Abstract

Colonies of social insects are capable of solving complex tasks that far exceed the abilities of each individual insect. The colonies do not use any supervisor or blueprint to organize their work, instead the solutions emerge from the interactions between the insects and their environment. Two of the tasks that social insects perform are clustering of corpses and sorting of brood. In this thesis we describe our work with creating swarms of simple agents that perform similar tasks. Previous research within this field has hand-coded the behavior of the individual agents and then seen if a swarm of the agents is capable of solving the desired task. We take a different approach and evolve the individual behavior by evaluating the patterns that are formed by the swarm. We evolve swarms to solve three different types of tasks: Clustering, patch sorting, and annular sorting. The first two tasks involves the grouping of identical objects, and the grouping of different types of objects into separate groups. Annular sorting involves the creation of a targe-like structure that contains circular bands. This task has not been solved successfully in the past, and we are able to create a dense and well separated structure. A shorter presentation of the work can be found in [19].

Contents

Chapter 1

Introduction

Colonies of social insects are capable of solving complex tasks despite the simplicity of the individual insects. They are able to create complex architectural structures, optimize their foraging, and sort their brood into different patterns. In none of these tasks is there any supervisor or blueprint that the insects use to organize their work. The insects are all relatively simple with only access to information about their immediate surroundings. Through the interactions between the individual insects, and the insects and their environment, the colonies are able to solve tasks that far exceed the abilities of each individual insect.

Social insects have in the later years inspired problem solving in computer science. This approach has been termed 'swarm intelligence', and it emphasizes distributedness, direct or indirect interactions among simple agents, flexibility, and robustness [4]. By replicating the way that social insects solve complex tasks, better techniques may be discovered in computer science. The replication of insect behavior may also lead to a better understanding of nature by discovering how the individual behavior is related to the behavior of the colony as a whole.

One field of research within swarm intelligence is the investigation of social insects' ability to cluster and sort objects. Some species of ant deposit their corpses in piles outside the nest, while some species sort their brood into different patterns. The majority of research within this field has dealt with the replication of insect behavior by investigating their behavior, and then forming hypotheses about the rules governing this behavior. These rules are then hand-coded and tested experimentally in computer simulations to see if the desired patterns emerge at the global level of a swarm of agents. If the desired patterns do not emerge, the behavioral rules are tweaked and re-tested until the patterns emerge.

Our approach to the replication of insects' ability to cluster and sort objects differs from the traditional approach described above. We create the individual behavior of the insects by artificial evolution. Artificial evolution has been successful with solving problems in a variety of areas, but it has never been used to evolve the individual behavior of a swarm of insects capable of clustering or sorting objects. The individual behavior is evolved by allowing a swarm of agents to interact with a collection of objects for a specified amount of time. Then the behavior is evaluated in terms of how close the emergent pattern of objects is to the desired pattern.

The tasks for which we evolve swarms of agents are clustering, patch sorting, and annular sorting. In the clustering task the evolved swarm of agents must group a collection of identical objects into a single cluster. The patch sorting tasks consist of grouping different types of objects into separate clusters, such that each cluster contains objects of only one type. The tasks of clustering and patch sorting have been solved by several researchers in the past. The task of annular sorting has, on the other hand, not previously been solved successfully[1]. This task involves the creation of a single target-like structure that contains circular bands of objects, with each band containing objects of only one type.

In our work we use simple agents that only have information about their immediate surroundings, and that can pick up and deposit objects in addition to move around. Through artificial evolution we evolve feedforward neural network controllers that control the pick up, deposit, and movement actions of the agents. By doing this we are able to create swarms of homogeneous agents that are capable of clustering, patch sorting, and annular sorting.

In the next sections of this chapter we introduce the biological background for the work described in this thesis. We briefly introduce different aspects of social insects, before giving a more detailed description of some forms of pattern formation in social insects. After this we introduce the concepts of self-organization and stigmergy. These concepts describe the emergence of patterns at the global level of a system from interactions between individuals at the local level. Having introduced the biological background and important concepts, we move on to giving a formal problem definition of the tasks to be accomplished in our work. After this we describe our motivations before giving an overview of the structure of the remainder of this thesis.

1.1 Social Insects

Social insect colonies consist of hundreds or thousands of individuals. Each of these individuals seem to follow their own agenda, yet the colony appears to be highly organized. The swarm raids of army ants may for example contain up to 200000 workers and be up to 15 meters wide at the front[2]. These ants can sweep over an area that is $1500m^2$ in a single day. The ants in these raids do not have any knowledge about the raid as a whole, but rather they respond to stimuli in their immediate surroundings. Other insects use different strategies to forage. Some ants and honey bees are able to dynamically choose food sources that are both of the best quality and closest to the colony. Ants create trails covered with pheromones that show the way to the richest and closest discovered food source, and honey bees recruit other bees to good food sources by dancing for the other bees in the colony when they return with pollen and nectar. In both of these examples the colony as a whole is able to efficiently select the best food source based on the individual actions of the insects in the colony.

One of the most astonishing achievements of insects is their creation of nests with highly complex architectures. Tropical wasps build nests comprised of a series of horizontal combs protected by an external envelope, and that are connected to each other by a peripheral or central entrance hole. However, the perhaps most impressive

[1]Wilson et al. [33] have managed to create an annular-like structure, but this is inferior to our solutions both in terms of compactness and separation between the bands of objects.

[2]Raids by the ant *Eciton burchelli*.

structures built by insects are termite mounds. Termites of the *Macrotermes* species create large mounds that contain fungus gardens that the termites harvest, and the mounds are air-conditioned to keep the inside temperature at an optimum and to provide fresh air to the inhabitants.

What is common for the tasks accomplished by social insects is that the individual behavior of the insects is simple and uncomplicated compared to the global tasks that are solved by the colonies as a whole. Each individual that is part of a group performing a particular task does only have knowledge of its immediate surroundings, and does not know anything about the task that is to be solved at the global level. A termite that is building a pillar does, for example, not know anything about the architecture of the termite mound, it is only responding to local stimuli. The insects are thus capable of performing complex global tasks with each individual operating at a local level by responding to stimuli.

In the next sections four examples of global pattern formation by social insects are presented. The first section describes how an ant species forms a wall surrounding its brood. Then a description is given of how honey bees create a pattern of concentric regions of brood, pollen and honey on the wax combs of their hive. The last two sections show that ants are capable of clustering and sorting objects. In all of the examples the individual insects respond to local stimuli with no knowledge of the global pattern to be formed, yet swarms of the insects are able to create complex patterns at the global level.

1.1.1 Wall Building by Ants

The ant *Leptothorax albipennis* creates nests in narrow crevices in rocks. These ants are very small, and the cracks which they colonize may be only 2 or 3 cm wide and long with the gap being only 1 or 2 mm wide. Colonies of *Leptothorax albipennis* contain a single queen, workers, and a number of brood in different developmental stages. The structure of the nest formed by the ants is a C-shaped wall of 'bricks' that surround a cluster of brood and workers. A sketch of a *Leptothorax albipennis* nest is shown in Figure 1.1. The wall has only a single entrance, and the cluster is usually located opposite this entrance. Between the cluster of workers and brood there is a corridor of open space. When a colony of *Leptothorax albipennis* ants migrate to a new crevice, they first cluster the brood. Then the queen and the workers form a tight cluster over and around the brood. Only after this cluster is formed do the ants begin to form the surrounding wall which makes up the nest [9].

In their studies of the nest construction by *Leptothorax albipennis* ants Franks and Deneubourg [14] have found that the building process has a number of distinct stages. At the beginning there are a number of external workers that depart from the cluster of workers and collect a grain of building material. After they have picked up the grain they walk back to the cluster of brood and workers, and deposit the grain at a distance of one to two body lengths from the cluster. Ants that carry stones tend to release their stone after making direct contact either with a cluster of other workers, or other stones that have been deposited previously. As the building process advances the use of other stones as a landmark for depositing building material seems to increase in importance. Later in the construction process the ants appear to use their carried stone as a battering ram that they push into existing structures of stones,

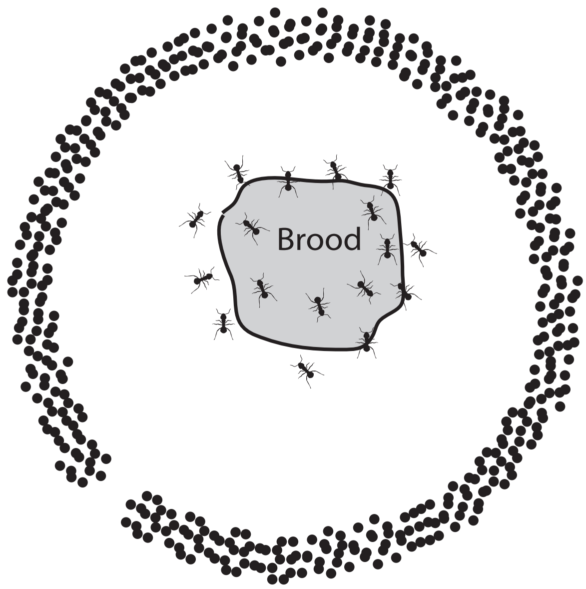

Figure 1.1: Sketch of a *Leptothorax albipennis* nest.

and only release the stone when they face a certain amount of resistance. This causes the walls of the nest to be tightly packed. The external workers that fetch material from the outside of the nest rarely pick up stones that have been deposited inside the nest. However, the workers that remain in the nest frequently pick up objects in the nest and push these outwards into the emerging wall.

After observing the creation of the nest structure, Franks and Deneubourg [14] investigated the behavior of external and internal workers more closely. The external workers go out into the surroundings of the nest to fetch building material. When they return to the nest they walk around randomly until they encounter other nest mates or stones that have been deposited inside the nest. The ants are more likely to put down their stone when encountering other stones than when meeting nest mates. At the beginning of the building process the stones may be encountered fairly randomly throughout the nest, but as the construction progresses other stones are mostly encountered along the walls of the nest. The internal workers pick up stones inside the nest and most of the time push them outwards away from the central cluster of brood and other workers. They bulldoze the stones into other stones, and this causes a dense packing of the stones in the walls of the nest. The position of the stones inside the nest determine how likely they are to be picked up and moved by the internal workers. Franks and Deneubourg [14] found that stones that were most exposed to ant traffic were the most likely to be moved by the workers. Since stones that are buried in the wall are rarely encountered by workers compared to stones that are inside the nest, this means that the workers will mostly move stones that are not already part of the wall. In other words, the stones that are not in the correct position are the most likely to be moved by the workers.

Investigation of the behavior of the workers showed that they appear to use the central cluster of brood and workers as a mechanical template for the formation of the nest walls. The first building workers that make contact with the cluster turn around and move away for a short distance before depositing their building material. The *Leptothorax albipennis* ants can in this way create a nest that is of the correct

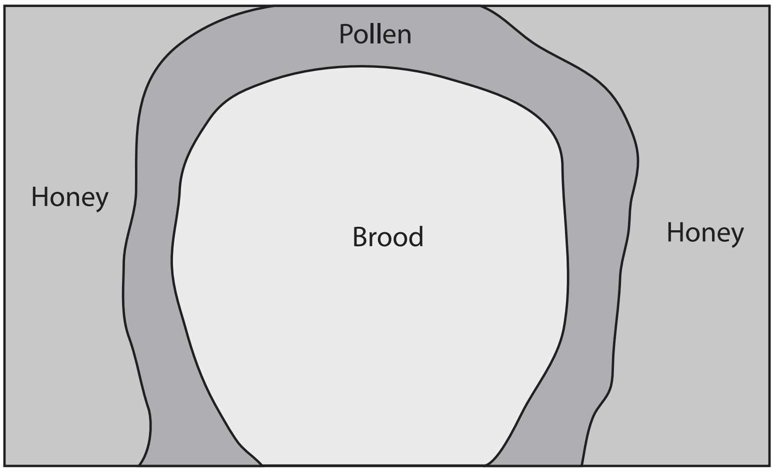

Figure 1.2: The wax combs of honey bee colonies contain a characteristic pattern of brood, pollen, and honey.

size with respect to the colony. The size of the cluster of brood and workers will vary according to the size of the colony, and the building workers can thus use this cluster to determine where the walls should be placed in order for the nest to be of the proper size for the colony. The building of walls by *Leptothorax albipennis* ants can be summarized to be the effect of simple behaviors of external and internal workers that do not communicate directly, and that use the formed cluster as a template for the formation of the nest walls.

1.1.2 Honey Bee Combs

A typical honey bee colony consists of female worker bees, male drones, and a single egg-laying queen. In addition to these adult bees there is immature brood in various developmental stages. The colony also stores honey and pollen in a series of parallel wax combs. These wax combs contain a characteristic pattern of brood, honey and pollen. A sketch of this pattern is given in Figure 1.2. The different items are positioned in three distinct concentric regions on the combs. In the center there is a brood area, which is surrounded by a band of pollen, with a large peripheral region of honey surrounding the area containing the other two items. This characteristic pattern of brood, pollen, and honey continue to exist despite the constant actions of the bees. The queen lays eggs, and other bees deposit pollen and honey in the cells, as well as remove pollen and honey from the cells. In spite of these continuous actions and seasonal variations in the availability of pollen and honey, the concentric pattern on the combs persists.

