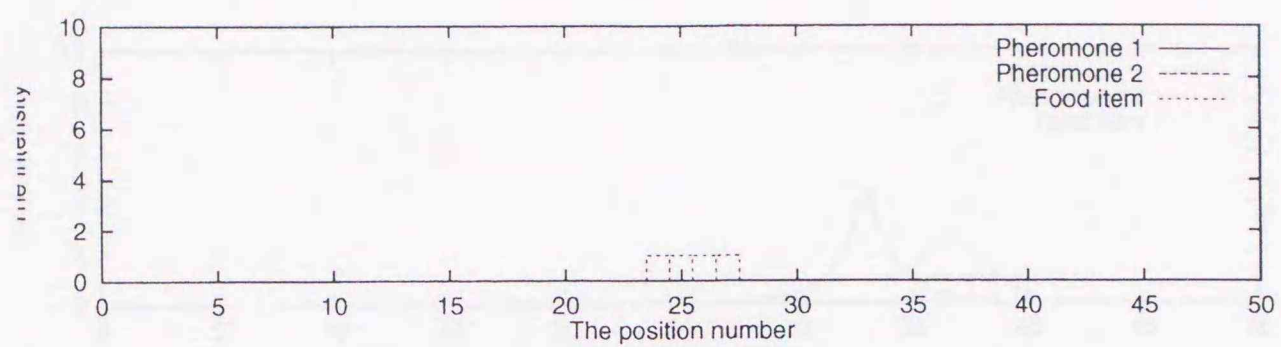
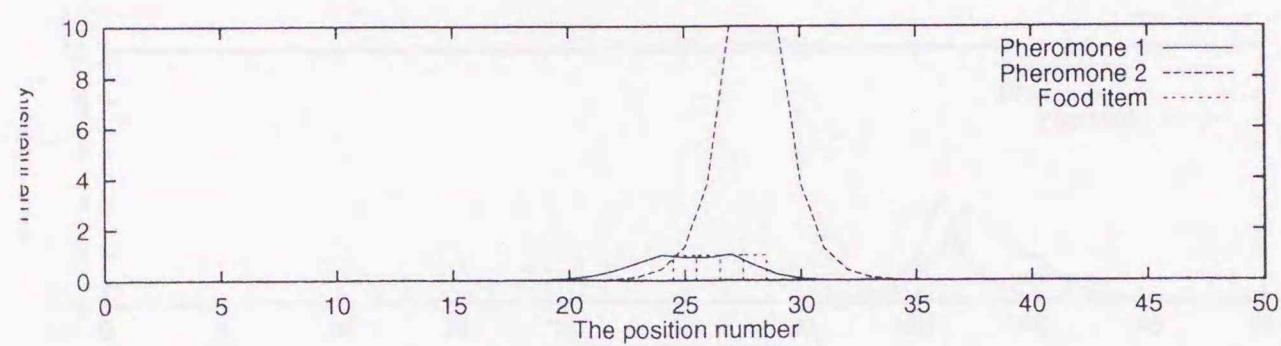


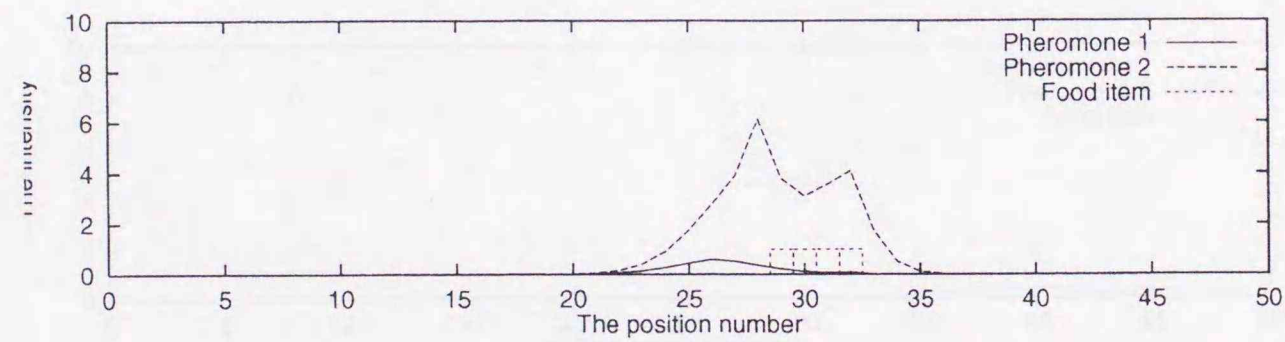
Figure 4.21: The weights from input $i_{10} - i_{11}$ to outputs in the experiment 2-1.



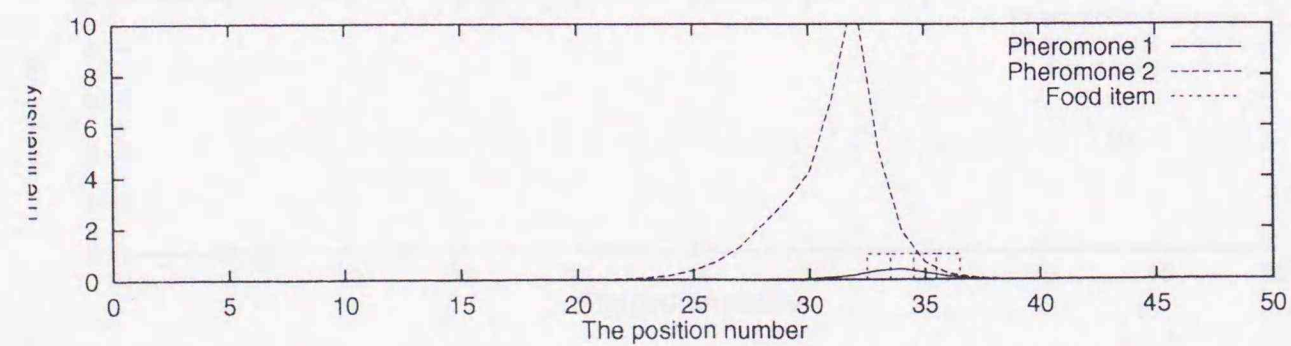
(a) $t = 0$



(b) $t = 40$

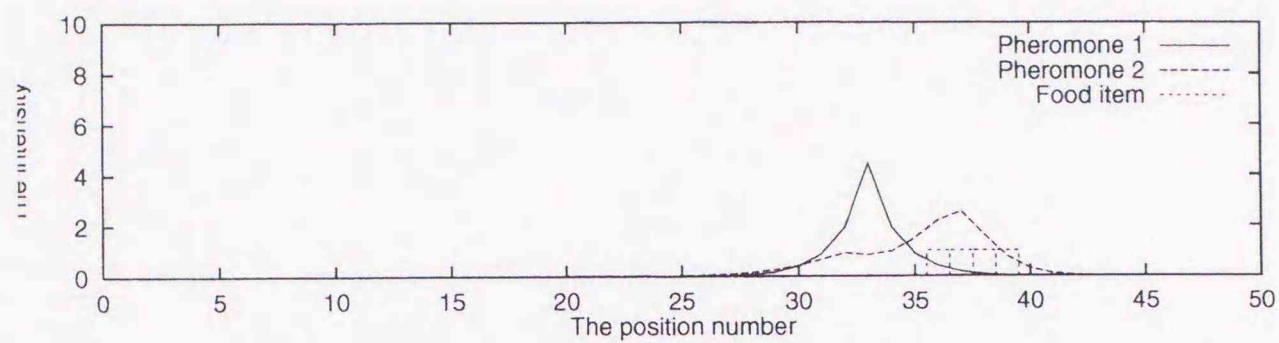


(c) $t = 80$

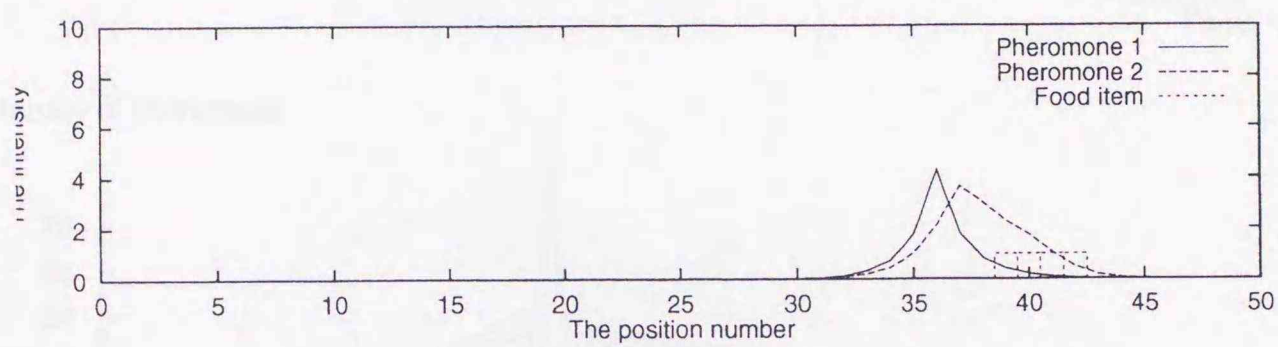


(d) $t = 120$

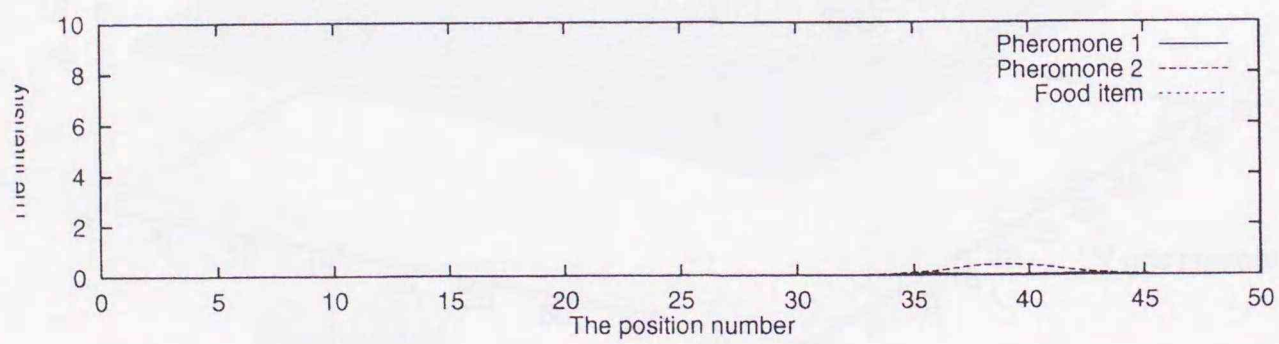
Figure 4.22: The transitions of intensity of pheromone in experiment 2-1 (A).



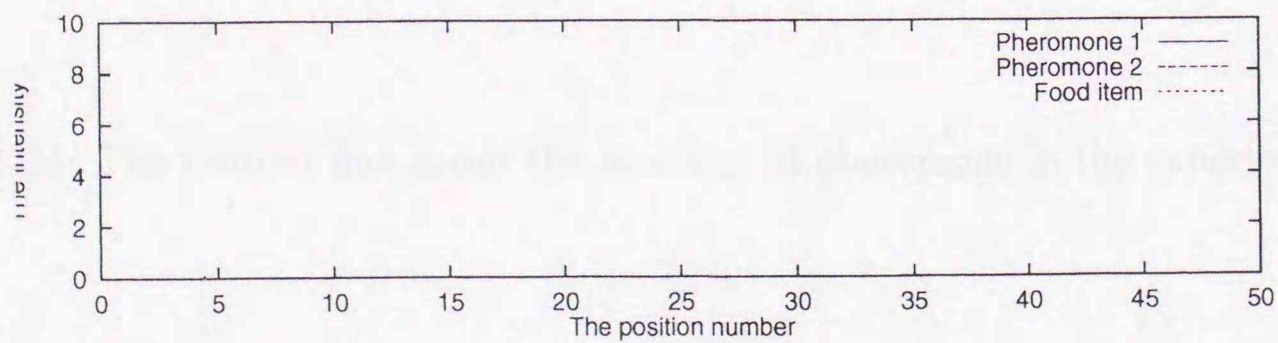
(e) $t = 160$



(f) $t = 200$



(g) $t = 240$



(h) $t = 280$

Figure 4.23: The transitions of intensity of pheromone in experiment 2-1 (B).

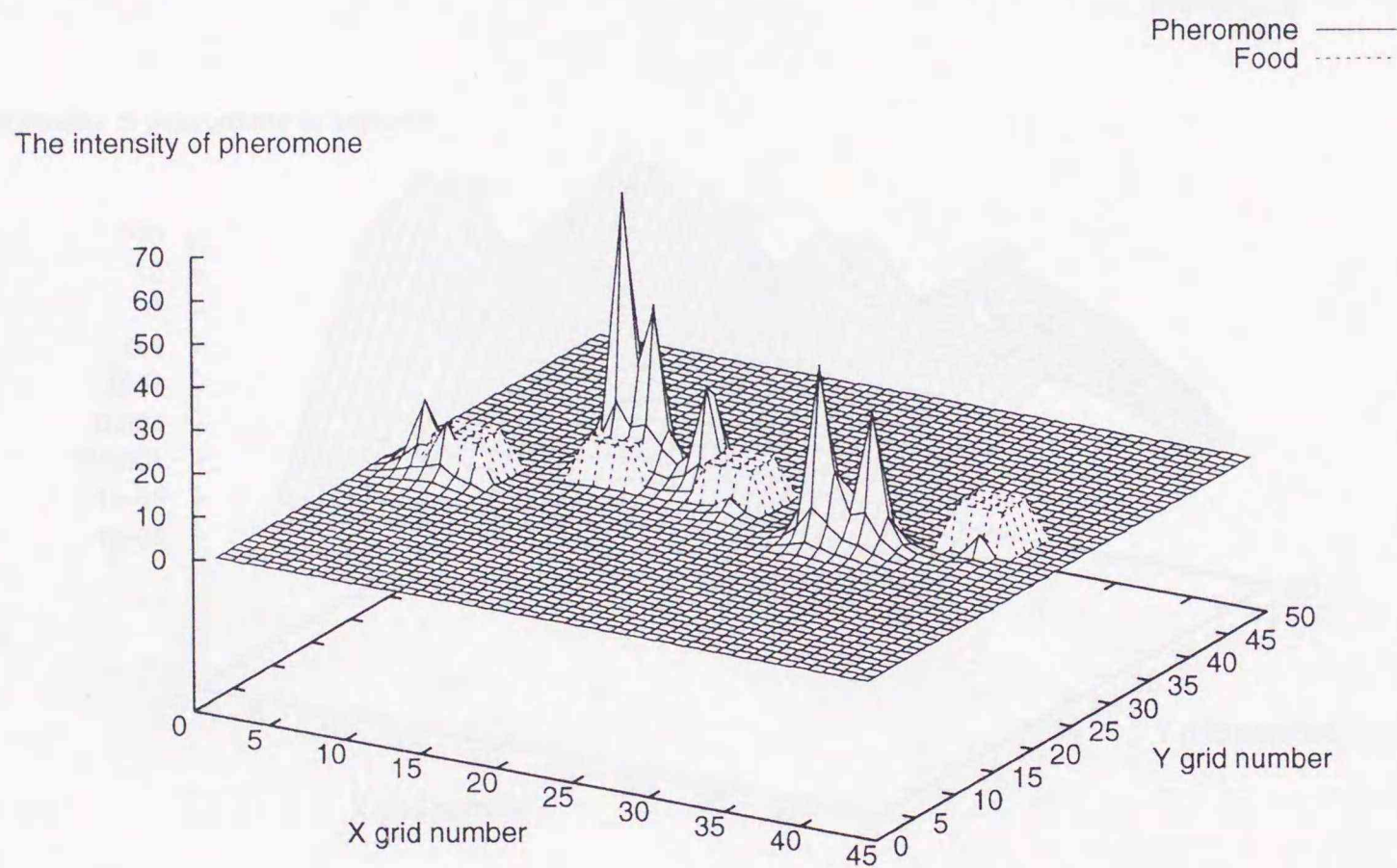


Figure 4.24: The contour line about the intensity of pheromone in the experiment 2-1.

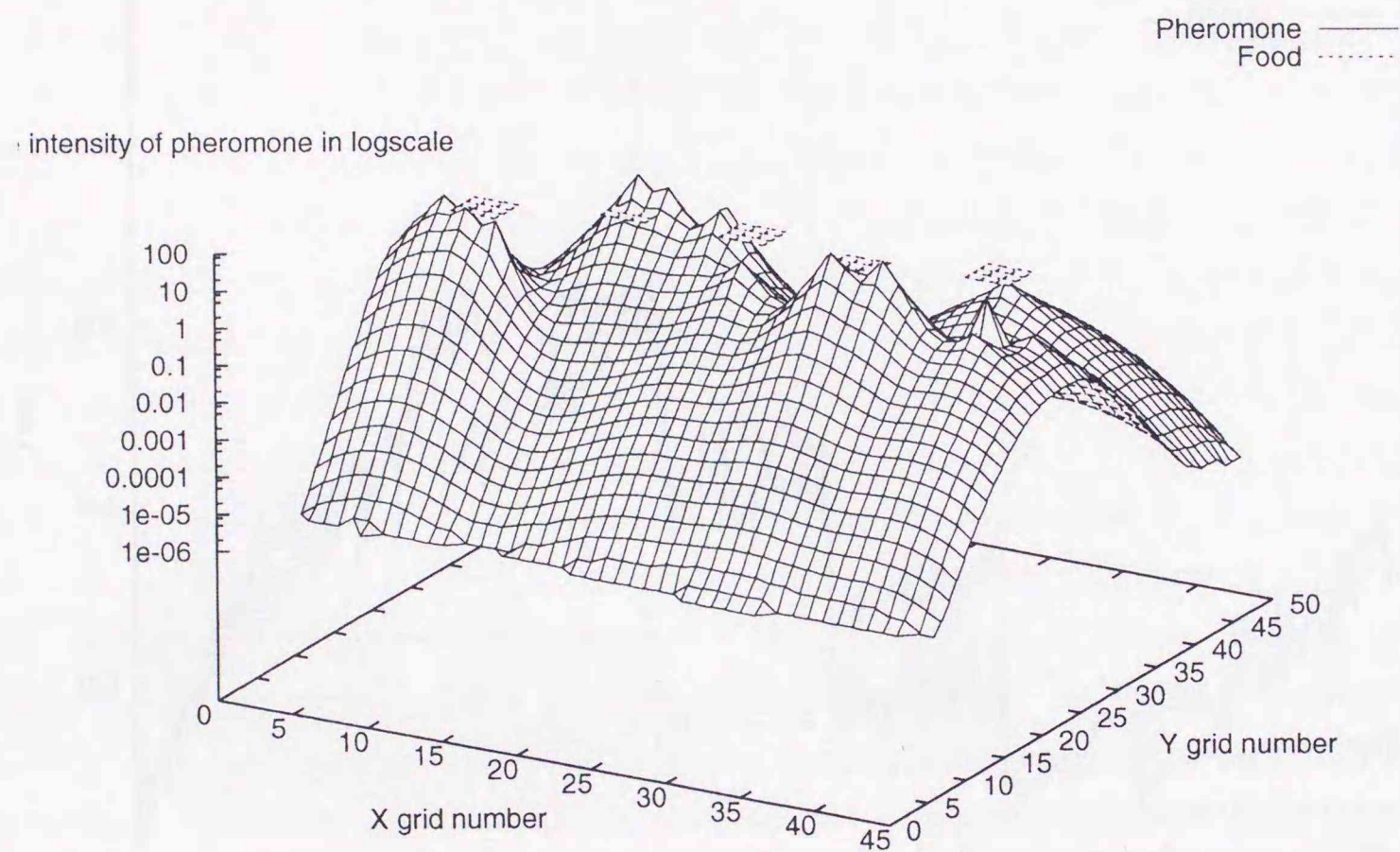
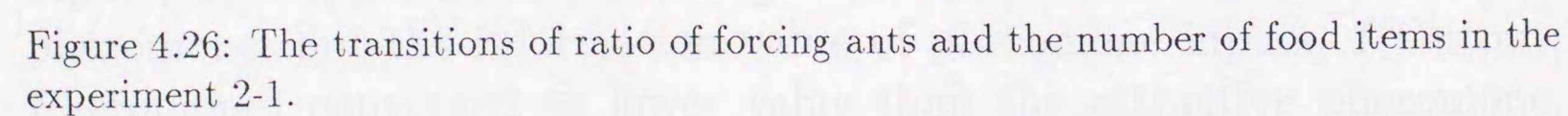


Figure 4.25: The contour line about the intensity of pheromone indicated by log scale in the experiment 2-1.

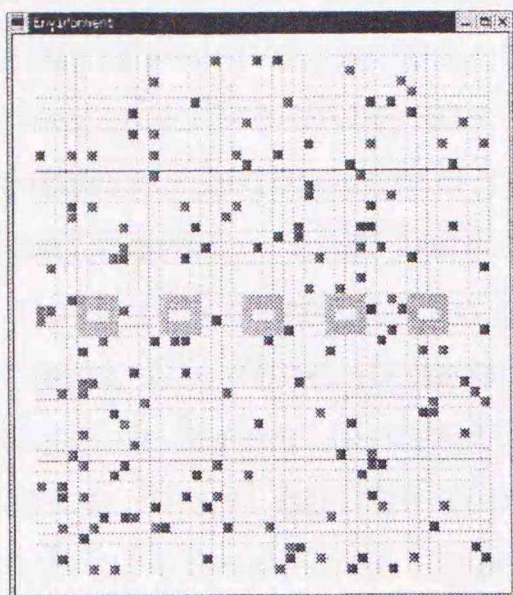


4.5.2.2 Result 2 about Repulsive Pheromone

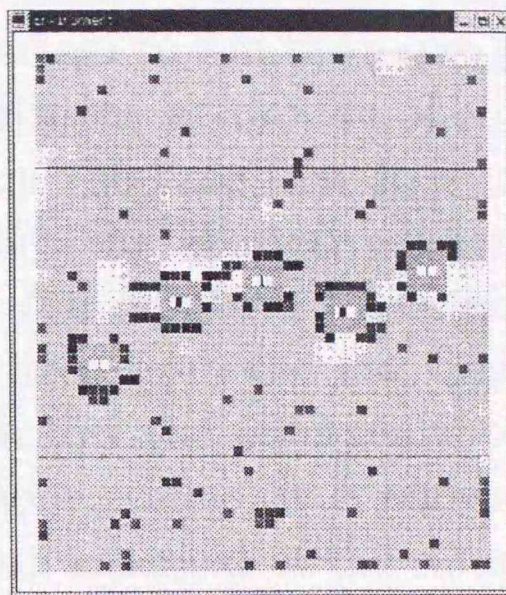
In this subsection, I show the another pheromone style communication obtained through evolutionary process. This pheromone style communication is named as repulsive pheromone communication because the repulsive pheromone works with the opposite effect compared with the previous attractive pheromone, namely, this pheromone seems to repulse ant agents. Fig. 4.27 shows the screen shots of ants war with agents in the final generation. In each time, agents seems to put the pheromone randomly, and it is difficult to judge whether this behavior is effective for agents to win this game or not, thus, it will be cleared in following discussion.

At first, it is confirmed whether the evolution properly occurred or didn't. The Fig. 4.28 and Fig. 4.29 represent the ratio to win against the final population and the transitions of $H(\text{Output})$, $I(\text{Output}, \text{Sensor})$, $I(\text{Output}, \text{Pheromone})$ respectively. Compared with the evolutionary situation of the attractive pheromone, the ratio to win against the final population is increased more slowly. Moreover, the entropy of outputs is decreased slowly than in the case of attractive pheromone. From this results, it is guessed that at first stage of evolution, ants tended to put the pheromone randomly, and the pheromone worked like noise in the world. Thus, it was difficult for ant agents to emerge intent of pheromone for communication. According to the progress of time, the intent of pheromone gradually emerged, and the entropy of outputs was decreased slowly. In the case of attractive pheromone, it was perhaps caused by the affection that the ant agent tended to put the pheromone near the food at first stage, but this repulsive pheromone was considered to be caused with no such affection. In addition, the information value of sensory inputs, $I(\text{Output}, \text{Sensor})$ converged to similar value of the attractive pheromone, but the information value of pheromone inputs, $I(\text{Output}, \text{Pheromone})$ converged to lower value than the attractive pheromone. The reason is considered as that it is easy for the attractive pheromone to have clear intent which represents that there is the food item near here, but the repulsive pheromone is used to represent that the food item doesn't exist near here. This intent of repulsive pheromone is maybe often unclear because the space in which no food item exist is larger, thus, the pheromone tends not to be uniformly. These facts are considered as the reason of causing slower evolution in the repulsive pheromone communication. Fig. 4.30 represents the transition of the average ratio to put the pheromone in this experiment. Throughout the evolutionary

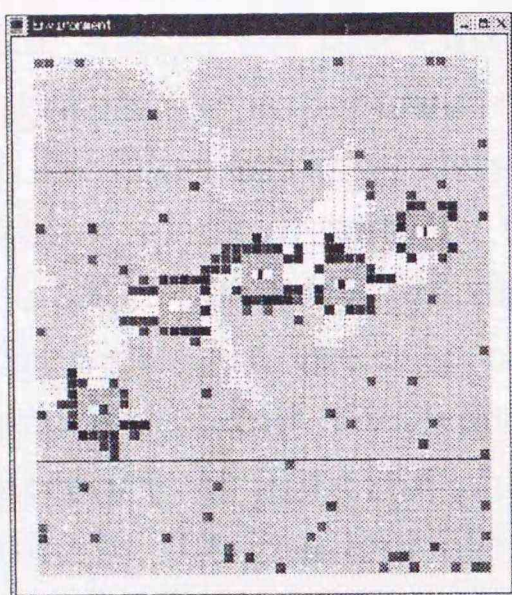
(a) Initial state



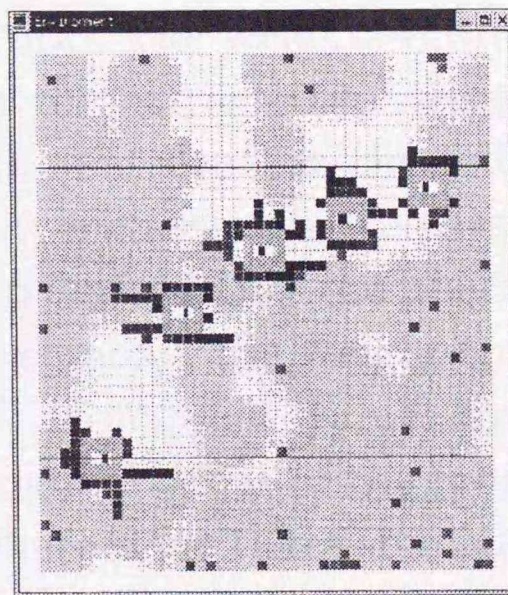
(b) $t=100$



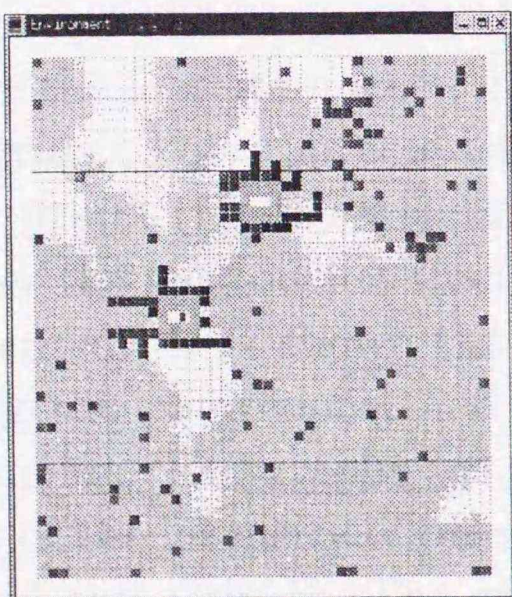
(c) $t=200$



(d) $t=300$



(e) $t=400$



(f) $t=500$

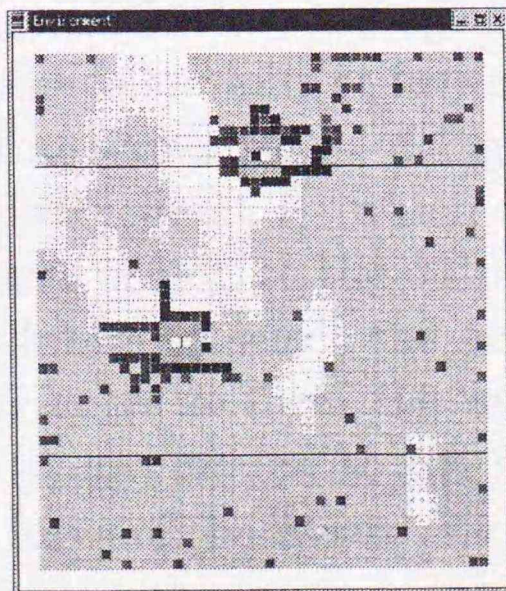


Figure 4.27: The screen shots of the ants war in experiment 2-2.

process, the ratio keeps high value.