Camazine [8] and Camazine et al. [10] have investigated the pattern formation on the wax combs of honey bee colonies. They theorize that the pattern emerges as a dynamic process involving local interactions. From this standpoint they investigate the lower-level components of the pattern formation process. A single cell on the wax comb may contain brood, pollen, or honey, and they investigate egg-laying, pollen and honey deposition, and pollen and honey consumption to create a hypothesis of how the global pattern emerges. Their approach is thus to investigate the lower level components of the system to gain an understanding of how a pattern is formed at the global level. Camazine [8] investigated the behavior of a honey bee colony and formed a hypothesis about the formation of the pattern based on these observations. He then tested the validity of the assumptions made by performing a Monte Carlo simulation which incorporated the elements of the hypothesis. In the first hypothesis he only considered the deposition of pollen and honey in addition to the egg-laying by the queen. The resulting Monte Carlo simulation did not manage to recreate the pattern observed in the bee colonies, and so he went back to observe the colonies. In the second hypothesis he also included the consumption of pollen and honey in the model. The resulting Monte Carlo simulation successfully recreated the pattern that was observed in the honey bee colonies. The different rates of consumption and deposition of pollen and honey in addition to the egg-laying rate in the simulation is listed below.

1. The maximum egg-laying rate is 1 egg/min, 24h/day.
2. Honey and pollen are deposited in cells 12h/day.
3. The ratio of honey removal to input is 0.6.
4. The ratio of pollen removal to input is 0.96.
5. The ratio of pollen input to honey input is 0.2.

Based on the observed processes in the honey bee colonies and the successful replication of the concentric pattern in the Monte Carlo simulation, Camazine et al. [9] made the following observations concerning the processes that contributes to the formation of the pattern. The central brood area is formed by two different processes. The first is the egg-laying by the queen. The queen will lay eggs in the vicinity of other eggs, and this induces a clustering of the eggs. Pollen and honey are also deposited in the central brood area, but pollen and honey located near brood are preferentially consumed. This leads to pollen and honey being consumed at a higher rate near brood than pollen and honey located on the periphery of the comb. The central brood area is thus formed because it is continually freed of pollen and honey, while at the same time being filled with new eggs.

The segregation of honey and pollen on the periphery is the result of rates of consumption of pollen and honey, as well as the rates on which they are collected. Pollen is consumed at the periphery at nearly the same rate that it is deposited. This means that when pollen is deposited here, it is soon removed by consumption. At the same time honey is consumed at a lower rate, in addition to being collected more often than pollen. This means that the cells where pollen is removed are more likely to be filled with honey than pollen. As a result of the higher consumption of pollen and the higher rate of collection of honey, the peripheral region of the comb will consist almost entirely of honey.

From the description above it is clear that the central area of the comb will contain brood whereas the peripheral region will contain honey. The question is then why pollen is located between these two regions. Pollen and honey that is located close to brood are preferentially removed. This means that cells close to brood will be emptied at a relatively high rate. Other cells in the central area are also emptied when the brood is fully developed after 21 days. The only empty cells were pollen can be deposited is thus near the central region of brood. In this area there is a high turnover of brood, pollen, and honey. Pollen is consequently located between the brood and the honey because this is the only place where pollen can be deposited.

1.1.3 Cemetery Formation

Workers in several species of ants form cemeteries outside their nests. When ants die, their bodies are carried out of the nest by workers. The workers then deposit these dead bodies in piles outside the nest. Through this behavior the ants clean their nest and form cemeteries on the outside of the nest. This cemetery formation has been observed experimentally in the ants *Lasius niger*, *Pheidole pallidula*, and *Messor sancta* [4, 11, 12]. Deneubourg et al. [12] observed cemetery formation in the ant *Pheidole pallidula* experimentally. They found that when ants die, workers carry the corpses out of the nest and place them in a pile. Bonabeau et al. [4] report of a more detailed experiment on cemetery formation in the ant *Messor sancta*. They found that if a sufficiently large part of corpses are distributed randomly in the experimental arena at the beginning of the experiment, then the ants will form clusters within a few hours. If there were heterogeneities in the experimental arena, then the clusters were formed such that they followed these heterogeneities, ie. along the edges of the experimental arena if this was small, or around larger objects in the arena.

Based on observations of the ants in their experiments, Bonabeau et al. [4] described the underlying mechanism causing the aggregation of corpses in piles. According to Bonabeau et al. [4] the basic mechanism underlying the aggregation phenomenon is an attraction between dead bodies mediated by the workers. Small clusters of bodies will grow because they attract workers to deposit more bodies. It is thus a positive feedback mechanism that causes the formation of larger and larger clusters of corpses (the larger the cluster the bigger is its attraction on bodies). Even if the basic mechanism underlying the phenomenon is known, the individual behavior of the ants that implements the positive feedback is still unknown.

1.1.4 Brood Sorting

In an ant nest neither workers, brood nor food are distributed randomly in the nest. Rather, they all appear to move or be arranged in predetermined patterns. According to Deneubourg et al. [12] "*. . . the eggs are arranged in a pile next to pile of larvae and a further pile of cocoons, or else the three categories are placed in entirely different parts of the nest.*" The brood in the nest is in other words sorted either in the same location or in different locations within the nest.

According to the description of brood sorting by Deneubourg et al. [12], the different brood are placed in different piles that are separate from each other. A detailed analysis of the brood sorting in the ant *Leptothorax unifasciatus* performed by Franks and Sendova-Franks [15] does, however, show that the brood are not sorted into piles, but rather into a cluster that has a pattern of concentric rings. *Leptothorax unifasciatus* ant colonies occupy flat crevices in rocks. In these crevices the ants form a nest by creating a wall with only one opening that surrounds the nest. Within this wall the ants position the brood in a single cluster, usually away from the single entrance. The ant colonies can be easily observed in experiments by creating an artificial crevice for the ants between two glass plates that are sandwiched together with a piece of cardboard separating the plates. Franks and Sendova-Franks [15] studied the sorting of brood by *Leptothorax unifasciatus* ants by forcing the ant colonies to move from one nest site to another. By forcing the ants to move their brood they were able to observe how the ants form a single cluster of the brood.

When the ant colony is forced to move to a new nest site they initially place the brood randomly in the new nest. However, brood items that are transported later in the emigration tend to be deposited next to brood that are already in the new nest. This happens especially frequently if the two brood items are of the same type. The brood that are deposited in the new nest site are picked up and relocated, even if all the brood has not yet been transported to the new site. Brood items that are picked up are usually deposited in positions that are nearer to larger groups of the same type of brood. As this occurs several growing clusters of brood are formed. According to Franks and Sendova-Franks [15] "*These growing clusters of items compete with another to 'capture' other brood items that are being moved about. Items seem to be more frequently removed from smaller rather than large clusters.*" This process of removal and deposition eventually leads to the formation of a single cluster with the standard pattern.

The brood is distributed in the cluster in a distinct pattern. In the middle of the brood structure there is a cluster of eggs and micro-larvae, and the other different brood stages are arranged around this central cluster in concentric rings. Franks and Sendova-Franks [15] describe the pattern in the finally formed cluster of brood as follows: "*. . . different brood items are arranged roughly in concentric rings in the brood cluster. The standard pattern is for eggs and micro-larvae to be in the middle, with the larger larvae further from the center in order of increasing size. Pre-pupae and pupae are distributed in positions between the outer ring of the largest larvae and those of the next largest size.*" The *Leptothorax unifasciatus* ants are thus capable of performing an annular sort of their brood. That is, to create a structure of annular bands with only one brood type in each band. Even after this structure is formed the ants continue to sort the brood as the different brood grow and enter a new stage in their developmental process.

The underlying mechanism that the ants use to sort their brood in annular bands is still unknown. In their original article, Franks and Sendova-Franks [15] point out that part of the mechanism may involve conditional probabilities of picking up and putting down each brood item that depend on each item's neighbors. They also propose another mechanism that may be responsible for the sorting of the brood. In a packet of muesli there is a self-sorting according to size where smaller items percolate down to the bottom of the packet by moving down the crevices between the larger items [1]. Franks and Sendova-Franks [15] propose that a similar process may occur in the sorting of brood, where smaller brood items move toward the center of the structure by moving between the larger brood. Combined with this there is a force that attracts the brood towards the center of the structure that is caused by the clustering behavior of the ants. By leaving larger gaps between the larger brood the ants will thus be able to move the smaller brood between these larger brood, and thereby position the smaller brood items closer to the center of the structure. The mechanism that sets the distance between the various types of brood is however not known, and may for example use pheromones or metabolic waste products to determine the correct distance between various brood items.

1.2 Self-Organization

The previous sections have described collective behaviors by insects that result in complex patterns at the global level. The complex patterns emerge at the global level of the social insect societies even though the individual elements, the insects, are only capable of performing very simple behaviors. Theories of self-organization were originally developed in the context of physics and chemistry to describe the emergence of macroscopic patterns out of processes and interactions defined at the microscopic level [6]. The theory of self-organization has then been extended to social insects to show that complex collective patterns may emerge from the interaction among individuals with simple behavior. Camazine et al. [9] have given the following definition of self-organization: "*Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.*" If this is to be interpreted in terms of an insect society it means that the collective pattern that is displayed by the system, like the pattern of army ant raids or the clustering of dead bodies, is an emergent property of the system rather than a property imposed on the system by an external ordering influence like a blueprint or a supervisor. Self-organization has also been defined as "*a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components*" [6]. A self-organizing system has four basic ingredients: positive feedback, negative feedback, amplification of fluctuations, and multiple interactions. These four ingredients are described below.

The first ingredient of a self-organizing system is positive feedback. Positive feedback promotes changes in a system, usually in the same direction as the system is currently moving. It has been described as "*simple behavioral rules of thumb*" that promote the creation of structures [4]. An example of this is the aggregation of some birds when building nests. Nesting in groups may provide benefits in terms of protection from predators, or ease in finding food. Because of this, birds may prefer to nest where other birds nest. If some birds for some reason decide to nest at a particular location, other birds will be attracted to nest on the same site, because they want to nest where other birds nest. This constitutes a positive feedback in the form that the more birds that nest at a location, the more birds will be attracted to nest at that location, and as a result the nest site will increase in size. Positive feedback can have a snowballing effect in that it reinforces an initial change in a system in the same direction as the initial deviation. In fact, explosive chemical and nuclear reactions are classic examples of positive feedback. Positive feedback consequently has a destructive potential in that it can create a cascade of change in one direction.

The second ingredient of self-organization is negative feedback. Negative feedback counterbalances positive feedback and helps to stabilize the collective pattern. As mentioned above, positive feedback has a destructive potential in that it can have a snowballing effect. This destructive potential can be counterbalanced by negative feedback. Negative feedback can work in the opposite direction of a positive feedback mechanism, and thus provide inhibition to offset the amplification provided by the positive feedback. By doing this, negative feedback can help shape the amplification into a specific pattern. Negative feedback can take the form of saturation, exhaustion or competition. It often arises from purely physical constraints, for example in the form of depletion of a resource. The building of a structure may for example be inhibited by the reduced amount of building material that is available.

Self-organization relies on amplification of fluctuations, which is its third basic ingredient. These fluctuations may take the form of random walks, errors, deposition of objects in random positions, and so on. This randomness is a key part of self-organizing systems. The random events that occur in a self-organizing system are crucial to the system because they allow the system to discover new solutions. The random fluctuations that occur in the system can be amplified, and through this, new structures or solutions may be found. For example in the case of ant foraging, a forager may get temporarily lost from its current trail and discover a new rich food source. This food source can then be exploited by the ant colony. If it were not for the error of the first ant this food source would not be discovered, and the colony would not be able to exploit it.

The last ingredient of self-organization is multiple interactions. These interactions may be either directly between the individuals of the system, or indirectly through the elements that are handled by the individuals in the system. In this latter form two individuals may interact when one individual modifies the environment, and the other individual responds to the modified environment at a later time. This modification of the environment may, for example, take the form of deposition and removal of objects.

A self-organizing system can usually be characterized by a few properties [6]. They need not be present, but their presence indicates that the observed phenomenon is self-organized. The first property is the creation of spatiotemporal structures in an initially homogeneous medium. These structures may take the form of nest architectures, foraging trails, or social organization of the individuals in the system. An example of this is the cemetery formation by ants described in section 1.1.3. Before any dead corpses are moved out of the nest, the surroundings of the nest are homogeneous in the sense that there are no clusters of dead bodies. As the workers clean the nest by picking up corpses and deposit these in piles outside the nest, the surroundings are no longer homogeneous, but contain spatiotemporal structures in the form of piles of dead bodies.

The second property of self-organizing systems is the possible coexistence of several stable states, or multistability. Since structures emerge by amplification of fluctuations, any such fluctuation may be amplified and the system can thus converge to any one of several possible stable states depending on the initial conditions of the system. For example, when there are two identical food sources presented at the same distance from an ant colony, one of them is eventually massively exploited whereas the other is neglected. The quality of each food source is in theory identical, but based on random initial events one source will be exploited while the other is left alone [4].

The last property is the existence of bifurcations when parameters are varied. At bifurcations the behavior of a self-organizing system changes dramatically. Self-organizing systems thus consist of the four basic ingredients positive feedback, negative feedback, amplification of fluctuations, and multiple interactions. All of these ingredients must be present in a self-organizing system. Furthermore, a self-organizing system is usually characterized by the emergence of spatiotemporal structures in an initially homogeneous medium, multistability, and bifurcations.

1.3 Stigmergy

The concept of stigmergy was first introduced by Grassé [17] in 1959, who observed building behavior by termites, and introduced stigmergy to explain some of the observations he made. He observed that when termites encountered certain configurations of the ongoing construction of the nest, they were stimulated to a high degree of activity and to the addition of further building material to the encountered configuration. The structure would then grow until it was complete, and the termites would turn to create another such structure or begin on a different task. From these observations Grassé [17] inferred that important determinants of an individual's building behavior were stimuli from work previously accomplished, either by the individual itself or by other individuals.

The basis of stigmergy is that previous activities of individuals influence activities at a later stage. It is in other words a form of indirect communication that works through modifications to the environment. Grassé [17] saw stigmergy as consisting of a series of stimulus–response sequences. One specific stimulus triggers a specific response that alters the environment in a way that causes a different stimulus in the future, that again triggers a different response. An example of such a stimulus–response sequence is evident in the construction of nests in termites of the genus *Macrotermes* [4]. A configuration of soil pellets triggers a response from a termite that transforms the configuration. This new configuration of soil pellets will then trigger a different response from the same or a different termite. The basis of stigmergy is thus that individuals respond to the environment in a way that alters the environment. This change then causes the same individual, or a different individual, to respond differently to the changed environment. By making alterations to the environment the individuals can thus organize their own activities, or the activities of a group of individuals.

The basic principle of stigmergy is summarized by Holland et al. [21]: "*Traces left and modifications made by individuals in their environment may feed back on them. The colony records its activity in part in the physical environment and uses this record to organize collective behavior.*" The colony can record its activity in the environment in several ways, including the use of pheromones, material structures, or spatial distribution of colony elements. Although the basic principle of stigmergy is agreed upon, views of what stigmergy is and how it should be seen differ in the literature. According to Holland et al. [21] "*...stigmergy is a mechanism that allows an environment to structure itself through the activities of agents within the environment.*" Here the state of the environment and the distribution of agents will determine changes in both the environment and the distribution of agents in the future. Holland et al. [21] thus see stigmergy as a mechanism that can explain how an environment changes. They relate stigmergy to self-organization by saying that stigmergy may occur independent of self-organization, but that only those instances of stigmergy that give rise to self-organization will produce useful effects. Camazine et al. [9] views stigmergy from a different perspective. They regard stigmergy not as a complete theory of building activity, but as an important concept that helps explain the flow of information between the builders. From this perspective stigmergy is only one of the mechanisms that is present when a group builds a structure. Stigmergy is thus a source of information that the individuals use to determine appropriate actions.

This is the view on stigmergy that we take in our work. We see stigmergy as a way of indirect communication between the agents that the agents use to modify their actions. That is, the agents modify the environment in some way, and at a later stage any one individual may respond to this change in a different way. The agents thus communicate with each other indirectly, through modifications to the environment, to organize their collective behavior.

1.4 Problem Definition

Earlier in this chapter we described patterns that are formed by social insects. It was described how honey bees create concentric regions of brood, pollen, and honey on the wax combs in their hive, and that ants are able to cluster and sort objects in different patterns. The way that the social insects solve these pattern formation tasks have attractive features. The patterns are formed by swarms of homogeneous individuals that each make use of only local stimuli to select what actions to perform. This enables the swarms to solve the tasks in a flexible and robust way. The ants are able to adapt to changing environments and they are robust because the functioning of the swarm as a whole is not dependent on any one individual. If an individual dies its work can easily be replaced by another individual. The swarm thus has robustness and reliability through redundancy.

Inspired by the patterns formed by swarms of insects, we want to replicate this behavior in computer simulations. We want to create swarms of agents that, like the ants described, are capable of clustering and sorting objects in different patterns. The agents created will only make use of local information when deciding on which actions to perform, and through the interactions of a swarm of identical agents the collective patterns should emerge. To aid in the construction of these swarms of agents we make use of the introduced concepts of self-organization and stigmergy. Self-organization shows how global patterns can be formed based on the actions of individuals, and thereby gives a theoretical basis for the notion that a group of simple agents can perform complex tasks through interactions. Stigmergy does, on the other hand, explain how this interaction can be done indirectly through modifications to the environment.

It is difficult to design self-organizing systems. The traditional approach to creating self-organizing systems has been to define the actions of the individuals, and then see if this leads to the desired properties at the global level of the swarm of individuals. The paths to problem solving are however not predefined but emergent in these systems, and result from the interactions among individuals, and between individuals and their environment, as much as from the behaviors of the individuals themselves. It is therefore very difficult to determine the individual actions that will lead to the desired global patterns.

Because of the problems mentioned above with designing self-organizing systems, we have chosen an evolutionary approach to create swarms of agents that are capable of forming the desired patterns. Rather than hand-coding the individual behaviors we evolve the behaviors. The concept of self-organization shows how individual behaviors can lead to global patterns. This concept can also be used the other way around, by saying that swarms that create the desired global pattern will have individuals that display the correct behavior for forming these patterns. We make use of the connection from global patterns to the individuals' behavior in our evolution of the correct behavior. In our work we will evolve individual behavior by evaluating the patterns formed at the global level of the swarm. At each stage of the evolution a variety of different behaviors at the individual level will exist. The desirability of the different evolved behaviors will be evaluated by allowing a swarm of identical agents, with each set of evolved behaviors, to interact for a certain amount of time in an environment containing objects that can be moved. The different sets of behaviors will then be evaluated in terms of the global patterns that the swarms of individuals form. Behaviors that lead to desirable patterns will be selected for reproduction, and through this evaluation of the global patterns that are formed, the individual behaviors will be evolved.

We have chosen to create self-organizing systems that solve three different tasks of pattern formation. These are inspired by the cemetery formation and brood sorting done by ants that were described in sections 1.1.3 and 1.1.4. The first task that we wish to solve is clustering of identical objects similar to the cemetery formation performed by ants. This is a fairly simple task to accomplish, and the successful solution to this task will show that our approach to the creation of self-organizing systems is viable. The second task that we will address is the sorting of different types of objects into separate piles. In this task the agents must cluster different types of objects into separate clusters, such that each cluster contains only objects of one type, and no two clusters contain objects of the same type. This type of sorting is called *patch sorting* [27]. In our work we will create solutions that are capable of forming separate clusters of two and three types of objects. The last task of pattern formation that we address is the formation of a cluster with an annular pattern similar to the brood sorting in *Leptothorax unifasciatus* described by Franks and Sendova-Franks [15]. This task is termed *annular sorting*, and the swarm of individuals must create a central cluster of one type of object that is surrounded by concentric bands of other objects, with each band containing only one type of objects. This is a very difficult pattern to form and there have not previously been any successful attempts at creating such a pattern that are both compact and where the bands of each type of objects are well separated.

After describing the different tasks of pattern formation that we are going to address we are able to give a more formal definition of the problem to be addressed:

> Create swarms of homogeneous agents that are capable of clustering, patch sorting, and annular sorting by evolving individual behavior through the evaluation of patterns formed at the global level of the swarm.

1.5 Motivation

The two main motivations of the field of swarm intelligence are to gain a better understanding of nature, and to develop better techniques within computer science. The motivations thus have a biological perspective and a computer science perspective. The motivation for addressing the problem of sorting by a swarm of agents can be seen from both of these perspectives. From the biological perspective the motivation is that the underlying mechanisms responsible for the behavior leading to sorting of objects is yet unknown. By evolving behaviors that lead to the desired global patterns in simulations, an understanding of the underlying mechanisms may be established. In our work we use agents that are equipped with a limited sensory apparatus and a few simple actions, and thereby say something about the minimal requirements at the individual level that are needed to create the patterns at the global level.

From the computer science perspective the main motivation is that the application of social insect behavior may lead to better techniques. Social insects have many attractive features such as individual simplicity, flexibility, scalability, and robustness and reliability through redundancy. Algorithms inspired by social insects have previously been applied successfully to complex problems of optimization, scheduling, routing, and assignment [13]. They have, for example, been used to provide solutions to the well known traveling salesman problem, to routing in telecommunication networks, and to schedule jobs in factories [5, 13]. Computer models based on the clustering and sorting by insects may lead to improved algorithms in a wide range of areas, such as search, data mining, experimental data analysis, graph partitioning, and classification problems. The amount of information stored in computers around the world increases rapidly, and ideas from the insect world can help organize, search, and visualize these massive amounts of information. The stored information often consists of complex data that has hidden commonalities. Algorithms based on the brood sorting performed by ants may, for example, be used to visualize structures in the stored data. These algorithms do not need any initial information about the existing structures[3], and can visualize the data either by forming clusters of data with similar characteristics, or by organizing the data in an annular structure of concentric rings. Social insect behavior also has attractive features in terms of scalability. To solve problems of increasing magnitude, all that is needed is to add more agents. This will enable both new and larger problems to be solved, in addition to more efficient performance of solutions to existing problems.

In addition to the two motivations given above that are connected to the general motivations of the field of swarm intelligence, we also have two motivations of our own. The first is to show that artificial evolution is a viable approach to the creation of swarms capable of clustering and sorting. Previous research within this field has usually been done by first designing the individual behavior of the agents. Then a simulation has been performed and the emergent patterns have been observed. If the desired patterns are not observed, the individual behavior is tweaked until it leads to the desired emergent pattern. We wish to show that artifical evolution is an alternative approach to this specification-and-trial process. The second motivation is that we wish to show that simple individual behavior can lead to the solution of complex tasks. Previous research has shown that swarms of simple agents can perform clustering and patch sorting. The task of annular sorting is, however, considerably more difficult than these tasks. We therefore intend to show that a homogeneous swarm of simple agents is capable of solving such a difficult task as annular sorting.

[3]Conventional clustering methods require an initial specification of the number of clusters that are present.

1.6 Structure of the Thesis

In this chapter we have described the biological background for the work that we have done. We have also given a formal definition of the problems that we intend to solve, and the motivations for investigating these problems. In the next chapter we present previous research that is related to our work. The work presented constitutes the most important research that has been done concerning the application of swarms of agents to clustering and sorting. After describing related work chapter 3 describes the methods of our work. In this chapter we describe both the components of the evolved solutions as well as aspects of the evolutionary method. Chapter 4 presents the five experiments that we have performed. For each experiment the evolutionary progression is described as well as characteristics of the final evolved solution. In chapter 5 we discuss certain aspects of the solutions. Chapter 6 ends the thesis with some concluding remarks.

Chapter 2

Related Work

The tasks of clustering and sorting have only recently been addressed in the field of swarm intelligence. The article that signifies the birth of this field of research was written by Deneubourg et al. [12] in 1991, and since this a limited amount of work has been published. In this chapter computer simulations and robotic experiments that address the tasks of clustering, patch sorting, and annular sorting are presented. For each task both computer simulations and robotic experiments are described.

2.1 Clustering

The task of clustering can be defined as the grouping of objects of the same type in a continuous area [27]. The modeling of this behavior in ants was the first to be investigated, and these works have later been extended to patch sorting, experimental data analysis, and annular sorting.

2.1.1 The Basic Model

In their seminal article Deneubourg et al. [12] present two closely related models to account for cemetery formation and brood sorting in ants. They say they have been inspired by how ant colonies sort their brood, and present a distributed sorting algorithm that is to be used by robot teams. However, they never actually go beyond a computer simulation of the model and so the algorithm is never actually implemented in physical robots.

The model is based on randomly moving agents that come across objects in the environment. The key ingredient of the algorithm is the modulation of probabilities for whether the agents should pick up or put down objects. When they detect an object, the probability of picking up the object is greater if the object is isolated. This means that if the object is encountered in a neighborhood that is sparsely populated with objects, there is a higher probability of the object being picked up than if it is encountered in a densely populated neighborhood. When carrying an object, the agent's probability of putting down the object is greater if the agent is currently in a neighborhood that is densely populated with objects. Deneubourg et al. [12] claim that either one of the pick up and drop down behaviors is sufficient to create a clustering behavior by the agents, but that the two behaviors combined cause clusters to be created faster.