Next, the effect of repulsive pheromone in the swarm condition is noted. Fig. 4.31 represents the ratio to win against the ant agents which were forbidden to put the pheromone in various numbers of agents. This graph shows this repulsive pheromone has an advantage to win the game according to the number of agents. Compared with the attractive pheromone, the repulsive pheromone is more effective in the case that it is used by single ant agent. This fact is based on that the attractive pheromone exerts once discovering the food item but the repulsive pheromone exerts to seek the food item on un-searched area from the first time. The advantage in larger number of agents indicates the emergence of communication based on the repulsive pheromone. Thus, it is showed that two kinds of pheromone style communication emerged of the same agent architecture and evolutionary strategy.

On the other hand, the graph (a) in Fig. 4.32 to (k) in Fig. 4.35 show the weight set of evolved neural network. The weights represented in the graph (a) is peculiar compared with the attractive pheromone in Fig. 4.18. The input i_1 detects whether the ant touches the food item at neighbor grids or not. In the previous experiment, the weights to the action 6 and 7, namely, forcing the food item and putting the pheromone, are large value. In the case of repulsive pheromone, the weight to action 6 is very large, but the weight to action 7 is negative value, thus, the ant scarcely ever puts the pheromone at neighbor grids of the food item. Accordingly, the gradient of the repulsive pheromone takes smaller to the area near food items. The graphs (b), (c), (d) and (e) are concerned with the sensory inputs detecting whether the fellow ant exists at neighbor grids or not. These graphs show the characteristic that ant agents tend to force the food item when in contact with the fellow ants at forward and right grids. This characteristic is observed in screen shots of Fig. 4.27. In the graph (f), the weights from the sensory input i_7 , which detects whether the ant touches the forcing fellow ants or not, to the outputs are represented. It is shown that this input inhibits to select the action of putting the pheromone, and this fact contributes to form the repulsive pheromone. The graphs (g), (h), (i), and (j) are related to the sensory inputs of the intensity of pheromone at the forward, backward, right, and left grid respectively. The graph (g) shows that the input i_7 inhibits to choose the action of moving forward, namely, other actions are probably chosen. The graph (h) indicates the same aptitude of the sensory input i_8 . The sensory input i_9 excites to select the action moving to right, and i_{10}

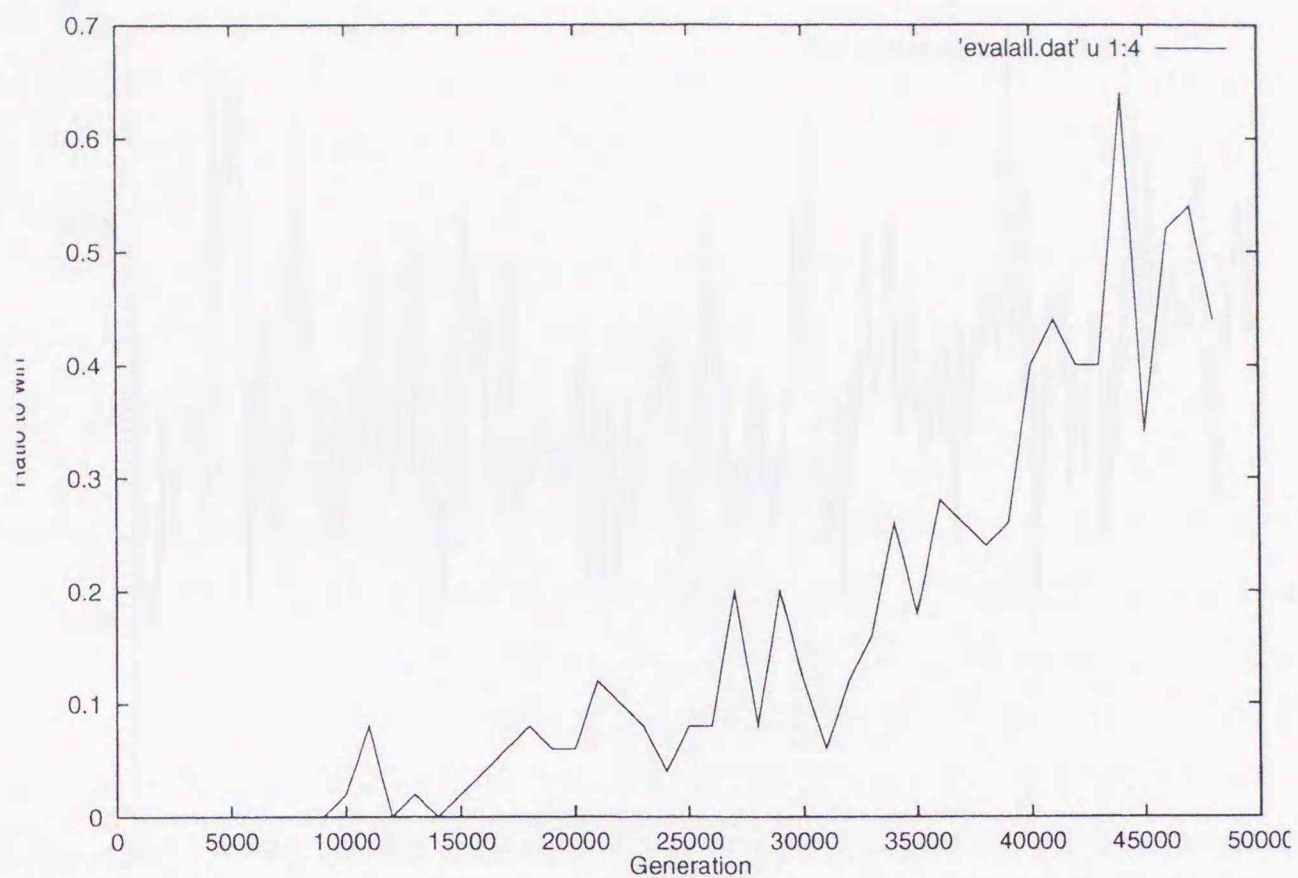


Figure 4.28: The ratio to win against the final population in the experiment 2-2.

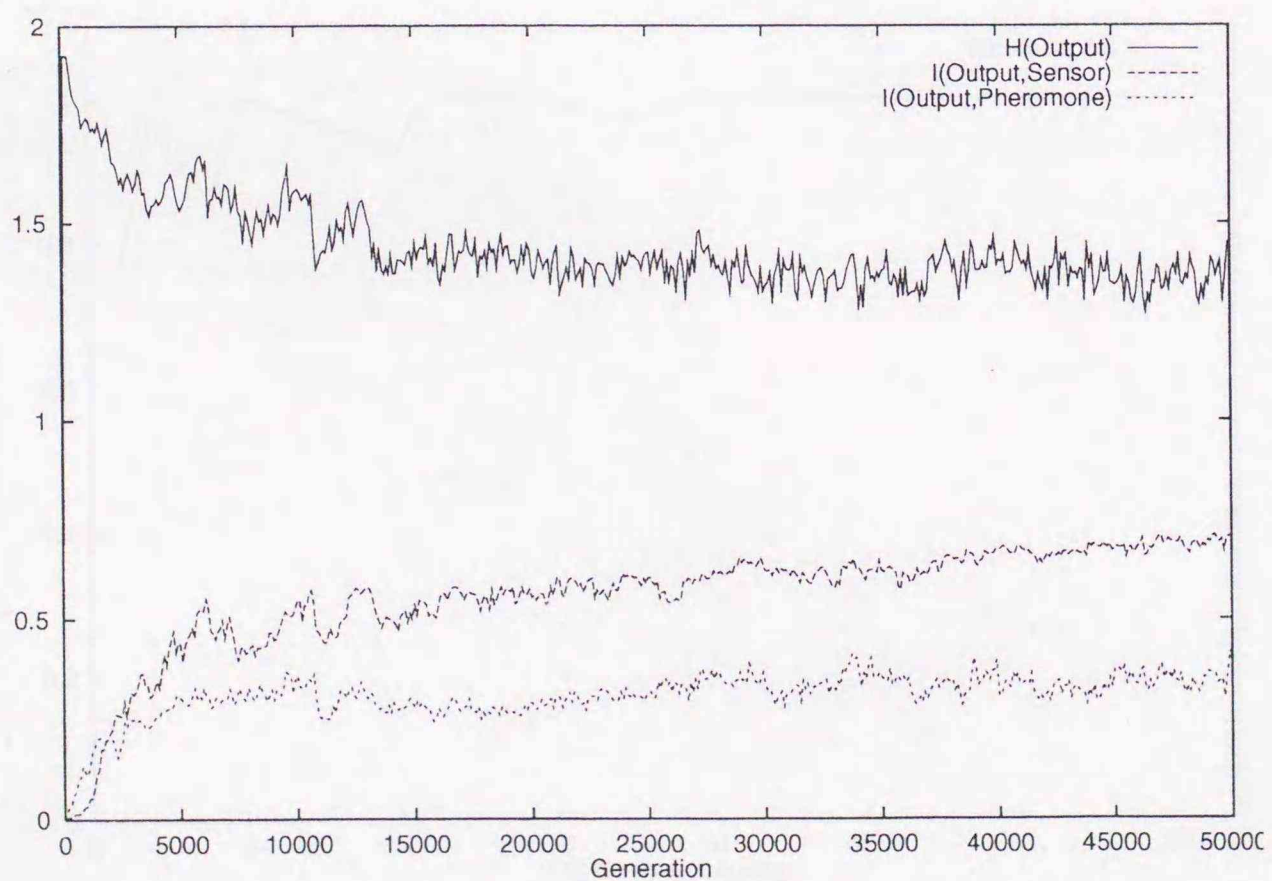


Figure 4.29: The transitions of $H(\text{Output})$, $I(\text{Output}, \text{Sensor})$, and $I(\text{Output}, \text{Pheromone})$ in experiment 2-2.

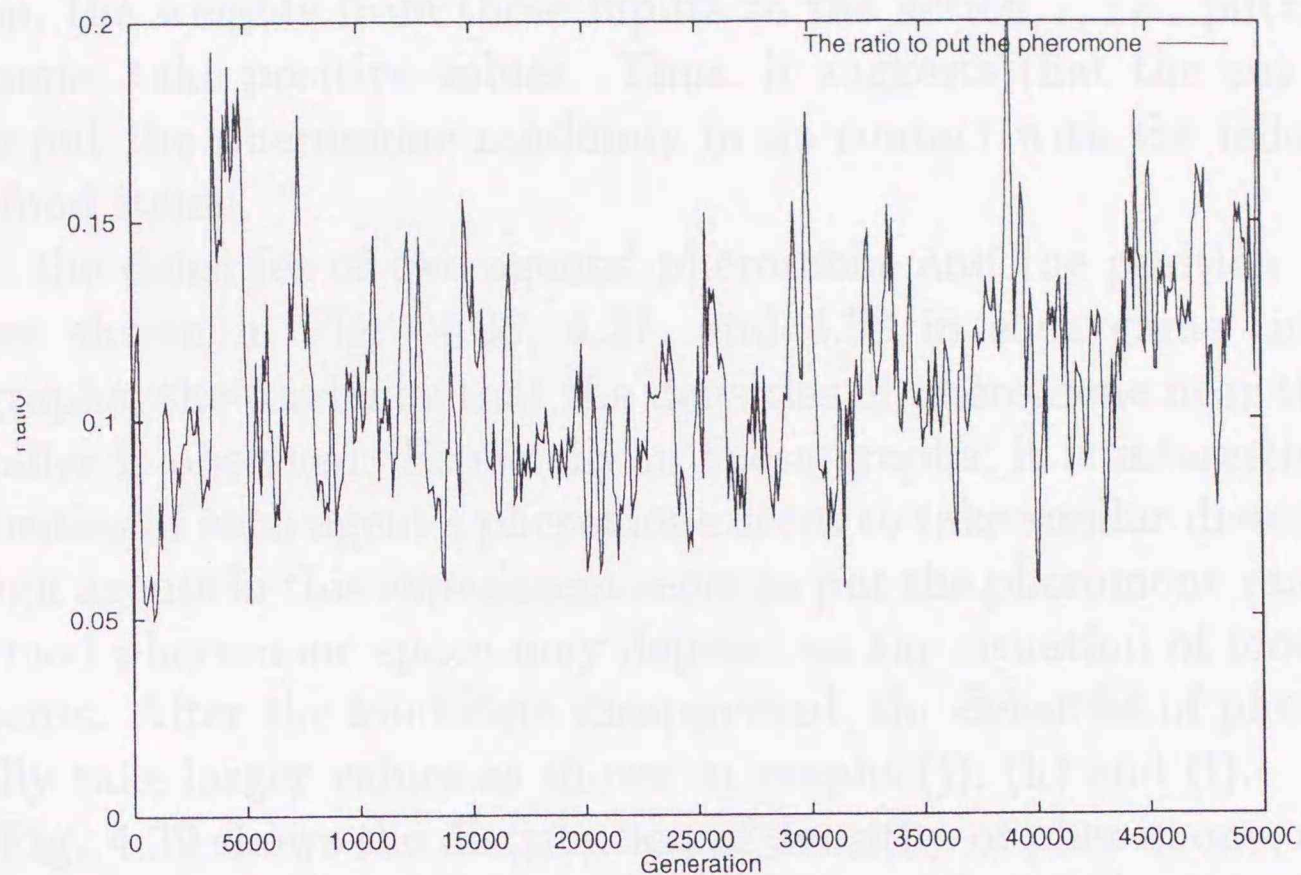


Figure 4.30: The transition of the ratio to put the pheromone in the experiment 2-2.