The environment in which the agents reside is a square network of points, and at the beginning of a run the agents and objects are dispersed randomly in the environment. Only one object, one agent, or one object and one agent are allowed to occupy a point in the environment. At each time step in the simulation the agents are stepped in random order. Each step by an agent consists of a movement and a decision of whether to pick up or put down an object. The agents pick a random direction to the north, east, south, or west and move to the neighboring cell in this direction if there is not a wall in that direction, and the point is not already occupied by another agent. If the agent has moved to a point that contains an object, and the agent is not currently carrying an object, it will calculate a probability for whether it should pick up the object or not. The probability P_p that the agent will pick up the object is greater if the immediate environment is sparsely populated with objects. The function used to calculate the probability for whether the agent should pick up an object is given in Equation 2.1.

$$P_p = \left(\frac{k^+}{k^+ + f} \right)^2 \tag{2.1}$$

In the equation, k^+ is a constant and f is an estimation of the density of objects in the neighborhood. The probability of picking up an object decreases with f from 1 when f is 0, to 0.25 when $f = k^+$ and less as f tends to 1. If the agent has previously picked up an object, then for each move it makes that leads to an empty point, the agent will decide whether to put down the object or not. The probability that the agent will put down the object, P_d, is greater if the neighborhood is densely populated with objects. This probability is calculated using the function given in Equation 2.2.

$$P_d = \left(\frac{f}{k^- + f} \right)^2 \tag{2.2}$$

Here f is the same as before and k^- is a constant. The probability of putting down a currently carried object increases with f from 0 when f is 0, to 0.25 when $f = k^-$ and more as f tends to 1.

To estimate f, the fraction of nearby points occupied by objects, Deneubourg et al. [12] provide the agents with a short term memory. This memory records what has been encountered by the agent in the previous m steps. If the agent encounters an object after a move, this is recorded in the next position of the memory. Otherwise a 0 is recorded representing that no object was encountered. The memory functions as a queue, so that when the agent has moved $m + 1$ times the oldest record in the memory is replaced by the current recording. To calculate f, the number of objects recorded in the memory is divided by the size of the memory, m. In other words, f is the fraction of objects recorded in the short term memory of the agent.

In their article Deneubourg et al. [12] give an example of the result of an experiment using their algorithm. In the experiment they use 100 agents and 1500 objects in an environment that has 290×200 points. The agents have a memory, m, of 50 steps, the constant k^+ is set to 0.1, and the constant k^- is set to 0.3. As the algorithm is run, small evenly spaced clusters rapidly form that later merge into fewer larger clusters. Throughout the run of the algorithm the clusters are constantly having objects removed and added to them. This causes a drift of the clusters across the environment such that they may meet each other. When two clusters meet they merge into one cluster. Although a few larger clusters are formed, Deneubourg et al. [12] do not report of a single cluster forming at any stage. Bonabeau et al. [4] has replicated the algorithm in a run of their own, and they are not capable of forming a single cluster either. It is possible, however, that a single cluster may eventually arise, but that as a few large clusters arise it will take a long time for them to merge.

2.1.2 A Minimal Clustering Model

Martin et al. [25] have recently argued that the intelligence attributed to the ants in Deneubourg et al's [12] model is not necessary for the clustering of objects. The intelligence of the ants in the latter model (section 2.1.1) is reflected in the ants' ability to estimate the cluster size by using their short term memory, and act accordingly by calculating probabilities. Martin et al. [25] propose what they term a 'minimal model' that shows that much simpler ants are capable of forming clusters without the need to calculate pick up and put down probabilities, and with only a local perception of the size of the clusters. The reason they call their model a minimal one is because the ants in their model only require a minimal knowledge about their environment in order to perform clustering.

The main difference between the two models is that the dynamics of Martin et al's [25] model is provided by alternating a pick up and deposition rule with a random walk rule. In addition to this, the ants are not allowed to walk over corpses as they could in the other model. The pick up behavior of the ants in Martin et al's [25] model is as follows. When the ant is not currently carrying an object, it investigates the contents of its eight surrounding cells. If one of these cells contains an object the ant will pick up the object. If more than one of the surrounding cells contain an object the ant will choose one of these objects to pick up randomly. On the other hand, if the ant is currently carrying an object it will attempt to drop its object, provided that at least one of its eight surrounding cells contain an object. The ant will then choose one empty cell in its neighborhood at random and place its carried object in this cell. Martin et al. [25] believe that it is the pick up and deposition rules that are significant in relation to Deneubourg et al's [12] model, although they also use a different movement rule. The movement of the ants is one cell at the time, but they will move in the same direction for a specified number of steps. The ants will randomly select a new direction and length of its walk in three different situations. First, if the ant has walked the specified number of steps, it will choose a new direction and length for its walk. Second, if there is an object in the trajectory of the ant, and the ant does not choose to pick up this object, the ant will pick a new direction and length. Third, to include obstacle avoidance, the ants will choose a new direction and length of its walk if there is an obstacle in its trajectory that it cannot pass. At each time step in the model all the ants are stepped sequentially, and in the same order each time. Each ant will thus move to the nearest cell along its current trajectory if this is possible. If not, it will choose a new direction and length of its walk. After the ant has moved, it will attempt to either pick up or deposit an object.

Martin et al. [25] provide the result of a run of their model. In the experiment 100 ants and 1000 objects are distributed randomly in a 400×250 cell environment with periodic boundary conditions. The ants pick a length of their walk randomly from the interval $[1 \ldots 100]$. After 1.3 million iterations a cluster that contains all the objects is formed. To explain why a single cluster is formed, Martin et al. [25] propose an explanation in terms of the underlying mechanism that biases the dynamics so as to form only one large cluster. Each cluster in the environment is fed and depleted at the same rate, but because the dynamics is noisy there are fluctuations that temporarily favor either growth or depletion of certain clusters. This means that every cluster may increase or decrease in size at some point. However, once a cluster shrinks to a certain size there is no way to make it grow again. As a consequence, statistical fluctuations may cause clusters of any size to vanish, but the probability of this occuring decreases with the size of the clusters. In short, small clusters have a higher probability of disappearing than larger clusters. Because of this, smaller clusters will disappear and larger clusters will increase in size until there is only one remaining cluster. Martin et al. [25] summarize the underlying mechanism as follows: "*. . . the dynamics is biased towards forming larger clusters, because small ones are more likely to disappear under statistical fluctuations and clusters cannot build up from nothing. We term this effect a fluctuation-threshold effect.*" They thereby say that the main ingredient in the cluster formation process is the action of taking an object from one cluster and adding it to another, and that the agents are purely a local implementation of this action.

In contrary to Deneubourg et al's [12] model, Martin et al's [25] model is capable of forming a single cluster. However, the formed cluster is not as compact as the clusters created by the former model. Also, neither of the clusters formed by the two models is static. In both cases the formed clusters are dynamic as the agents are constantly picking up objects from the clusters and dropping them somewhere along the clusters' perimeter.

2.1.3 Clustering by Robots

Beckers et al. [2] were among the first to design a group of identical robots that were capable of clustering objects. Inspired by the clustering of corpses performed by ants, they decided to develop a system using multiple robots to cluster pucks that are initially dispersed in the environment. In order to make the robots capable of displacing the pucks the robots were equipped with a C-shaped gripper. This gripper allowed the robots to push a number of pucks at the same time. In addition to enabling the robots to push objects, the gripper also aids the robots in estimating the density of the pucks in their neighborhood. This is done by adding a sensor that measures the resistance of the pushing of the pucks. If there is a high resistance to the pushing, this means that there is a high density of pucks in the local environment, and if the resistance is low, there are not many pucks in the immediate environment. The resistance applied to the gripper activates a microswitch if the resistance equals the pushing of three pucks. In addition to the microswitch, the robots are equipped with two infrared sensors. These enable the robot to detect and avoid obstacles.

The behavior of the robots is based on the subsumption architecture [7], and they have three possible behaviors with only one active at any time. The default behavior is active when no sensors are activated. This means that no obstacles are detected and that the robot may push none, one, or two pucks. When this behavior is active the robot moves in a straight line. This behavior continues until the microswitch on the gripper is activated or an obstacle is detected. When an obstacle is detected the robot activates the obstacle avoidance behavior. This causes the robot to turn on the spot away from the obstacle at a random angle. After the robot has turned the default behavior takes over and the robot moves in a straight line. Because of the C-shaped gripper, any pucks that the robot was pushing previous to the encounter with the obstacle are retained in the gripper through the turn. Thus the robot does not loose any of its currently pushed pucks when facing an obstacle. The third behavior is triggered by the microswitch. This switch is activated when the robot pushes three or more pucks in its gripper, or when it encounters a cluster of three or more pucks. When this happens the puck-dropping behavior is activated. This causes the robot to reverse its motors for one second and thereafter to turn at a random angle, before continuing with its default behavior of moving in a straight line. The reversal of the motors causes the robot to drop the pucks that it was currently pushing. The pucks are consequently dropped at a location that has a high perceived density of pucks.

In the experiments performed by Beckers et al. [2] the robots are placed in the center of a square arena of 250×250 centimeters, each pointing in a different direction. 81 pucks are placed regularly in a grid pattern with 25 centimeters between each. To measure the progress of the robots, every ten minutes of runtime the robots are stopped and the positions of the clusters of the pucks are recorded. The definition of a cluster used is a group of pucks that are separated by no more than one puck diameter. In their experiments three successive qualitative phases are observed irrespective of the number of robots that are used to perform the clustering. In the first phase at the early stage of the clustering process, the robots move and collect pucks in their gripper one at a time. When the robot has collected three pucks in its gripper, it backs up and drops a cluster of the three pucks. This causes the formation of small clusters containing three or more pucks. These clusters can not be moved and are formed within a short time from the beginning of the experiment. In the second phase, the robots remove a few pucks from the existing clusters by striking them at an angle. These pucks are then added to other clusters as the robot collides with them. In this phase some of the clusters grow rapidly, causing the formation of a few relatively large clusters. The third phase has the longest duration. In this phase there are occasional removals of pucks from the large clusters that are later added to one of the other clusters, or the same cluster that they were removed from. This phase ends with the formation of a single cluster. The formed cluster is, however, not static as pucks are constantly being removed and added to the cluster.

Beckers et al. [2] provide an explanation as to how the clusters are formed by the robots. The robots may approach the clusters in three different states. These states are defined by the robots carrying none, one, or two pucks. If the robot is not pushing any pucks, then it may remove one or two pucks from a cluster that it encounters. In the second state the robot is already pushing one puck, and so it may either add this puck to the encountered cluster or it may remove one puck from the cluster and consequently end up pushing two pucks. The pushing of two pucks is the third state, and when the robot encounters a cluster in this state it may only add the two pucks to the cluster. When evaluating the effects of these states in terms of the size of the clusters, and when taking the paths of the robots and the angle of approach into account (to withdraw pucks from a cluster the robot must approach it tangentially, and to add pucks it must have a frontal approach), Beckers et al. [2] have come to the following conclusions: 1) When a robot that is not pushing any pucks encounters a cluster it can only remove pucks, and the probability of this occuring decreases with increasing cluster size. 2) Robots that encounter a cluster while pushing one puck will tend to add the puck to the cluster because the probability of dropping the puck is greater than the probability of removing a puck from the cluster. The probability of this occuring increases with increasing cluster size. 3) If the robot is pushing two pucks it can only add to the size of a cluster, and the probability of this occuring again increases with cluster size. Because of this, larger clusters will always be more likely to gain pucks and less likely to loose pucks than smaller clusters. The two behaviors of the robots, adding pucks and removing pucks from a cluster, will thus increase the probability of the same outcome happening at the same location in the future.

2.2 Patch Sorting

Patch sorting involves two or more classes of objects that are individually clustered, while the clusters of each class are separated from each other. This means that a successful solution to this task must be capable of separating different types of objects, such that each type of object forms a continuous group of objects that are not incident with objects of other types. To solve this problem, both Deneubourg et al. [12] and others have extended their basic model. The research concerning the task of patch sorting has split into two directions. The first direction is an extension and application of Deneubourg et al's model [12] to the field of exploratory data analysis where the objects to be sorted can be characterized by a continuous similarity metric. The other direction is in robotic experiments where robots sort objects that have a discrete similarity, meaning that they are either identical or not.

2.2.1 Extension of the Basic Model

Deneubourg et al. [12] have provided a simple extension of their clustering algorithm to perform sorting of two types of objects. The behavior of the ants is the same and the functions for calculating the probabilities are also the same except for the calculation of f (see Equations 2.1 and 2.2). When there are two types of objects, the memory of the ants consists of recordings of either no object, or of an object of either of the two types. In order to compute f the type of the object being considered to be picked up or put down must be taken into consideration. To compute f the fraction of objects in the memory of the same type as the object being considered is computed. For example, if there are two types of objects, A and B, and the current memory of the ant of length ten is 0ABBA00AA0, then if an ant is considering if it should drop or pick up an object of type A, then $f = 0.4$.

The functioning of the algorithm when used with two types of objects has been described by Deneubourg et al. [12]. Isolated objects in the environment are picked up, and small clusters form due to random fluctuations. These small clusters encourage the agents to drop similar objects in the vicinity of the clusters as the density of the object type in the neighborhood increases. This positive feedback mechanism causes the clusters to grow, and any nearby object of another type will be engulfed in the cluster and thus make it 'isolated'. This causes the different type object to be more likely to be picked up by passing agents. According to Deneubourg et al. [12]: "*Sorting is the consequence of this clustering and crowding out behavior.*"

2.2.2 Clustering of Complex Data

Lumer and Faieta [24] have generalized Deneubourg et al's model [12] to apply it to exploratory data analysis. The algorithm they propose consists of projecting the space of the attributes of the data onto some lower dimensional space (in their article the space is two-dimensional and the attributes are real-valued), such that clusters occur in this space. Attribute distances between objects in these clusters should be smaller than the attribute distances of objects belonging to different clusters. The model consists of ants that perform a random walk on a two-dimensional grid. The dimensions of the grid are such that its number of sites exceeds the number of ants by about an order of magnitude, and the number of objects exceeds the number of ants by at least another order of magnitude. The model has discrete time steps, and at each step an ant is selected at random. This ant then decides whether it should pick up or put down an object at its current location. Each time step ends with the selected ant moving to one of its four adjacent nodes.