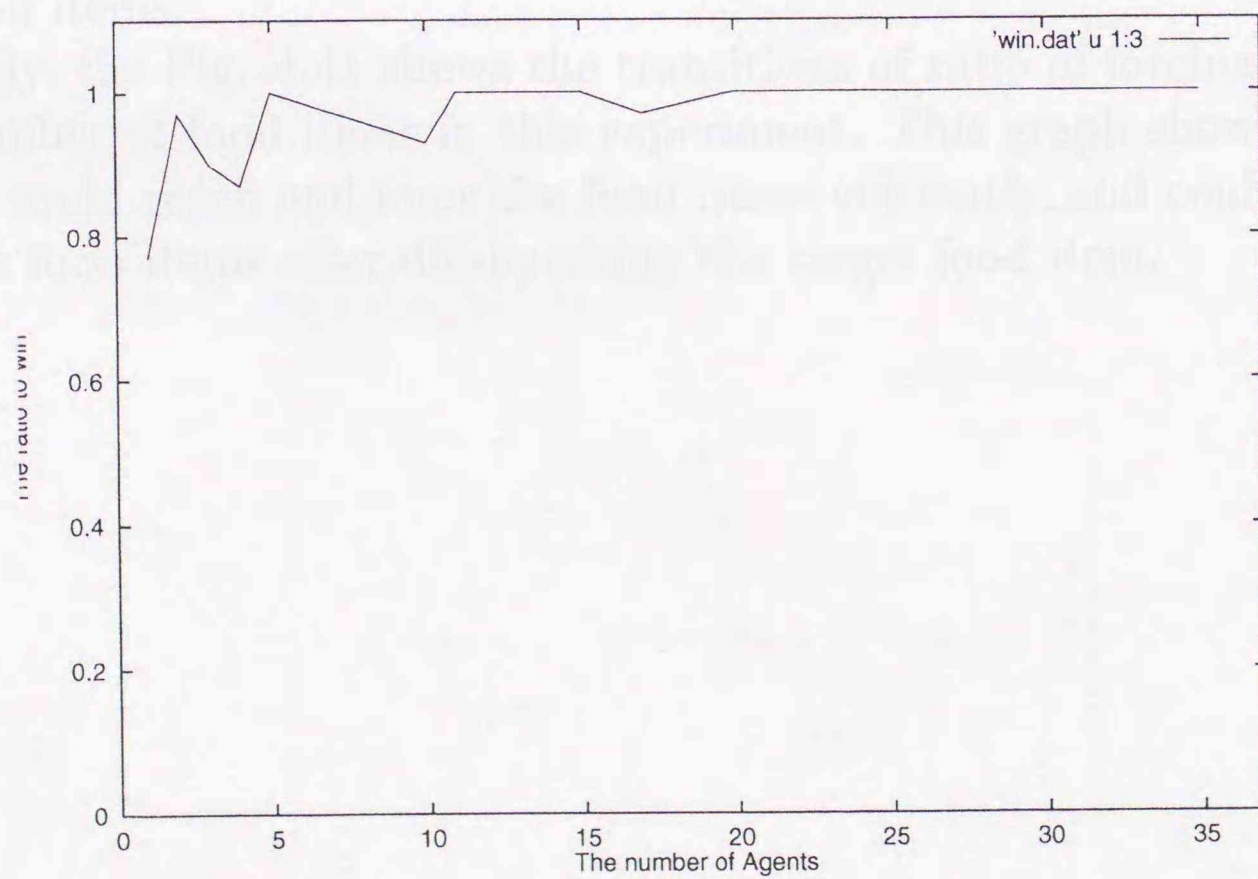


Figure 4.31: The ratio to win against the ant agents which were forbidden to put the pheromone in the experiment 2-2.

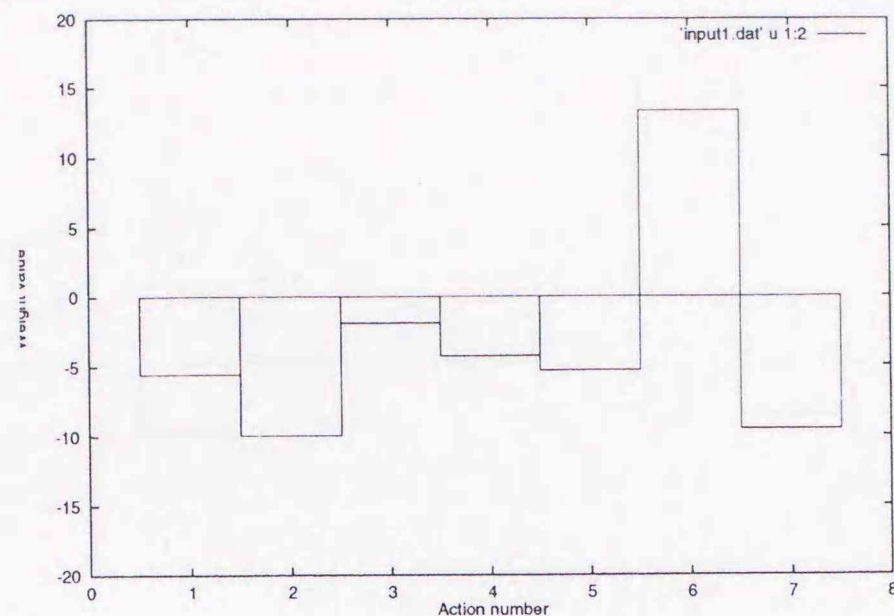
seems not to affect to select the actions concerned with move directions. Resultingly, these sensory inputs about the pheromone make ant agents repulse the strong pheromone according to the gradient of density. In addition, the weights from these inputs to the action 7, i.e., putting the pheromone, take positive values. Thus, it suggests that the ant agents tend to put the pheromone randomly in no contact with the fellow ants or the food items.

Next, the densities of two agents' pheromone and the position of food item are shown in Fig. 4.36, 4.37, and 4.38 in each game time. In these graphs, the tendency that the densities of pheromone near the food are smaller is observed. Especially in these graphs, it is interesting that the densities of each agent's pheromone seem to take similar distribution. Although agents in this experiment seem to put the pheromone randomly, the formed pheromone space may depend on the situation of food items and agents. After the food item disappeared, the densities of pheromone generally take larger values as shown in graphs (j), (k) and (l).

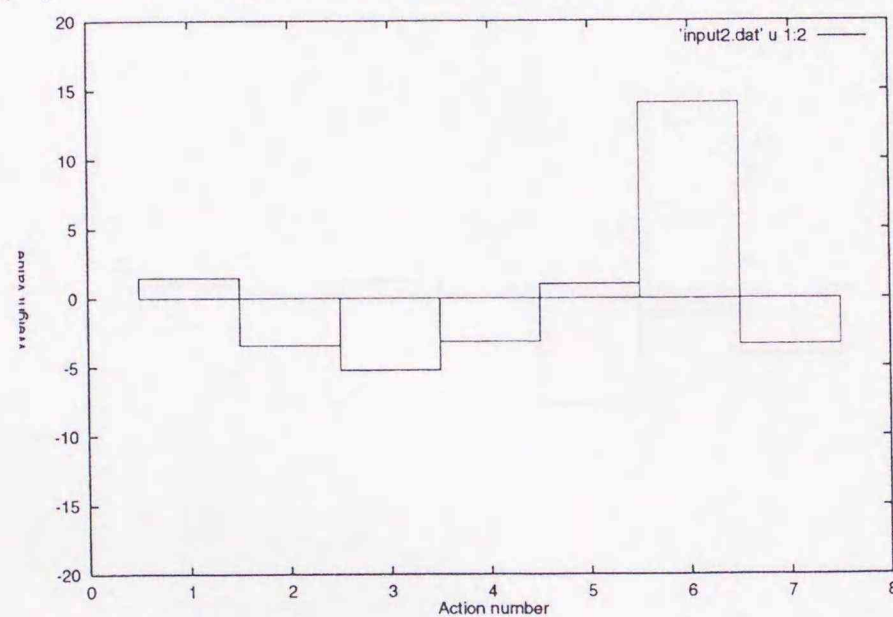
The Fig. 4.39 shows the distribution of densities of pheromone, and the Fig. 4.40 shows the same data with using log scale for density values as z axis. It is difficult to specify the characteristic of repulsive pheromone but it seems that the densities of pheromone take larger values apart from the food items.

Finally, the Fig. 4.41 shows the transitions of ratio of forcing ants and the number of food items in this experiment. This graph shows the ant agents could reach and force the food items efficiently, and could recover to next food items after disappearing the target food item.

(a) The weights from input i_1 to 7 outputs



(b) The weights from input i_2 to 7 outputs



(c) The weights from input i_3 to 7 outputs

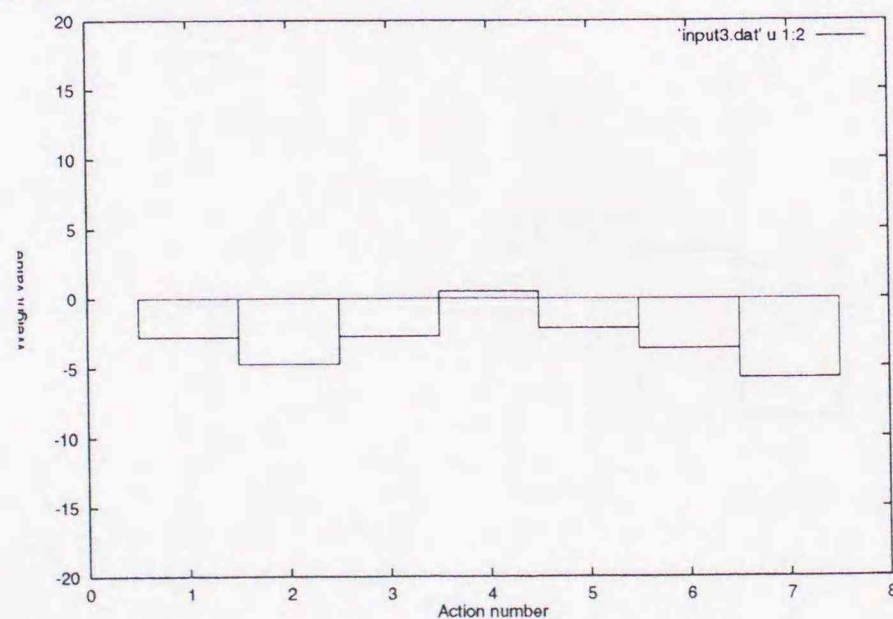
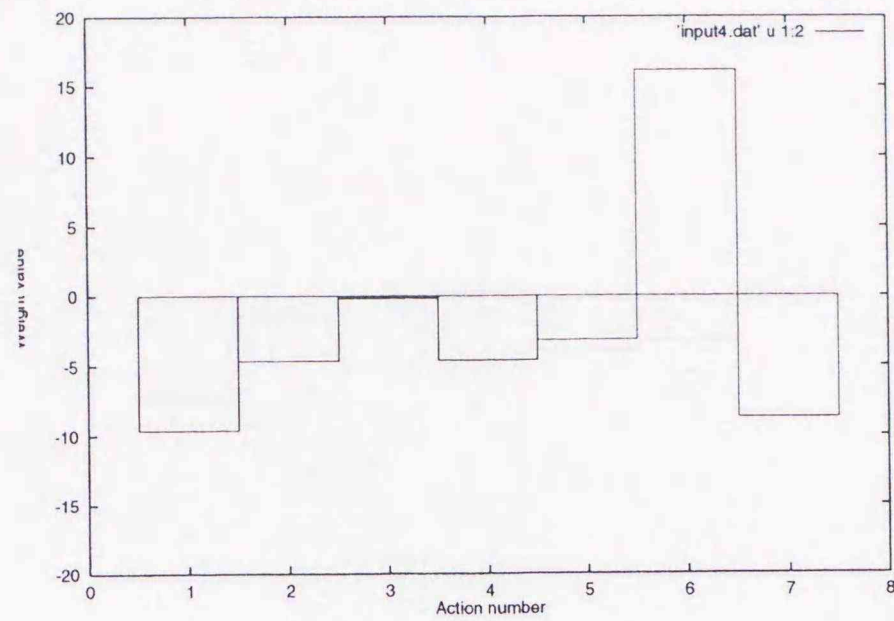
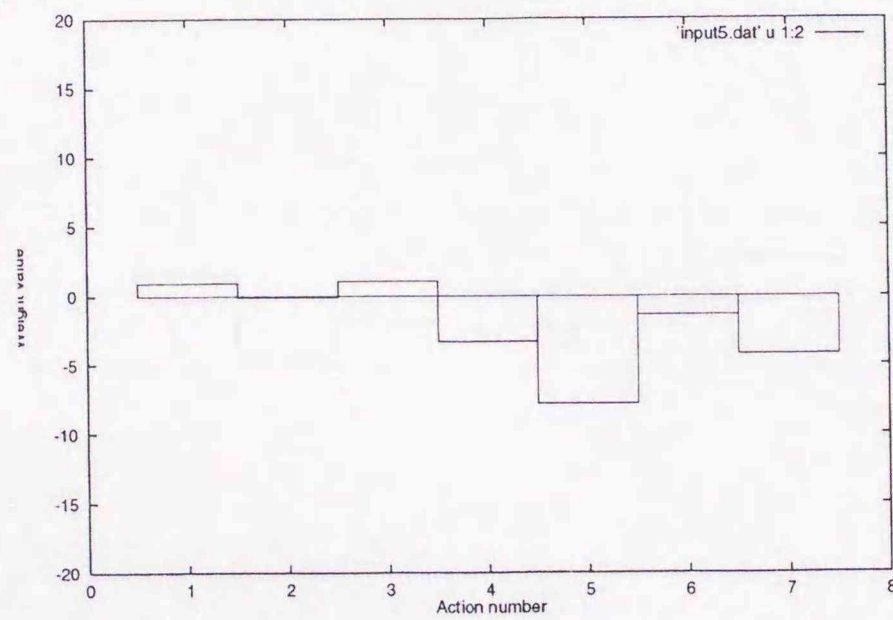


Figure 4.32: The weights from input $i_1 - i_3$ to outputs in the experiment 2-2 (A).

(d) The weights from input i_4 to 7 outputs



(e) The weights from input i_5 to 7 outputs



(f) The weights from input i_6 to 7 outputs

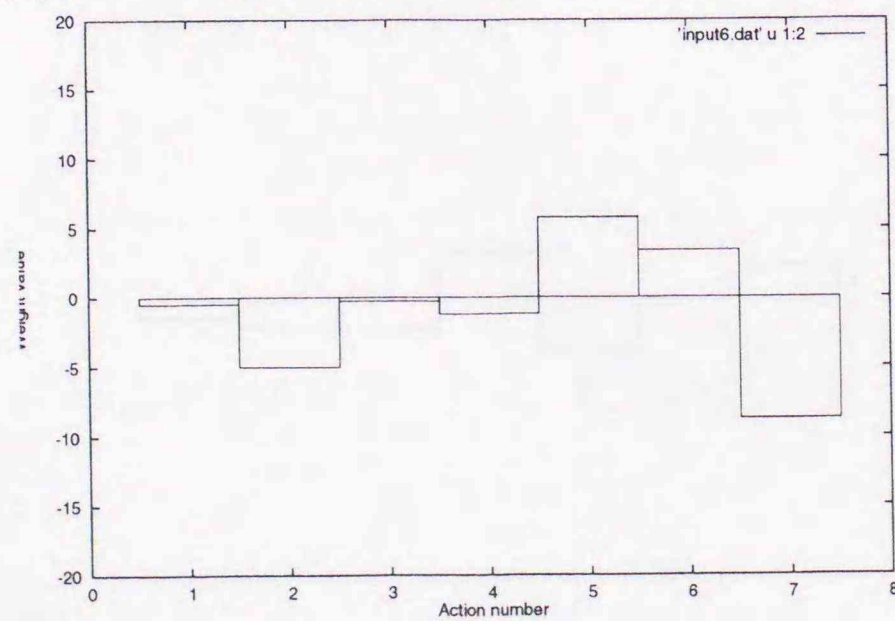
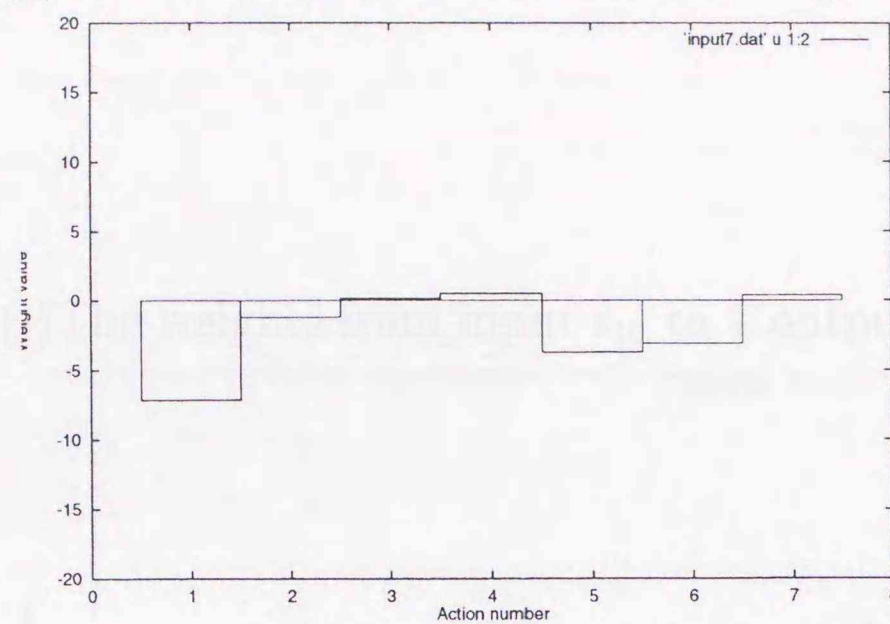
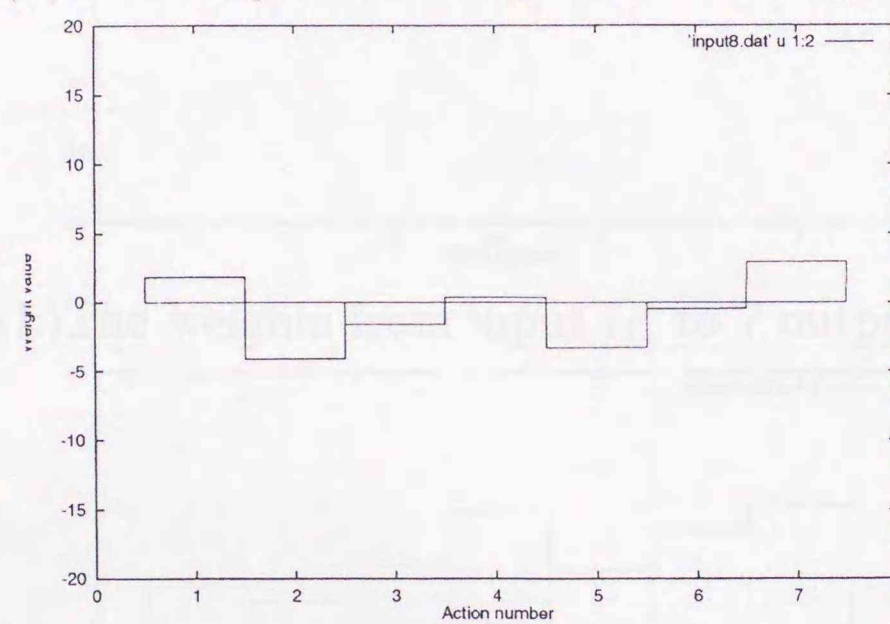


Figure 4.33: The weights from input $i_4 - i_6$ to outputs in the experiment 2-2 (B).

(g) The weights from input i_7 to 7 outputs



(h) The weights from input i_8 to 7 outputs



(i) The weights from input i_9 to 7 outputs

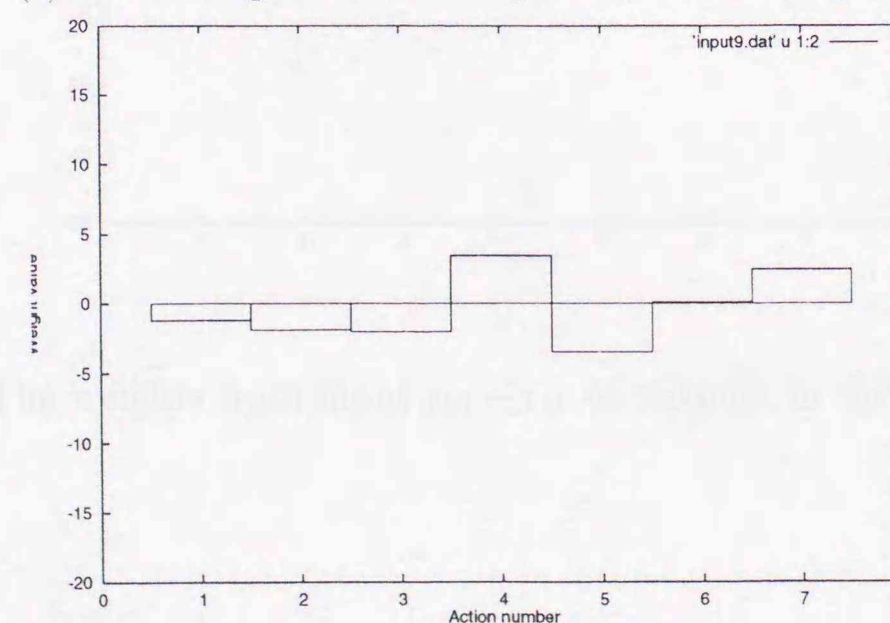
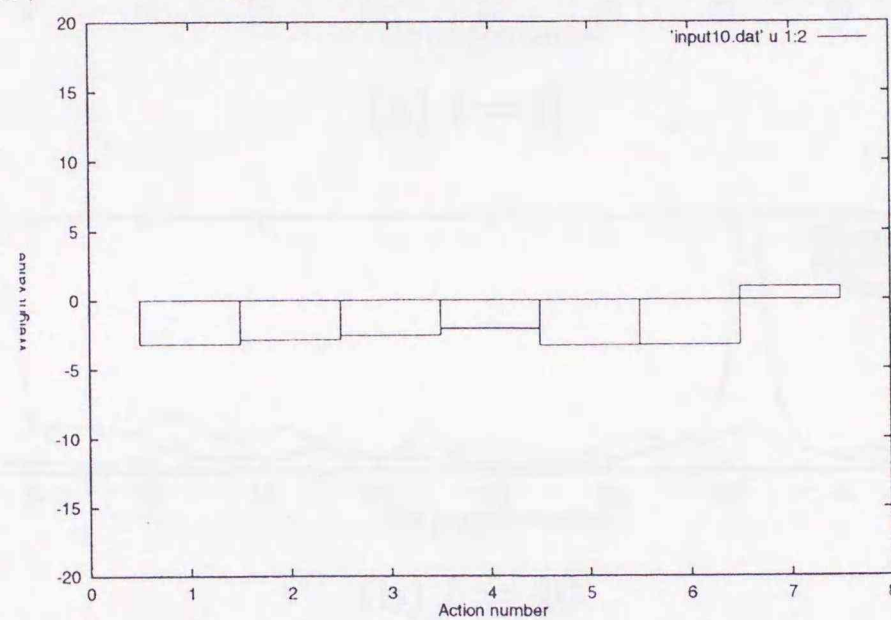


Figure 4.34: The weights from input $i_7 - i_9$ to outputs in the experiment 2-2 (C).

(j) The weights from input i_{10} to 7 outputs



(k) The weights from input i_{11} to 7 outputs

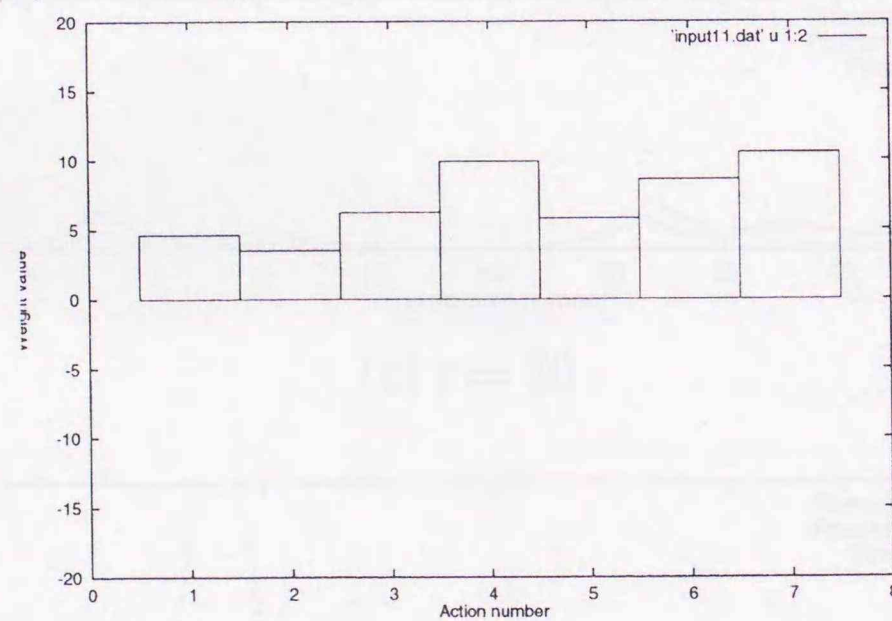
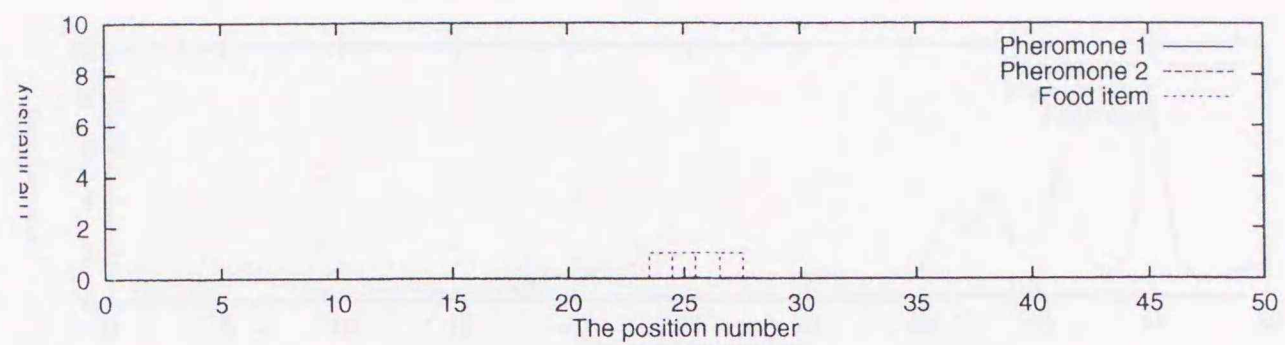
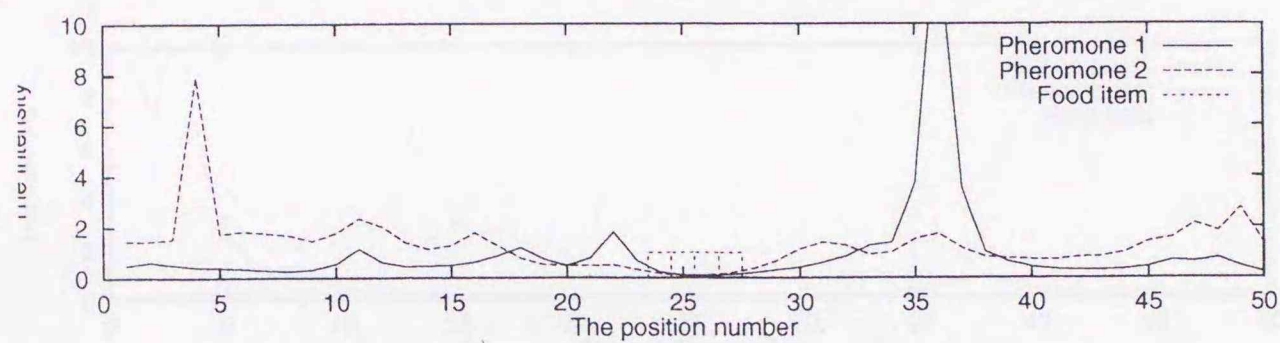