The decision of whether to pick up or put down an object is the key ingredient of the algorithm. The function that calculates the probability for whether an ant should pick up an item at its current location resembles the function given by Deneubourg et al. [12] (see Equation 2.1), and is given below in Equation 2.3.

$$P_p(o_i) = \left(\frac{k_1}{k_1 + f(o_i)}\right)^2 \tag{2.3}$$

In the equation, the constant k_1 plays a similar role to the constant k^+ in Equation 2.1 and the estimation of the density of the local neighborhood, f, has been replaced by the function $f(o_i)$. Like the probability function for picking up an object defined by Deneubourg et al. [12], this function causes objects that are located in a neighborhood with a high density of dissimilar objects to have a higher probability of being picked up than objects that are located in a neighborhood with similar objects. There is, however, a major difference between the function proposed by Deneubourg et al. [12] and this new function. Whereas Deneubourg et al. [12] only considered objects that were either identical or not, Lumer and Faieta [24] use objects that have some degree of similarity to each other. Because of this, some form of continuous similarity metric must be used when estimating the resemblance of an object to the other objects in its neighborhood. The similarity metric used, $d(o_i, o_j)$, is simply the Euclidean distance between objects o_i and o_j. This similarity measure is used in the function $f(o_i)$ to measure the similarity between object o_i and the other objects in its $s \times s$ neighborhood. The density function $f(o_i)$ is shown in Equation 2.4.

$$f(o_i) = \begin{cases} \frac{1}{s^2}\sum_j \left(1 - \frac{d(o_i,o_j)}{\alpha}\right) & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

In the equation the term s^2 is a normalizing term that introduces a density dependency in the function, and the constant α scales the dissimilarities. If the objects in object

o_i's neighborhood are similar to o_i, then $f(o_i)$ will tend towards 1 and the probability of picking up the object is low. However, if the objects surrounding object o_i are dissimilar to o_i, $f(o_i)$ will be small and the object will have high probability of being picked up. When the ant is deciding whether an object should be put down or not, the probabilities are reversed. If the ant is currently in a neighborhood with objects that are similar to the object being carried, then the ant has a high probability of putting down the object. Likewise, if the ant is in a neighborhood with objects that are dissimilar to the carried object, then the probability of putting down the object is low. The probability function for the deposition behavior is given in Equation 2.5, where the constant k_2 has a similar role to the constant k^- in Equation 2.2.

$$P_d(o_i) = \begin{cases} 2f(o_i) & \text{if } f(o_i) < k_2 \\ 1 & \text{otherwise} \end{cases} \tag{2.5}$$

When Lumer and Faieta [24] ran their experiments they discovered that the algorithm continually produced more clusters in the projected space than there were clusters in the initial distribution. To correct this tendency to create more clusters than were present in the initial distribution of the data, Lumer and Faieta [24] added three features to the algorithm. The first feature is ants with different movement speeds that modify the probability of picking up and dropping objects. The speed, v, of an ant is the number of grid units walked in one step in its current direction. The speed of the ants is distributed uniformly in the interval $[1 \ldots V_{max}]$ and in their experiments Lumer and Faieta [24] uses $V_{max} = 6$. The speed of the ants is taken into account in the density function, $f(o_i)$, such that the speed of the ants influences the probabilities of picking up and dropping objects. The modified density function is given in Equation 2.6.

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_j \left(1 - \frac{d(o_i, o_j)}{\alpha\left(1 + \frac{v-1}{V_{max}}\right)}\right) & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

Due to the modified density function, fast moving ants are not as selective as the slower moving ants in their estimation of the similarity of an object and the objects in its neighborhood. The addition of this feature to the algorithm allows the algorithm to perform a coarse-grained sorting, by the fast moving ants, while at the same time doing a finer sorting with the slower moving ants.

The second added feature is a short-term memory in the ants. This memory contains the last m items that the ant has dropped together with the location where the item was dropped. Every time the ant picks up an object, it is compared to the objects stored in the memory, and the ant moves towards the location where the most similar object was put down. The addition of this behavior causes a reduction in the number of similar clusters that are created. The third added feature is a behavioral switch which allows the ants to switch from gathering elements to destroying clusters if they have not picked up an element in a specified number of time steps. This enables clusters that are formed and which contain 'acceptably' similar objects to be broken up and their objects to be moved to more appropriate clusters.

2.2.3 Exploratory Data Analysis

The work performed by Lumer and Faieta [24] inspired others to investigate similar algorithms with the aim of sorting complex data. These algorithms are applied to large collections of data and have the advantage that the number of clusters present in the data does not need to be known in advance. In the following, two different ant-based algorithms for the task of data clustering are presented.

AntClass

Monmarché et al. [22] have presented a hybrid algorithm for data clustering. The algorithm combines an ant-algorithm with the K-means algorithm, which enables it to automatically discover clusters in numerical data without prior knowledge of the number of classes. In the algorithm the objects are vectors of the same length containing numerical values, and the similarity between two objects is measured by the Euclidean distance between their corresponding vectors. When the algorithm starts the objects are scattered randomly on a two-dimensional square grid with periodic boundary conditions. The size of the environment is decided by the number of objects, and the relationship between the number of objects, n_o, and the number of cells in the environment, m, is $m = 2 \times \sqrt{n_o}$. A cell in the environment may contain one ant, one object, or a heap of objects. A heap is a collection of two or more objects located on the same cell. The algorithm runs in discrete steps and at each step all the ants in the environment are allowed to perform their actions. Each ant can move in eight directions, and when it is allowed to move it has a certain probability of continuing to move in its current direction, or else it selects another direction randomly. The ants also have a speed which specifies how many cells in the current direction they will move in a single step. These are two of several parameters that may differ between the various ants in the environment. The algorithm consequently has a heterogeneous population of ants.

After the ants have moved, they can either perform a pick up or a deposit action depending on whether they are currently carrying an object or not. If the ant is not carrying an object, it looks for an object or a heap of objects in its eight surrounding cells. If an object or a heap is found, there are three different configurations of objects that may be encountered. First, if a single object is discovered this will be picked up with a fixed probability. Second, if a heap of two objects is discovered, the ant has a fixed probability of picking up either of the two objects, thereby destroying the heap. Third, if a heap of more than two objects is found the ant must consider the similarity of the objects in the heap. If the objects are sufficiently dissimilar the ant will pick up the most dissimilar object of the heap. As with the picking up of objects, there are three different situations that may occur if the ant is currently carrying an object. The ant again considers its eight surrounding cells in turn. If the ant finds an empty cell it puts down its currently carried object with a fixed probability. Second, if the cell already contains an object, the ant will put down its carried object provided that it is sufficiently similar to the object already in the cell. Finally, if the cell contains a heap of objects the ant will put down its object provided that it is closer to the center value of the heap than the most dissimilar object of the heap.

When the ant-algorithm has gone through a specified number of iterations the K-means algorithm is applied to the objects using the initial partition of the objects created by the ant-algorithm as a starting point. The K-means algorithm computes the center of each heap, and then assigns each object to the heap whose center is closest to the object. The application of the K-means algorithm helps to assign isolated objects to clusters, as well as reassigning objects that have been assigned to the wrong cluster by the ant-algorithm. After the K-means algorithm has finished, the ant-algorithm is again applied to the objects, only now the ants are capable of carrying heaps of objects and are unable to break up existing heaps. By applying the ant-algorithm again in this manner the number of heaps is again reduced, providing a more accurate number of clusters compared to the underlying distribution of data. When the modified ant-algorithm has run for a fixed period, the K-means algorithm is again applied to the objects.

The AntClass algorithm thus consists of four main steps: 1) Ant-algorithm creating an initial clustering. 2) K-means algorithm using the initial partition provided by the ant-algorithm. 3) Ant-algorithm applied to heaps of objects, and finally 4) K-means algorithm providing a final clustering of the data.

ATTA

Handl et al. [18] present an ant-based sorting algorithm that incorporates adaptive, heterogeneous ants and a time-dependent transporting activity. The algorithm they present is called ATTA which stands for Adaptive Time-dependent Transporter Ants, and it is "*...subjected to the most thorough analytical evaluation of an ant-based clustering and sorting algorithm undertaken to date*" [18].

The algorithm is separated into two main phases, an initialization phase and a main loop. In the initialization phase the objects are scattered randomly on a toroidal grid. Each of the agents then randomly picks up one of the objects before it is placed on a random location of the grid. This constitutes the initialization phase. In the main loop an ant is randomly selected and allowed to perform its actions. The ant first moves, and then decides whether to drop its data item or not. If the ant decides to drop its item, it will do this on the cell where it is currently located, or on an adjacent empty cell if the cell already contains an item. After the agent has dropped its item, it selects one of the items on the grid and jumps to its location. Based on the item's neighborhood, the ant decides whether to pick up the item or not. The ant will continue to jump to the locations of new items until it successfully picks up an item. Only after an ant successfully picks up an item is the main loop of the algorithm repeated and another ant selected to perform its actions. The main loop runs through a specified number of iterations.

The neighborhood function used to evaluate an item's neighborhood is a modified version of the function given in Equation 2.4 that was used by Lumer and Faieta [24]. The modified function is given below in Equation 2.7.

$$f^*(o_i) = \begin{cases} \frac{1}{s^2}\sum_j \left(1 - \frac{d(o_i,o_j)}{\alpha}\right) & \text{if } \forall o_j \left(1 - \frac{d(o_i,o_j)}{\alpha}\right) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

The constraint $\left(1 - \frac{d(o_i,o_j)}{\alpha}\right) > 0$ causes a heavy penalization of high dissimilarities which improves the spatial separation between clusters.

In addition to using a different density function than Lumer and Faieta [24], Handl et al. [18] also use different functions for calculating the probabilities for whether an ant should pick up or put down an item. The probability for picking up an item is given by Equation 2.8

$$P_p^*(o_i) = \begin{cases} 1.0 & \text{if } f^*(o_i) \leq 1.0 \\ \frac{1}{f^*(o_i)^2} & \text{otherwise} \end{cases} \tag{2.8}$$

and the probability for putting down an item is given by Equation 2.9.

$$P_d^*(o_i) = \begin{cases} 1.0 & \text{if } f^*(o_i) \geq 1.0 \\ f^*(o_i)^4 & \text{otherwise} \end{cases} \tag{2.9}$$

There is one additional factor that influences the picking up and dropping probabilities through the neighborhood function. Handl et al. [18] use a radius of perception of the ants that gradually increases over time. This means that in the beginning the ants only look at the immediate neighborhood when calculating probabilities for the picking up and dropping of items. Then, as time goes by, the ants consider a neighborhood of increasing size. This accelerates the dissolution of preliminary small clusters later in the run of the algorithm, while at the same time saving computations at the beginning of a run of the algorithm. The modification of the considered neighborhood influences the attainable values of the neighborhood function and thereby the picking up and dropping actions of the ants. At the beginning of the run of the algorithm the picking action is deterministic, and it is at this stage only the dropping action that favors dense and similar neighborhoods of items. As the run continues, greater values are attainable from the neighborhood function giving an additional bias towards the picking of misplaced data items. According to Handl et al. [18] this has the effect of decreasing the impact of density for the dropping action while at the same time increasing it for the picking action. This causes an improved spatial separation between the clusters formed by the algorithm.

In addition to the neighborhood function, the probability functions, and the increasing neighborhood considered, there are two more important characteristics of Handl et al's [18] algorithm. First of all, it extends Lumer and Faieta's [24] idea of a short-term memory in the agents. Each agent has a memory of a specified size that contains the locations of where it has previously dropped items. When an ant carries an object it evaluates each of the positions in its memory using the neighborhood function. The position of 'best match' is the location that gives the highest score for the neighborhood function. After investigating its memory, the ant jumps to the best location with a given probability. If the jump is not made, the ant's memory is deactivated until it successfully drops its currently carried object. The second important characteristic is that there is a short period during the algorithm where the neighborhood function is altered. After the algorithm has run for a period creating clusters, the scaling parameter, $\frac{1}{s^2}$, is replaced by N_{occ} in the neighborhood function. N_{occ} is the actual observed number of occupied grid cells within the neighborhood, and thus only similarity, not density, is taken into account in the neighborhood function. This has the effect of spreading out the data items in a sorted fashion. After the items have been spread out in sorted regions, the algorithm reverts to using the original neighborhood function.

The performance of the ATTA algorithm was compared to the performance of K-means, average link agglomerative clustering, and 1D-SOM[1] on several real and

[1]One-dimensional self-organizing map

synthetic data sets. In their comparisons, Handl et al. [18] found that the ATTA algorithm was the only one to perform consistently well on all the synthetic data sets. However, the results on the real data sets showed that the algorithm had problems if cluster structures existed on several levels. If this was the case, the algorithm only identified the cluster structures on the upper level.