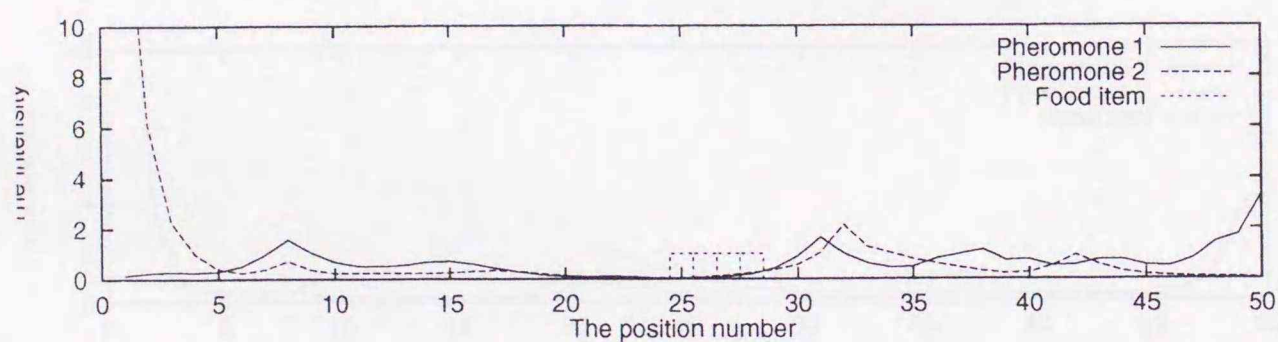
Figure 4.35: The weights from input $i_{10} - i_{11}$ to outputs in the experiment 2-2.



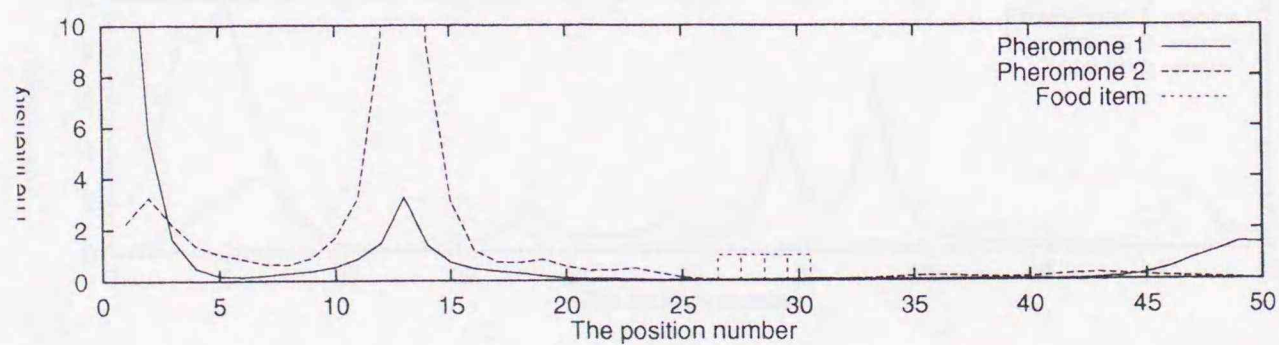
(a) $t = 0$



(b) $t = 40$

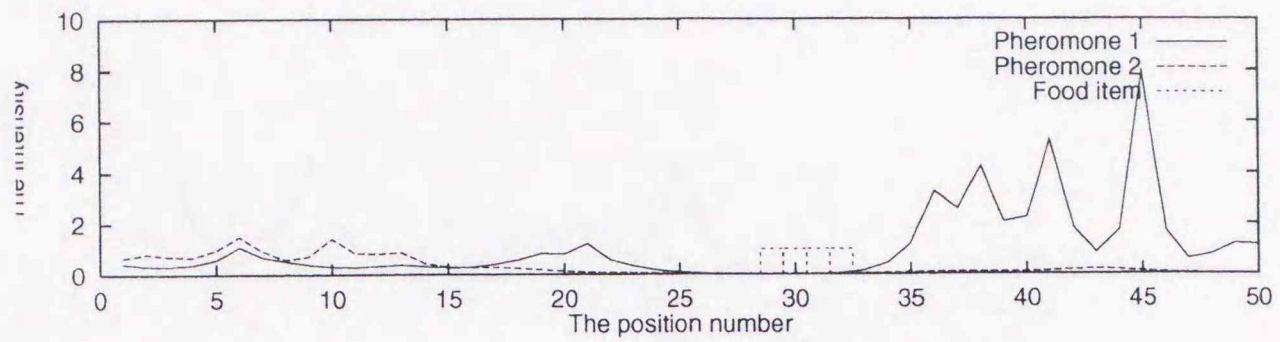


(c) $t = 80$

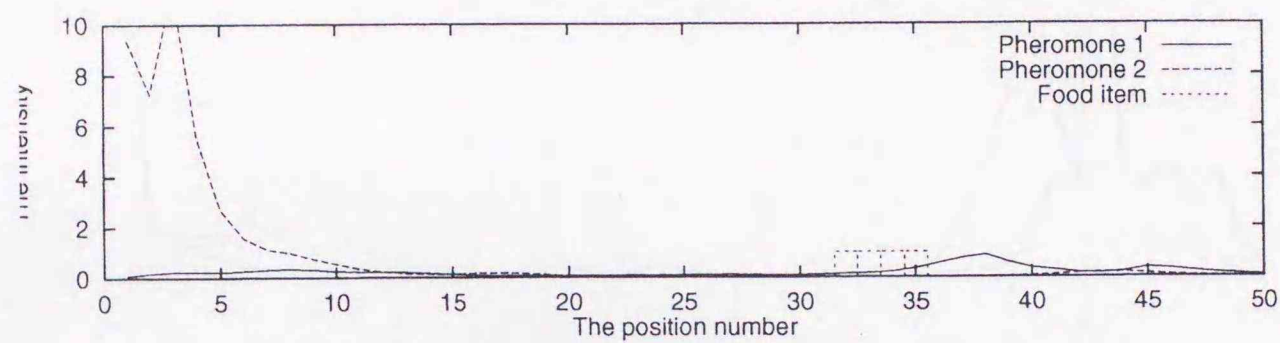


(d) $t = 120$

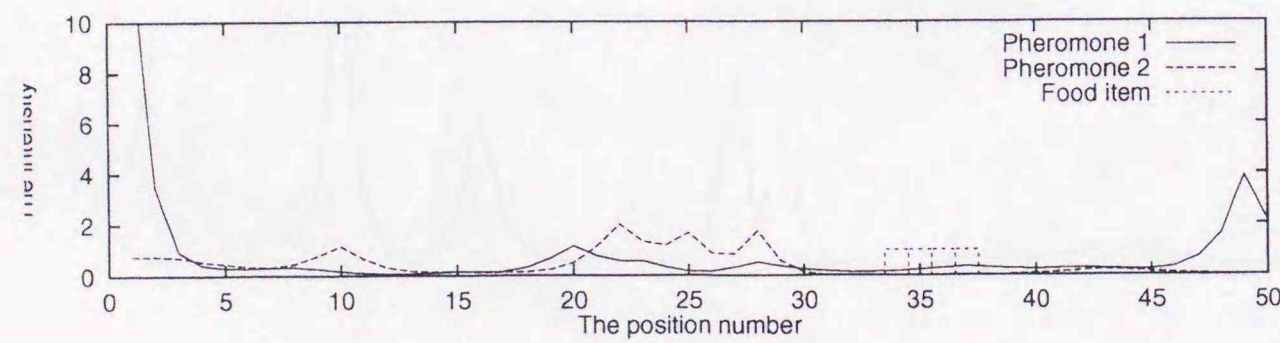
Figure 4.36: The transitions of intensity of pheromone in experiment 2-2 (A).



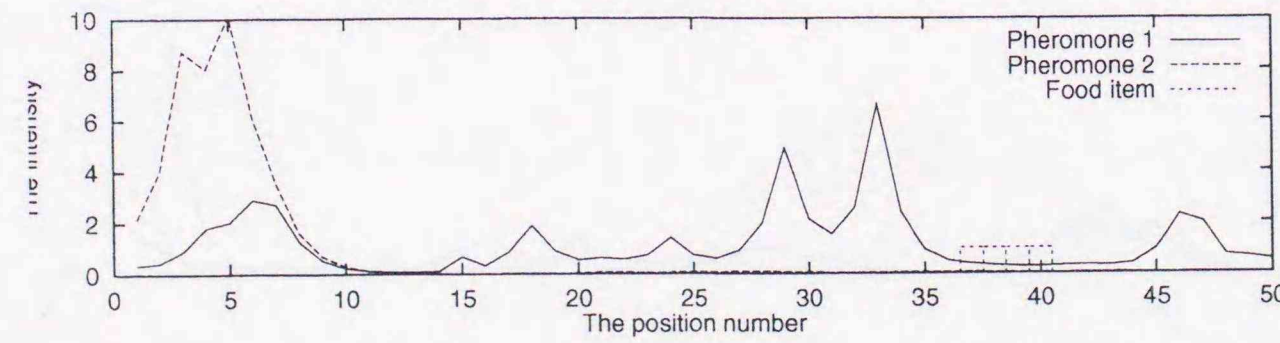
(e) $t = 160$



(f) $t = 200$

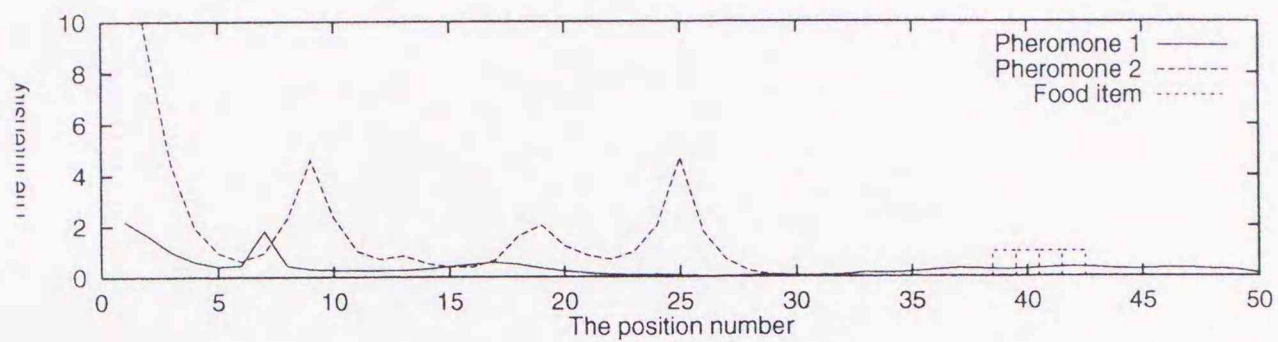


(g) $t = 240$

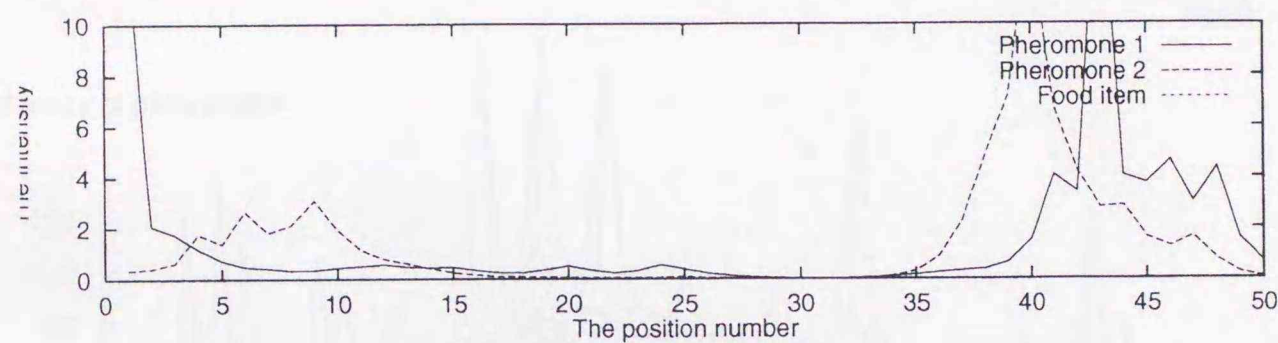


(h) $t = 280$

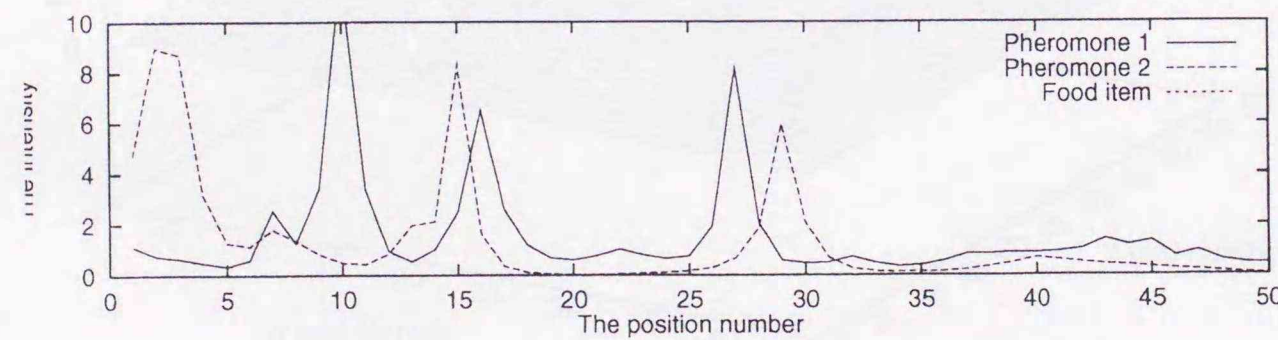
Figure 4.37: The transitions of intensity of pheromone in experiment 2-2 (B).



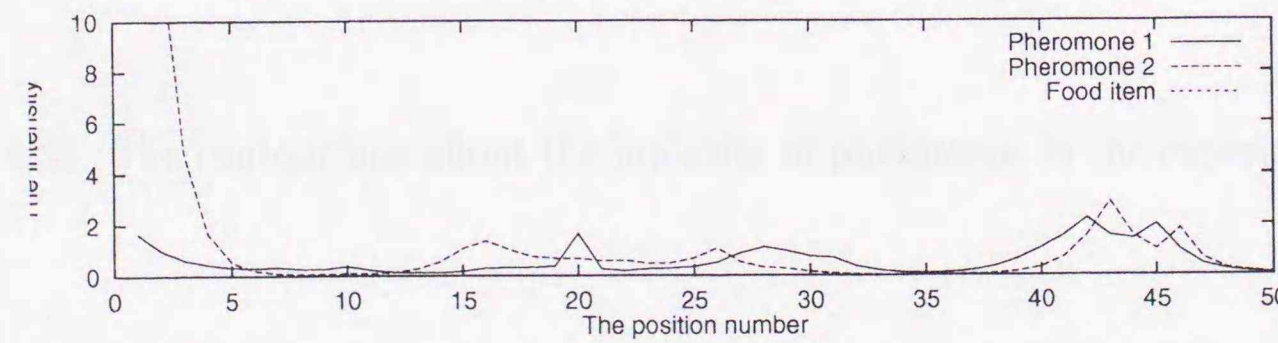
(i) $t = 320$



(j) $t = 360$



(k) $t = 400$



(l) $t = 440$

Figure 4.38: The transitions of intensity of pheromone in experiment 2-2 (C).

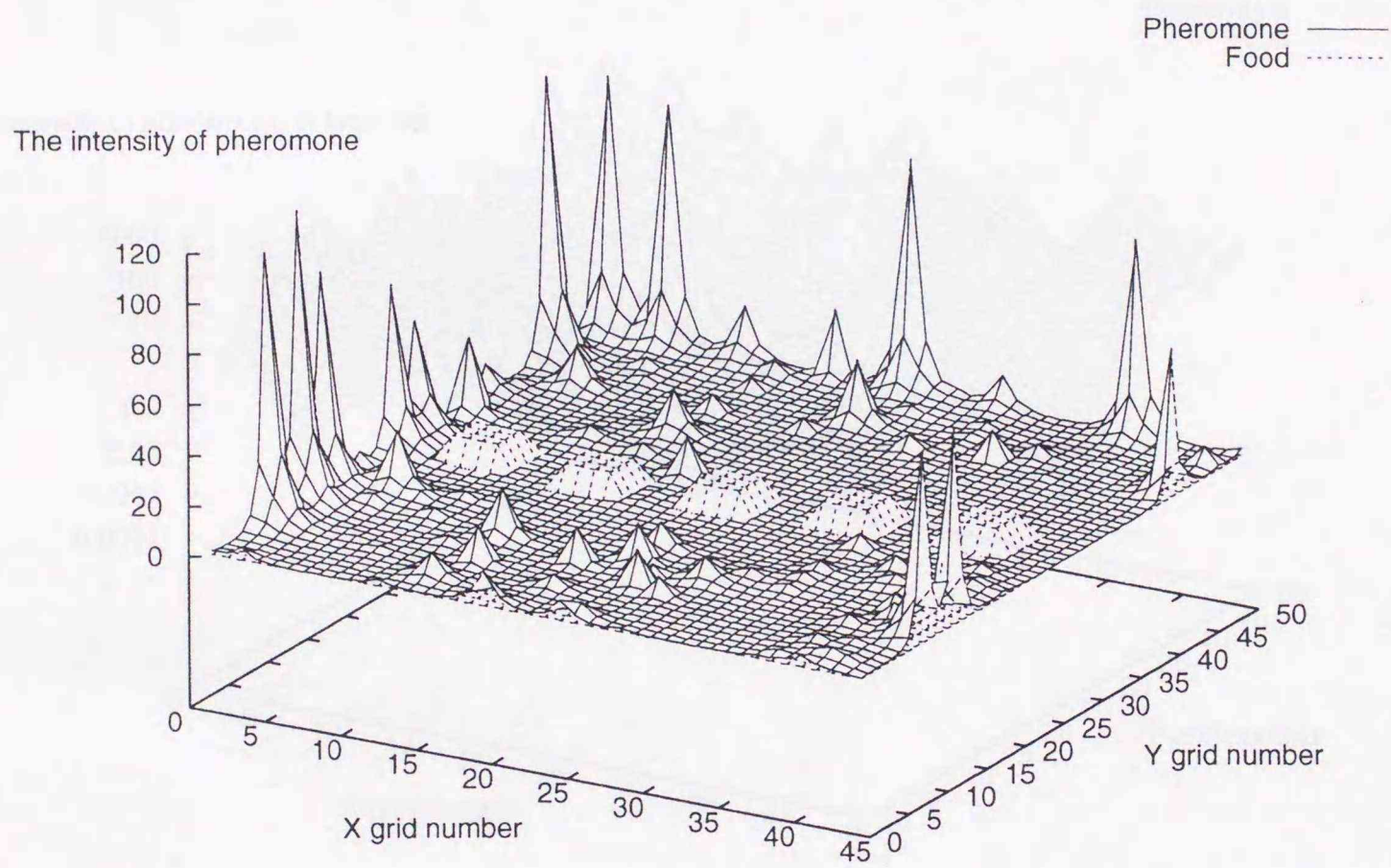


Figure 4.39: The contour line about the intensity of pheromone in the experiment 2-2.

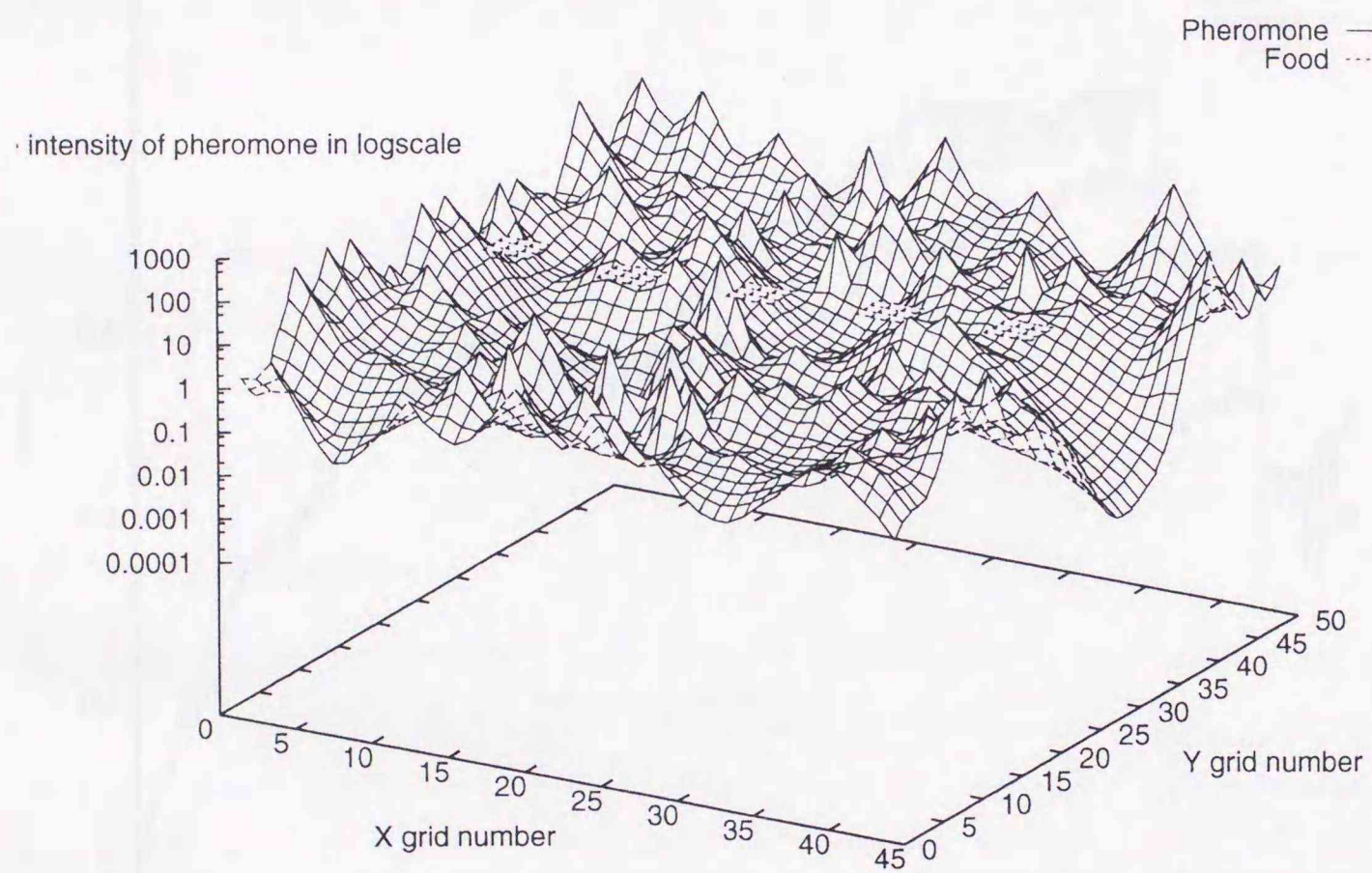


Figure 4.40: The contour line about the intensity of pheromone indicated by log scale in the experiment 2-2.

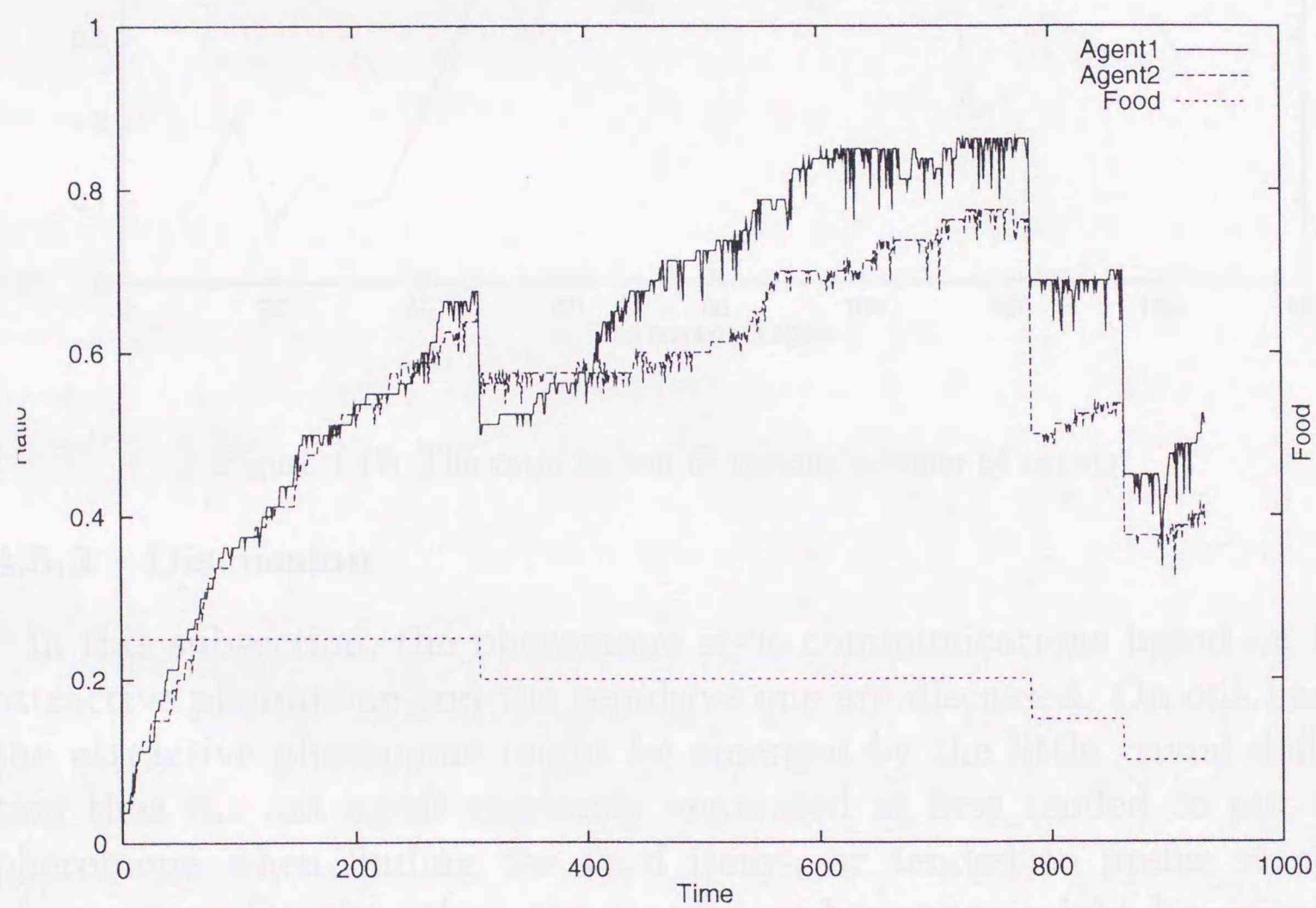


Figure 4.41: The transitions of ratio of forcing ants and the number of food items in the experiment 2-2.