2.2.4 Robotic Patch Sorting

The results of the sorting of objects by a group of homogeneous robots has been reported by Melhuish et al. [27] and Holland and Melhuish [21]. Their system consists of a homogeneous group of robots that are able to segregate two types of objects that differ only in their color. This is done using only a limited rule set and with no capacity for spatial organization or memory. The system is an extension of Beckers et al's [2] technique presented in section 2.1.3.

The robots used in the study are equipped with four infrared proximity sensors, with three facing forward and one facing backward. Instead of the pucks used as clustering objects by Beckers et al. [2], Melhuish et al. [27] use Frisbees. To enable the robots to move these Frisbees they are equipped with a gripper designed to sense, grip, retain, and release the Frisbees. When a robot moves directly toward a Frisbee, it will fit neatly inside the gripper, and will be retained in the gripper when the robot turns with the aid of two small weighted 'barbells'. When a Frisbee is inside the gripper, its type is detected by an optical sensor that senses the Frisbee's color. In order for the Frisbees to be retained in the gripper when the robot reverses, a small pin is located at the rear of the gripper that may be lowered inside the concave rim of the Frisbee. When the pin is lowered the Frisbee will be retained when the robot reverses. In addition to this, the gripper is equipped with a microswitch that is triggered if the robot pushes two or more Frisbees.

In the first of their experiments, Melhuish et al. [27] reproduce the clustering behavior observed by Beckers et al. [2]. The behavior of the robots is determined by the rule set given in Algorithm 2.1. Six robots using these three rules are then

---

Algorithm 2.1: Clustering algorithm

Rule 1:
if *gripper pressed and object ahead* **then**
  make random turn away from object
end

Rule 2:
if *gripper pressed and no object ahead* **then**
  reverse small distance
  make random turn left or right
end

Rule 3:
  go forward

---

given the task of clustering 22 Frisbees in an octagonal area with sides of two meters. When the experiment is run the robots first aggregate the Frisbees in small clusters, then form larger clusters, before finally a cluster containing 20 Frisbees is formed after 1 hour and 50 minutes.

To investigate if the slow progress of the previous experiment was due to the stability of the clusters, Melhuish et al. [27] modified the behavioral algorithm of the robots such that the Frisbees were not deposited hard up against each other. In the experiment they use both red and yellow Frisbees, but the yellow Frisbees are pulled back a certain distance before being released when the robot hits another Frisbee, or a cluster of Frisbees. The rules of the pullback algorithm are given in Algorithm 2.2.

Algorithm 2.2: Pullback algorithm

Rule 1:
if *gripper pressed and object ahead* **then**
    make random turn away from object
end

Rule 2:
if *gripper pressed and no object ahead* **then**
    if *yellow puck carried* **then**
        lower pin and reverse pull-back distance
        raise pin
    end
    reverse small distance
    make random turn left or right
end

Rule 3:
    go forward

The modification of rule 2 causes red Frisbees to be deposited right next to encountered Frisbees, whereas yellow Frisbees are deposited some distance away from the encountered Frisbees. If a robot that is pushing a red Frisbee collides with one or more Frisbees of any color, it will drop its pushed Frisbee immediately by reversing a small distance. However, robots that push yellow Frisbees will lower a pin inside the rim of the pushed Frisbee before reversing, thereby retaining the Frisbee when reversing. After reversing a set distance the robot raises the pin before reversing some more, causing the previously pushed Frisbee to be deposited. This causes yellow Frisbees to be deposited some distance away from the encountered Frisbees, whereas the red Frisbees are deposited hard up against the encountered Frisbees.

In the experiment that was performed using the pullback algorithm, 22 Frisbees of each type were used. After 7 hours and 35 minutes, 20 of the 22 Frisbees were located in a single cluster. The positions of the Frisbees in the arena at this point was described as follows: "*There is a central dense core of 16 reds, with 11 yellows and the other reds being packed around this core, and the other yellows scattered more loosely nearby*" [27]. According to Melhuish et al. [27] this outcome can be regarded as sorting of some kind, and it resembles a type of annular sorting.

2.3 Annular Sorting

There have been few reports of successful attempts to perform annular sorting by swarms of agents. One of the few successful attempts has been reported by Wilson et al. [33]. Their work is inspired by the brood-sorting behavior of the *Leptothorax* ant described by Franks and Sendova-Franks [15](see section 1.1.4). Based on three different hypotheses about the behavior of the *Leptothorax* ant, they propose three different mechanisms for producing an annular structure. The first mechanism uses a simple clustering algorithm on objects of different size. The second mechanism extends previous work [21, 27, 28], and the third uses a concept termed a 'combined leaky integrator'.

In order to compare the annular structures produced by the three different mechanisms, Wilson et al. [33] start by defining a performance metric. This metric consists of four components, with each component comparing a different characteristic of an annular structure with the same characteristic in an ideal structure. The ideal annular structure is described as "*... a compact circular cluster of one type of object, surrounded by perfect circular bands; each band formed from a different type*" [33]. After describing this ideal structure they postulate that an annular sort is a structure defined by four components: Separation, compactness, shape, and completeness. The proposed performance metric combines measures of each of these components, with each component contributing a maximum score of 100%.

The first component is the separation component, which tests every object of the structure to check if its position infringes on the 'home zone' of another object type. In order to do this, the radial distance from the center of the structure[2] for each object is measured, and then the lower and upper quartiles of distances for each object type are computed. After this, three different counts of objects are performed. First, the number of central type objects that have a distance to the center greater than the lower quartile of any other object type is counted, N_c. This rewards the positioning of central type objects closer to the center of the structure than the other object types. The second count that is performed concerns the outermost objects. Here the number of outermost objects that have a distance to the center lower than the upper quartile of any other type is counted, N_o. Outermost objects that are located closer to the center of the structure than other object types are thus penalized.

The two remaining counts are concerned with intermediate type objects. In their article, Wilson et al. [33] have given a wrong description of these counts [Sendova-Franks, personal communication] and the counts described here therefore differ from the ones originally reported. The first count for the intermediate type objects counts the number of objects which have a radial distance from the center greater than the lower quartile range of any object type further from the center, N_i^g. The second count for intermediate type objects counts the number of objects which have a radial distance from the center less than the upper quartile of any other object type closer to the center, N_i^l. These counts penalize intermediate objects that are either located closer to the center than objects belonging to a band of objects nearer the center of the structure, or located further from the center of the structure than objects belonging to bands further from the center of the structure. These four counts are combined

[2]The center of the structure is taken to be at the centroid of the largest cluster of the central object type.

as shown in Equation 2.10 to give the score of the separation component (n_o is the number of objects in the structure).

$$Se = 100 \times \left(1 - \frac{N_c + N_o + \frac{N_i^g + N_i^l}{2}}{n_o}\right) \tag{2.10}$$

The second component of the performance metric is a measure of the shape of the structure. This measure consists of two parts. The first part is the fraction of the central type objects located in the center of the annular structure, f_c. Each of the non-central type objects should form circular bands around the central cluster with each object located at approximately the same distance from the center as the other objects of the same type. A measure of this constitutes the second part of the shape component. To calculate this part the average Euclidean distance from the center of the structure for each non-central type is calculated, $\overline{o}_c$, and then the deviations from this average distance for the object type is summed up, d_c. This summed deviation is then normalized. The equation for the shape component is given in Equation 2.11, where m is the types of different objects.

$$Sh = \frac{100 \times f_c + \sum_{c=2}^{m} \left[100 \times \left(1 - \frac{d_c}{\overline{o}_c}\right)\right]}{m} \tag{2.11}$$

In an annular structure the objects in each band surrounding the central cluster should be spread out evenly. The completeness component of the metric is a measure of this spreading. To calculate this component each object type is considered in turn. For each object type a perfect angle between two objects of the type is calculated by dividing 2π radians by the number of objects of this type. Then the average deviation from this perfect angle for each pair of neighboring objects of the current type is calculated, $\overline{D}_c$. The completeness component is the mean of the average deviations for all the objects, and the equation for the component is given in Equation 2.12.

$$Cpl = 100 \times \frac{\sum_{c=1}^{m} \overline{D}_c}{m} \tag{2.12}$$

The final component of the performance metric is the compactness component, Cmp. This component measures how tightly the circular pucks are packed in the annular structure. This is done by comparing the average radial distance of the objects from the center of the structure, and the average distances of a perfect structure calculated with Donovan's algorithm and a poor structure. The four components, Se, Sh, Cpl, and Cmp, are then summed up to give the performance metric of the annular structure.

Having defined the performance metric, Wilson et al. [33] present the results of using three different mechanisms to create annular structures. The first experiment is performed in simulation only, and investigates whether a simple clustering algorithm (see Algorithm 2.1) applied to objects of different size may lead to the formation of an annular structure. In the experiment, pucks of four different sizes are used, and there are 15 pucks of each size. The first type of pucks is of a standard size, and the second, third and fourth types of pucks are two, three, and four times the size of the standard puck. After 500000 iterations the simulation is ended. The results of the experiment show that there is a slight improvement in the formation of an annular structure when the simple clustering algorithm is employed on objects of different size rather than on identical-sized objects.

The second mechanism investigated by Wilson et al. [33] is an extension of the pullback algorithm proposed by Melhuish et al. [27](see Algorithm 2.2). In the extended algorithm, different pullback distances are used for different types of objects. The puck types that should end up nearer the periphery of the annular structure are subjected to greater pullback distances than those types that should end up near the center of the structure. The varying pullback distances means that objects of the central type are deposited immediately on contact with other pucks, whereas the other types are pulled back a certain distance before being deposited depending on the type of the object. The differential pullback algorithm is given in Algorithm 2.3.

Algorithm 2.3: Differential pullback algorithm

Rule 1:
if *gripper pressed and obstacle ahead* **then**
    make random turn away from object
end

Rule 2:
if *gripper pressed and no obstacle ahead* **then**
    if *Type 1 puck carried* **then**
        drop puck
    else
        reverse distance according to puck type
        drop puck
    end
    reverse small distance
    make random turn left or right
end

Rule 3:
    go forward

The extended pullback mechanism was first investigated in a series of simulation experiments. These experiments showed, according to Wilson et al. [33], a reasonably good separation for two and three different object types, with a reduced separation for four and five different types. When there were a greater number of types, the outermost object types were less well separated than the central object types. After the simulation experiments, the mechanism was investigated in a real robotic experiment with three types of pucks. The result of this experiment was comparable to those from the simulation experiments.

The third mechanism investigated by Wilson et al. [33] is similar to the differential pullback algorithm, but with the important distinction that the pullback distances for the different object types are adaptive. In order to make the pullback distance for each object type adaptive, they introduce the concept of a 'leaky integrator', which provides information about the dispersal of each object type. There is one integrator in the form of a counter for each object type. Whenever a puck of the type associated with the integrator is dropped by the robot the counter is incremented. When the robots are about to drop an object, the pullback distance is calculated by summing up the values of the integrators that correspond to objects that are to be positioned closer to the center of the structure than the currently moved object. The integrators are 'leaky' such that their values are reduced over time if they are not constantly updated by continuing dropping operations. The leaky integrator algorithm is presented in Algorithm 2.4.

Algorithm 2.4: Leaky integrator algorithm

```
Rule 1:
if gripper pressed and obstacle ahead then
    make random turn away from object
end
Rule 2:
if gripper pressed and no obstacle ahead then
    if Type 1 puck carried then
        add 15 units to Type 1 integrator
        drop puck
    end
    if puck type f carried then
        add 15 units to type f integrator
        reverse distance proportional to the sum of integrators 1 to f
        drop puck
    end
    reverse small distance
    make random turn left or right
end
Rule 3:
if gripper not pressed then
    go forward
end
Rule 4:
if time counter reaches threshold then
    deduct 1 unit from all integrators and reset time counter
end
```

The first simulation experiments performed with the leaky integrator algorithm showed no improvement over the results obtained using the differential pullback algorithm. However, the algorithm requires several parameters to be set and problems with finding the correct values for these were the cause of the lack of good results for the algorithm. In an attempt to find more optimal parameters for the algorithm, Wilson et al. [33] employed a genetic algorithm. When the obtained parameter values were tested experimentally, an improvement was recorded in both separation and compactness in the resulting structure. Real robotic experiments using the same parameter values achieved comparable results.

2.4 Summary of Related Work

In this chapter we have described previous work on the creation of swarms of agents that cluster and sort objects. The work includes both models that have been implemented in computer simulations, and work on physical robots. All of the work share some key characteristics both in the developed models and in the methodology used to create the models. The researchers have all started with a set of hypotheses about the individual behavior of the agents that lead to the emergent patterns at the global level of the system. Based on these hypotheses they create a set of rules that govern the behavior of the individual agents. These rules all concern the behavior related to pick up/push and deposition/release of objects. The antecedents of these rules are either a set of conditions that must be true for the rule to be applied, or a calculation of a probability that decides whether the consequent of the rule will be executed or not. The information used by the agents when deciding which rule to invoke is limited, and only in Lumer and Faieta's model [24] and ATTA [18] (see sections 2.2.2 and 2.2.3) do the agents inspect all of their surrounding cells before taking action. All of the previous research has consequently used a set of rules to control the agents. The use of other controllers, for example a neural network, has previously not been investigated. The movement of the individual agents have in all of the previous research been random. In the computer simulation models the agents either pick a random neighboring cell to move to, or they pick a random direction and length of a walk. None of the agents in previous research consequently use information about the local configuration of objects to modify their movement.

The methodology used to create the set of rules controlling the agents is the same for all of the previous research. The basis is a set of hypotheses about the characteristics of the individual behavior that cause the emergence of patterns at the global level of the system. These hypotheses are then captured in a set of rules controlling the behavior of the individual agents. When this is done the agents are run either in simulation or in real-life. If the expected patterns are not observed, the rules are modified and the agents are run again. This process is repeated until the desired patterns emerge. The methodology used in previous research is thus a 'specify–test' approach where the behavior of the individual agents is first hand-coded, and then tweaked until the desired patterns emerge. None of the previous research has used a genetic algorithm to evolve the individual behavior instead of hand-coding it. The only use of a genetic algorithm so far has been to find the optimal parameters for one of the models [33].

Chapter 3

Method

The task to be achieved by the designed system is the evolution of a swarm of agents, called ants, which are capable of clustering and sorting objects. The structure of the system is inspired by social insects and their solutions to the tasks of clustering, patch sorting, and annular sorting. In chapter 1 several social insect systems were described. In all of these systems there are agents that reside in a physical world. Two of the things that are needed from the system designed to solve the tasks are consequently a group, or swarm, of agents and an environment which they inhabit. There must also be collections of objects in the world that the ants can manipulate. There are thus three main components of the designed system. These are ants, objects, and a world in which these two components are located. An overall logical structure of the system containing these components and their relations is shown in Figure 3.1.

As can be seen in Figure 3.1, the top level construct is an environment which contains the other components. The environment has a swarm of ants and a collection of objects that both inhabit a world. The ants in the environment can perceive the world, that is, sense what is in the world, as well as carry the objects in the world. The system that is built is based on the Swarm libraries[1].

3.1 The Environment

The environment is the main structure that contains everything needed for a solution to the given tasks. As can be seen in Figure 3.1 the environment contains three logical components. The first is a collection of objects that are to be moved around by the second component of the environment, which is a swarm of ants. The third component of the environment is a world which contains the objects and ants. This world is structured as a grid with periodic boundary conditions. The periodic boundary conditions of the world mean that an ant that moves out of the world on one side will enter the world on the opposite side. Thus, the two opposing sides of the world are located 'next' to each other in a logical sense. The cells of the world may either contain one object, one ant, or one ant and one object. This means that objects cannot be stacked on top of each other, and that only one ant can be located at a specific location at any one time. Previous work within this field of research has taken

[1]Swarm is a set of libraries that facilitate the implementation of agent-based models, see www.swarm.org

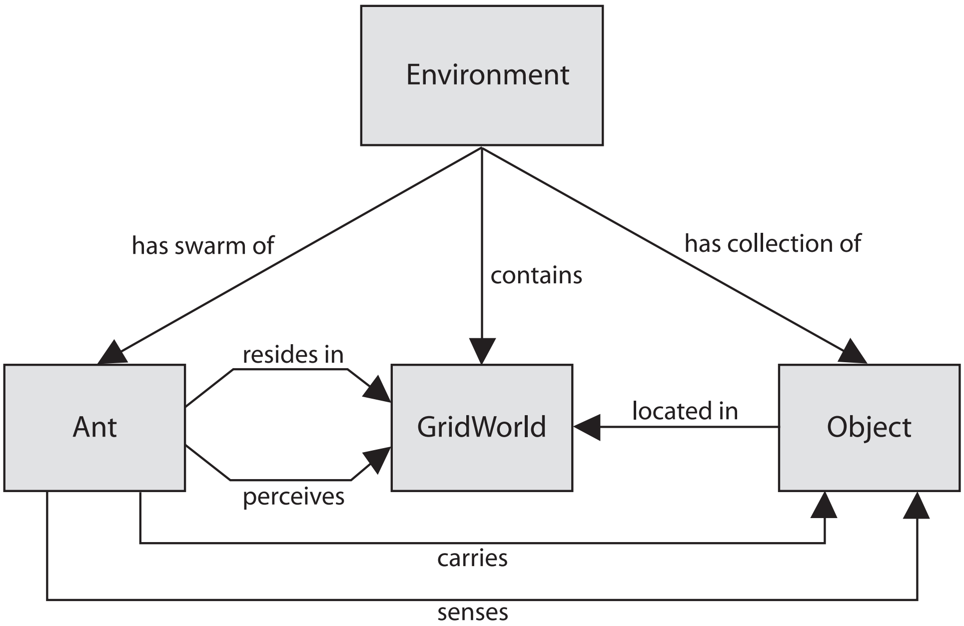

Figure 3.1: Logical view of system components.

different strategies concerning the point of whether ants should be able to move over objects. That is, the possibility of an ant and an object to be located in the same cell of the world. In the initial work by Deneubourg et al. [12] the ants were able to move over the objects, but later work by Martin et al. [25] and Monmarché et al. [22] removed this possibility. In our work we have chosen to allow the ants to move over the objects because we feel that this models the behavior of real ants more closely. This also requires less of the ants as they now only need to be able to pick up objects that are located in the same cell as themselves, and not objects located in any one of their eight surrounding cells[2]. The objects located in the world are elementary. They contain a location in the form of x and y coordinates as well as a value. The value is discrete and defines the type of the object such that the objects may be compared in terms of similarity.

In addition to containing objects and ants in a world, the environment has a notion of time. An environment goes through a series of discrete time steps that starts at the 'birth' of the environment. The environment consequently has an age that can determine how long it takes for a swarm of ants to solve a given task. At each time step, all the ants in the environment are stepped in random order and allowed to perform their actions.

The important characteristics of the environment can be summed up as follows. An environment contains a collection of objects and a swarm of ants located in a grid-world with periodic boundary conditions. At the beginning of the 'lifetime' of an environment, all the objects and ants are placed in random locations of the world. Then, when the 'age' of the environment increases by one unit, all the ants are allowed to perform their actions. This causes movement of the ants as well as the objects, mediated through the carrying by ants. As time goes by, the objects that were initially

[2] The added complexity is not an issue in computer simulations, but in a robotic implementation the added complexity may be prohibitory.

randomly dispersed in the world will be placed in patterns according to the desired solution of the problem. If the task for example is to sort the objects in different clusters according to type, the objects will be randomly distributed in the beginning, and as time goes by spatial regions in the world will tend to contain only one type of object.

An important characteristic of the environment is the relations between the size of the world and the number of ants and objects. The basic problem here can be phrased in terms of relating both the size of the world and the number of ants to the number of objects: Depending on the number of objects to be clustered or sorted, how many ants are needed to solve the task, and how large should the world be? There are two tradeoffs involved in this dilemma. The first concerns the number of objects compared to the size of the world. If the world is too small, then it may be impossible to solve the task. In the extreme case there may be one object in each cell of the world. If the task is to perform a patch sort, then it will not be possible to create separate clusters if every cell of the world is occupied. Even if a solution is possible despite the high density of objects, it may be difficult to achieve because the ants must search for a long time for a possible site to drop their currently carried object. On the other hand, if the world is too big, then a solution will be slow because the ants must move for a long time before finding possible objects to pick up, or finding possible sites to drop their currently carried object. We have chosen to use the tradeoff between the number of objects and the size of the world proposed by Handl et al. [18]. They use a square world where the length of each side is equal to $\sqrt{10 \times n_o}$, where n_o is the number of objects in the world[3].

The other tradeoff concerns the number of ants compared to the number of objects. If there are too few ants, then it will take a long time to solve the task simply because there are fewer workers to perform the task. On the other hand, if there are too many ants, then they may interfere with each others work and thus slow down the overall progress. Ramos and Merelo [30] have suggested that the ratio of the number of objects to the number of ants should be ten to one. In other words, the number of ants divided by the number of objects should be 0.1. The environment thus contains two parameters that are dependent on the number of objects. The first is the size of the square world were the sides are of length $\sqrt{10 \times n_o}$, and the second is the number of ants, n_a, where $n_a = 0.1 \times n_o$.

3.2 The Ant

The ant is the component of the environment that actually performs the actions. It is the ants that move around in the world and manipulate objects, thereby solving the given tasks. The ants have very limited capabilites in order to keep them as simple as possible. They have limited sensory capabilities and the unitary actions they are capable of can be split in two groups, the handling of objects and movement. The ants are capable of picking up objects, carrying objects, and putting objects down in addition to basic movement abilities.

To aid the ants with their actions they are equipped with limited sensory abilities. In a grid-based world every cell has eight neighbors. Since an ant occupies a single

[3]The resulting number is rounded upwards to the nearest integer.

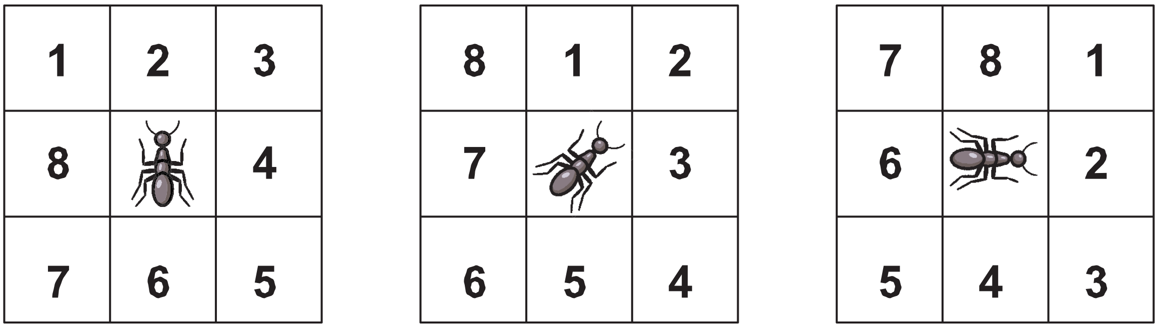

Figure 3.2: Sensory input numbers of the ant's surrounding cells when having the directions north, northeast, and east.

cell at any one time, it always has eight surrounding cells which define its eight-cell neighborhood. The ants are only capable of perceiving the objects contained in this neighborhood, in addition to the object that is currently being carried. For each cell in the neighborhood the ant is able to determine if there is an object located in the cell, and the value of any such object. The same is the case for the object being carried. The ant can sense if it is carrying an object, and if this is the case, the value of the carried object. Every ant's sensory apparatus is thus very limited since it can only perceive objects in a limited spatial area (objects in its immediate neighborhood) and it is unable to sense other ants at all.

Since the experiments are performed in computer simulations and not in the real world, noise is added to the perceptions to simulate the non-determinism of real-life behavior. In one percent of the perceptions the actual perception is replaced by one of the other possible perceptions. For example, if an ant senses that there is an object in a specific cell in its neighborhood, then the perception of this object may be replaced by the perception of one of the other possible objects in the environment, or with the perception of no object. Since no information about other ants is available to the ants, they can be seen as operating independently of each other, although they do affect the same world with their actions.

The ants all have a direction giving their current orientation in the world. This direction can take on eight values: north, northeast, east, southeast, south, southwest, west, and northeast. The direction of an ant affects both its movement and its perception. The direction of the ant is taken into account when the ant perceives its neighborhood such that the cell directly in front of the ant always constitutes sensory input number two to the ant. The sensory input number of the eight cells in the ant's neighborhood when the ant has the directions north, northeast, and east are shown in Figure 3.2.

Since the ants are capable of perceiving their eight surrounding cells as well as the object they are currently carrying, the ants have a total of nine perceptual inputs. These inputs are given to the ant's controller to create a control sequence that determines the actions of the ant. The controller may be any type of construct as long as it converts a nine object input into a six digit control sequence. In our ants we use a neural network controller which is further described in section 3.3. When the sensory input is given to the controller it produces a six-digit binary sequency. Each of these digits controls a unitary action that may be performed by the ant. The first

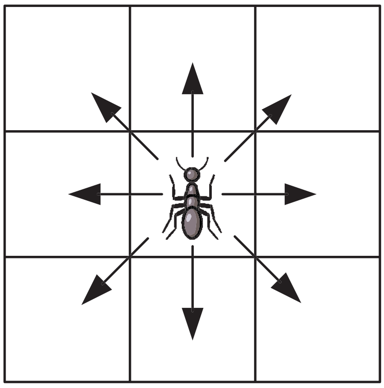

Figure 3.3: Possible movement directions of the ant.

digit of the sequence controls the picking up behavior of the ant. If the ant is not currently carrying an object and the first digit of the sequence is a 1, then the ant will attempt to pick up an object. The ant may only pick up an object from the cell where it is currently located. There are thus three conditions that must be satisfied in order for an ant to pick up an object. First, the ant must not already be carrying an object. Second, the control sequence must indicate that the ant should attempt to pick up an object. Third, the ant will only pick up an object if there is an object located in the ant's current position. If the first digit of the sequence is a 0, the ant will continue to the next digit of the control sequence.