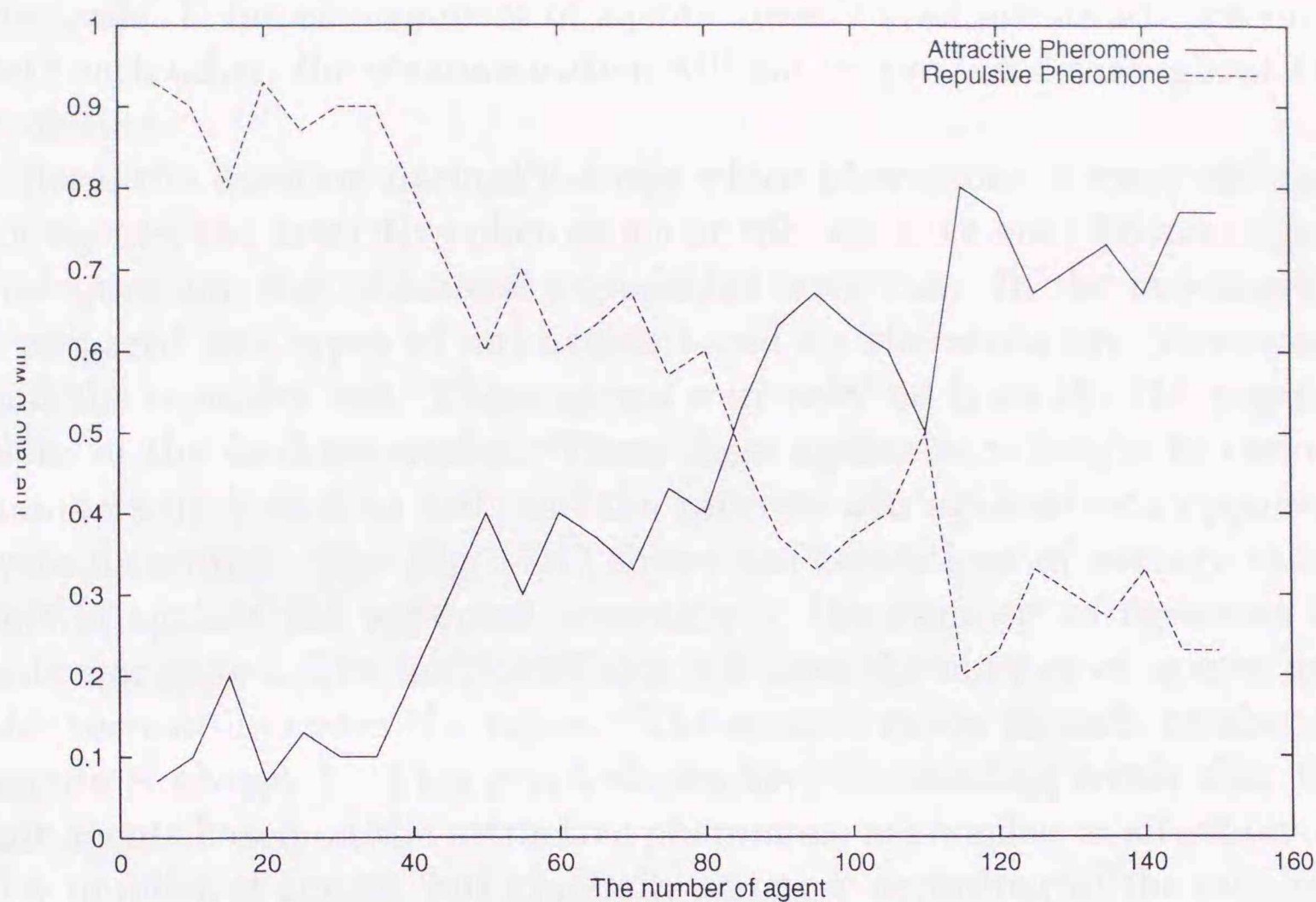


Figure 4.42: The ratio to win in various number of agents.

4.5.3 Discussion

In this subsection, the pheromone style communications based on the attractive pheromone and the repulsive one are discussed. On one hand, the attractive pheromone might be emerged by the little trivial deflection that the ant agent randomly generated at first tended to put the pheromone when finding the food items, or tended to prefer stronger pheromone. On the other, the repulsive pheromone might be emerged by that the ant agent generated at first stage tended to dislike stronger pheromone, or to broadcast the pheromone indiscriminately. Then, these deflections had been amplified by the selection pressure in the evolutionary process based on the results of ants wars. I can say that such deflection causes the emergence of communication, but it is difficult to answer what kind of deflection enables agents to emerge communication, namely, the reaction to putted pheromone, the situation to put the pheromone, or both. Moreover, it is very important fact that these amplifications of deflections were related to anneal the strategy of agents. Therefore, I consider that the source of emerged communication is the agents' environment in that the communication, which hasn't yet emerged at first stage of the evolution, enables swarms of agents to win wars for their

survivals. If the environment of agents doesn't need agents to communicate each other, the communication will not be produced throughout the evolution.

Here, the question naturally arises which pheromone is more effective for agents, the attractive pheromone or the repulsive one. To investigate this question, the additional experiment were run. In the experiment, I prepared two types of ant agents based on the attractive pheromone and the repulsive one. These agents were selected from the GA populations in the final generation. Then, these agents were fought in various numbers of from 5 to 150, and the ratio to win against each opponent were measured. The Fig. 4.42 shows the transitions of average ratios to win against the opponent according to the numbers of agents in 30 ants war games. The horizontal axis indicates the number of agents, and the vertical indicates the ratios. The sum of ratios in each number of agents is always 1. This graph shows very stimulating result that the ant agents based on the attractive pheromone are weaker in situations of few number of agents, but gradually stronger according to the numbers of agents, and strength of both agents are changed each other. Over the number of 120, the agents based on the attractive pheromone are admittedly stronger than the agents based on the repulsive pheromone. In addition, strength of both agents are almost balanced about the number of 80 agents, that is the default agent number used in the evolutionary scenarios. Although it has been shown in the previous subsections already that each pheromone is effective in large number of agents compared with no-pheromone agents, effectiveness of pheromones is surprisingly different with each other. In the case of the attractive pheromone, it is easily expected that the effectiveness of pheromone is increased according to the number of agents because the effect of attractive pheromone works once one agent discovers the food items, and second discovering ant, third one, forth one, etc., attracts other ants. The effect to attract is stronger according to the number of ants diffusing the pheromone just as emergent phenomenon. In another case of the repulsive pheromone, the pheromone is rather used as the earmark of already searched area than the information for interaction each other. This effect is useful in proper number of agents, but it would be anarchy in large number of agents because many ant agents diffuse the pheromone here and there, and effective gradient of densities cannot be formed. This result is very interesting to investigate the emergence level of swarm behaviors.

4.6 Conclusions

In this chapter, I defined the ants war, in which ant agents are evolved through the GA to survive, and made several computational experiments based on two types of agents, based on the continuous pheromone sensor, and the discrete pheromone sensor. In the continuous pheromone sensor, I confirmed three types of evolved agents, namely, the drifting ants without a pheromone, the ants drifting up and down with the primitive pheromone communication, and the marching ants with the pheromone.

In the experiments concerning with the discrete pheromone sensor, I confirmed the emergence of communication based on the attractive pheromone and the repulsive pheromone, which have opposite characteristics. Moreover, I discussed the emergence and effectiveness of pheromone style communication by using of several quantitative evaluable indicators.

Bibliography

- [1] W. C. Agosta, Chemical Communication - The Language of Pheromone -, W. H. Freeman and Company, 1992.
- [2] D. Cliff, and G. F. Miller, Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations, Proceedings of the Third European Conf. of Artificial Life, pp.200-218, 1995.
- [3] A. Colorni, M. Dorigo, and V. Maniezzo, Distributed Optimization by Ant Colonies, Proceedings of ECAL91- European Conference on Artificial Life, F.,Varela and P., Bourguine (Eds.), Elsevier Publishing, pp.134-142, 1991.
- [4] D. J. Montana, and L. D. Davis, Training Feedforward Networks Using Genetic Algorithms, Proceedings of the International Joint Conf. on Artificial Intelligence, pp.762-767, 1989.
- [5] M. Nakamura, and K. Kurumatani, Formation Mechanism of Pheromone Pattern and Control of Foraging Behavior in an Ant Colony Model, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems, C., G., Langton and K., Shimohara (Eds.), MIT Press, pp.67-74, 1997.
- [6] K. Suzuki, and A. Ohuchi, Reorganizations of Agents with Pheromone Style Communication in Multiple Monkey Banana Problem, Proceedings of Intelligent Autonomous Systems 5, Y., Kakazu, M., Wada, T., Sato (Eds.), Ohmsha, pp. 615-622, 1998.
- [7] G. C. Williams, The Pony Fish's Glow, Orion Publishing Group (1997)
- [8] H. Iba, The Method of evolutionary computation, Tokyo-daigaku-syuppankai, 1999. (in Japanese)
- [9] I. Matsuba, The information processing based on neural systems, Syoukoudou, 1993. (in Japanese)

- [10] T. Arita, Artificial Life, Kagaku-gijyutu-syuppansya, 2000. (in Japanese)
- [11] T. Hoshino, The evolutionary theory can't be known without computations, Kyoritsu-syuppan, 1998. (in Japanese)

Chapter 5

Summary

In this doctoral dissertation, multi-agent systems were focused as intelligent problem solving systems, in which autonomous agents behave based on local informations for achievement of given tasks. Especially, the interaction among agents plays very important role to emerge whole rational behavior by the base that the total behavior caused by individual agent behavior exceeds mere sum of components of systems. Therefore, it is easily recognized that the communication is one of best way for agents to interact each other. For designers of multi-agent systems, it is needed to construct reasonable communication which has enough ability to emerge total intelligence of agents systems, but it is naturally difficult issue alike many ones about learning, adaptiveness, planning, and so forth. In chapter 1, the recent works and issues about multi-agent systems were surveyed, and two types of communication forms, the direct communication and the indirect communication were described. For the effective communication of multi-agent systems, I focused on real ant colony system in which many ants excellently keep their colony. In ants colony, each ant plays various roles with organizing ability based on interactions. The pheromone, which is chemical substances, is used for ants as one of communication mean. The pheromone has enough flexibility and adaptability for real ants colony. Thus, this pheromone communication was introduced to multi-agent systems, and the formalization of pheromone style communication for multi-agent systems was described. On the other hand, several works based on pheromone style communication were surveyed according to two categories, simulation models of pheromone style communications, and ant colony optimization algorithms. Especially, the ant colony algorithm originally proposed by Colnani et al. were circumstantially introduced as one of most famous agent systems based on pheromone style communications. Resultingly, chapter 1 concerned with multi-agent systems based on pheromone style com-

munication, and described about the design concept of pheromone style communication.

In chapter 2 and 3, the applications of multi-agent systems based on the pheromone style communication were focused on to solve some kinds of combinatorial optimization problems. In the chapter 2, the ant algorithm by Coloni et al. was picked up for solving traveling salesman problems. The ant algorithm has many ant agents, each of which independently searches the problem space by moving from a city to city according to both of local information caused by the characteristics of the problems and global information caused by the pheromone. The pheromone are used to exchange global information about the solved problem. However, ant agents of the ant algorithm are based on same behavioral strategy, and it is not expected to emerge total inconceivable ability exceeding mere sum of agent abilities. Therefore, the concept of multi ant colonies were introduced to emerge the swarm intelligence using two types of pheromone effects, i.e., positive pheromone effects and negative pheromone effects. Agents belonging to each colony independently behave like the original ant algorithm but the colony level interactions based on pheromone effects cause to control whole behaviors of ant agents on each colony. The behaviors of colonies are affected each other, and the effectiveness of colony level interactions are worked. The computer experiments for several traveling salesman problems were made, and the effectiveness of proposed multiple ant colonies algorithm was revealed.

Chapter 3 concerned with problem solving based on coordinated behavior of agents. The distributed search algorithm was proposed to solve vehicle routing problems which is known as harder than traveling salesman problems. In the algorithm, there are several vehicle agents, and each agent independently searches own vehicle route. In search process, the competition situations, in which each agent wants to use same cities each other, often occurs in order to construct more preferable route. Thus, to resolve such competition situations, pheromone style communication were introduced as imprints which represent degrees of importance of cities for agents. Each vehicle agent could construct coordinated relations between other agents based on pheromone style communication. Throughout computer experiments, the effectiveness of pheromone style communication in distributed search algorithm was confirmed.

Chapter 4 was related to the evolutionary emergence of pheromone style communication rather than the application of pheromone style communication. In the investigation of pheromone style communication, ants

war was proposed. In this game, there were ant agents constructed by the neural networks, and each agent team on ants war gathers more food items than the opponent in order to survive in the evolutionary mechanisms based on the genetic algorithm. In addition, two types of pheromone sensors, i.e., the continuous pheromone sensor and the discrete pheromone sensor, were adopted as the pheromone sensor of agents. In the experimental cases of the continuous pheromone sensor agent, three types of agents emerged throughout evolutionary process, namely, the drifting ants without a pheromone, the ants drifting up and down with the primitive pheromone style communication, and the marching ants with the pheromone. In the cases of the discrete pheromone sensor, two types of agents based on the attractive pheromone and the repulsive one were observed in the computer experiments. These pheromone effects had antagonistic characteristic, but the both of effects of these pheromones were advantage against no pheromone ant agents. Especially, the emergence of pheromone style communication were investigated in various points. The measured information value about pheromone clearly showed the emergence of intent of pheromone. The experiments in changing the number of agents showed the level of emergent behavior based on evolved pheromone communications. As results of chapter 4, the possibility, effectiveness and emergence of pheromone style communication were cleared based on ants war game.

Finally, I conclude this dissertation with following items.

- The pheromone style communication is very effective way for multi agents to interact with each other in order to perform given tasks.
- The pheromone style communication is useful to construct coordinated relationship among agents.
- It is possible for designers to design effective pheromone style communication according to the objectives of designed agent systems.
- The evolutionary design of pheromone style communication and agent strategy is enough possible.
- The emergence level of agents based on pheromone style communication is high.

Acknowledgments

The author especially expresses to thank Professor Azuma Ohuchi of Hokkaido University for instruction and guidance from the inception to the completion of the present research. He thanks to Professor Yukinori Kakazu, Professor Eiichi Miyamoto and Professor Mitsuo Wada for beneficial comments to compile this study. He also wishes to thank Associate Professor Keiji Suzuki and Assistant Masahito Yamamoto in Hokkaido University. Many pieces of affectionate advice and discussion given by both were very worthful to improve this research. He would like to thank all students of the office, Harmonious Systems Engineering, Complex Systems Engineering. The conversations between them often intenerated his mind, and sometimes encouraged him. At last, he appriciates all what his family had done for supporting his life.

101 *Continued Agents of Science*