The second digit of the control sequence handles the deposition behavior of the ant. If the ant is carrying an object, then it may attempt to drop this object at its current position. Since each cell in the world may only contain one object, the ant cannot drop an object if the cell at its current position already contains an object. As with the picking up action of the ant, there are three conditions that must be satisfied in order for an ant to put down an object. First, the ant must be carrying an object. Second, the second digit of the control sequence must be a 1, and third, the cell at the ant's current position must be empty. If any one of these conditions is not satisfied then the ant will not be able to put down an object at its current position.

As described above, the first two actions defined by the control sequence has to do with the manipulation of objects. The remaining four digits of the control sequence all influence the movement of the ant. The ant is only capable of moving one cell at a time, but it may move to any one of its eight surrounding cells. The possible moves of the ant are shown in Figure 3.3. Four basic movement capabilities enable the ant to move in these eight directions. The first two are movement forwards and backwards. Forward movement is controlled by the third digit of the control sequence. If this digit is a 1 then the ant will move one cell forward. However, any one cell of the world cannot contain more than one ant, and the ant may therefore not be capable of moving if the cell it wishes to move to is already occupied by another ant. If this is the case, then the ant will remain stationary until it again is given the chance to move. The fourth digit of the control sequence controls the backward movement of the ant. If this digit is a 1 then the ant will move one cell backward. As with the forward movement, the ant will not move if the cell behind it is already occupied by another ant. One thing to note with the forward and backward movement is that they may cancel each other out if they are both active at the same time, thereby leaving the ant stationary.

If the ants were only capable of moving forwards and backwards, they would only be able to move to two of the cells in their neighborhood. In order to add the ability of moving to any one cell of their neighborhood, the ants are equipped with the actions of turning left and right. If the fifth digit of the control sequence is a 1 the ant will turn left, and if the sixth digit of the sequence is a 1 then the ant will move to the right. As with the forwards and backwards movement, the left and right turning may cancel each other out if they are both active at any one time. The combination of the turning behavior with the forward and backward movement allows the ants to move to any of their eight surrounding cells. A combination of forward movement and left or right turning will, for example, allow the ant to move diagonally with respect to its current direction. Since the ants are not capable of sensing other ants, deadlock situations of ants being unable to move may occur. This could happen if two or more ants wish to move to each others positions. To prevent these deadlocks from continuing for an infinite period of time, a counter is added to the ants that counts the number of steps for which the ant has remained stationary. If an ant remains stationary for ten consecutive steps, or 'moves', the ant will move to a random position in the world that is not occupied by another ant.

In the previous section describing the environment the notion of time steps was introduced. At each time step of the environment, all the ants are stepped in random order. The behavior of the ant when stepped is summarized in Algorithm 3.1. When an ant is stepped it first perceives its eight cell neighborhood, as well as its possibly carried object. These perceptions are then sent to the ant's controller which returns a six digit control sequence. The ant will then first attempt to pick up an object if it is not carrying an object and the control sequence instructs it to do so. If the ant is already carrying an object, then it will attempt to drop this object if this is indicated by the control sequence. After the picking up or dropping behavior is completed, the ant will turn left or right before moving forwards or backwards. The movement behavior concludes the ant's actions when stepped by the environment.

3.3 The Controller

The controller used by the ants to decide on which actions to perform depending on their perceptions is a feedforward neural network. Feedforward networks are a subclass of acyclic networks, and connections from nodes in one layer are only allowed to nodes in the next higher level [26]. The topology of the feedforward networks used can be seen in Figure 3.4. The networks all have nine groups of inputs and six output nodes. The number of nodes in the hidden layer varies because the number is dependent on the number of different objects that may be encountered by the ants.

The network has nine inputs. These nine inputs correspond to the nine different perceptions that the ants may perform. There are eight inputs from the surrounding cells of the ant, as well as one input from the possible perception of the object that the ant is currently carrying. Figure 3.4 shows the mapping from the ant's perceptions to the input nodes of the network. The nodes 1–8 correspond to perceptions from the ant's eight surrounding cells, and the node A is the perception of the object that the ant is carrying. In the experiments where there is only one type of object that may be encountered by the ants, there is only one input node for each perceptual input. When there is only one type of object in the environment, then each of the

Algorithm 3.1: Ant behavior

```
perceive neighborhood;
get control sequence Ctrl;
if not carrying object then
    if Ctrl = pickup and object at current location then
        pick up object;
    end
else
    if Ctrl = drop and no object at current location then
        drop carried object;
    end
end
if Ctrl = turn left then
    turn left;
end
if Ctrl = turn right then
    turn right;
end
if Ctrl = forward then
    move forward;
end
if Ctrl = backward then
    move backward;
end
if no movement possible then
    increment immobility counter;
    if immobility counter > maximum immobility then
        move to random location;
        reset immobility counter;
    end
end
```

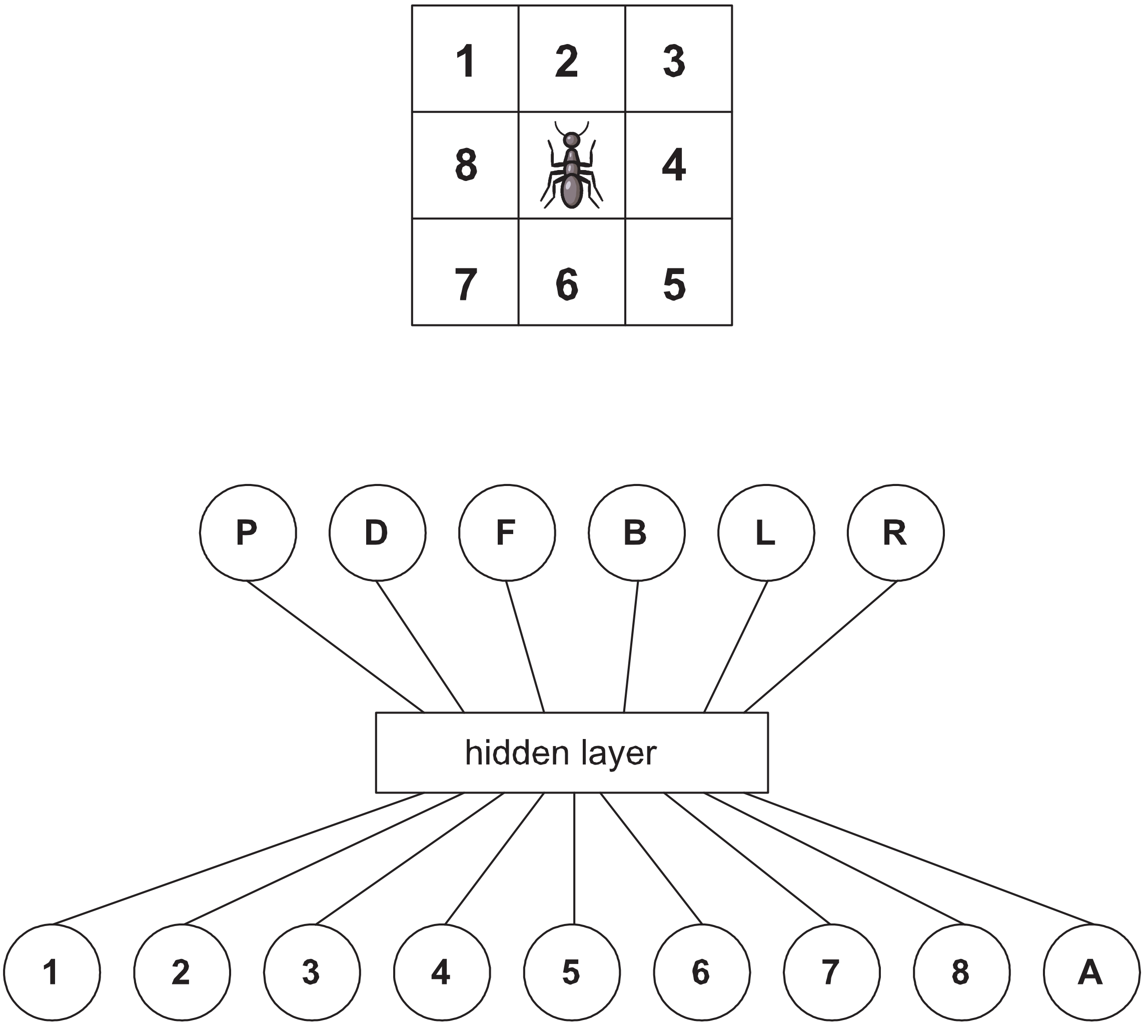

Figure 3.4: Topology of the feedforward network and the mapping of the ant's perceptions to the inputs of the network. The input to node A is a perception of the object that the ant is currently carrying.

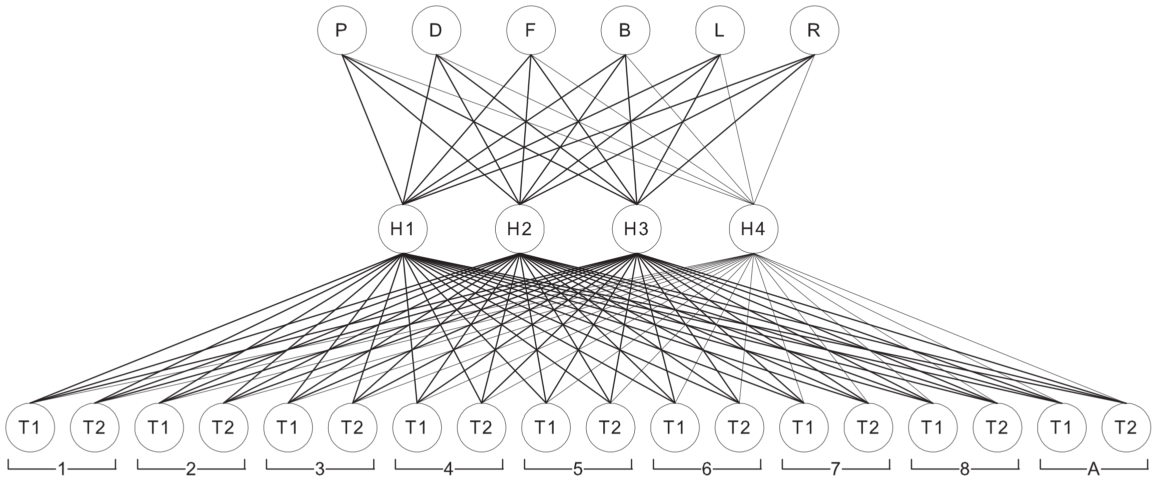

Figure 3.5: The feedforward network used for the task of patch sorting with two different object types. The network has three hidden nodes and nine groups of input nodes with two nodes in each group.

ant's neighboring cells may either contain an object or be empty. To make a correct perception of what is in each of these cells it is thus only necessary with two different stimuli, one for the existence of an object and one for an empty cell. This binary sensory stimuli can be represented in the input nodes with the nodes either firing or not. In the networks used in the experiments, an input node fires if an object is sensed by the ant in the corresponding sensory field.

When there is more than one type of object, a binary sensory stimulus is no longer sufficient. For each of the perceptual inputs the ants must be capable of differentiating between the different types of objects, in addition to the case were there is no object. This means that the inputs to each node in the input layer are no longer binary, and so the different sensory stimuli can no longer be represented by the input nodes being on or off. There are two different solutions to this problem. The first is to represent the different sensory stimuli with different firing activations of the node. In this case each input node is not simply firing or not, but there is a scale of different firing levels that represent the different sensory stimuli. The approach taken in this research is the alternative solution. Instead of having one input node for each perception, there are groups of input nodes for each perception. The networks have nine input groups, with the same number of input nodes in each group as there are object types. If, for example, an object of type A is located directly in front of the ant, then only the node corresponding to a type A object in the second input group will fire. Consequently it is only possible for one node in each input group to fire at any one time, because only one object may be perceived in any one of the ant's perceptual fields. A network used for patch sorting when there are two different types of objects is shown in Figure 3.5. This network has, like every other network, nine input groups. Each of these groups has two nodes corresponding to the two types of objects that may be perceived. In addition there are four nodes in the hidden layer.

All the networks used have six output nodes. The outputs of these nodes are used to create the six digit control sequence given to the ants. Each of the output nodes has a binary output and the control sequence will thus consist of six binary digits.

The first two outputs control the pick up and deposit behaviors of the ant (nodes P and D in the figures). The remaining four outputs control the movement behavior of the ant. Output nodes three and four control the forward and backward movement (F and B in the figures), and the remaining two nodes control the left and right turning behavior (L and R in the figures).

The hidden nodes and the output nodes of the network use similar activation functions. The hidden nodes of the network use the sigmoid function given in Equation 3.1.

$$f(net) = \frac{1}{1 + \exp^{-net}} \tag{3.1}$$

In the equation, *net* is the sum of the inputs to the node multiplied by their weights. The computation of *net* is shown in the equation below, where x_i is the output of the node's ith input node and w_i is its corresponding weight.

$$net = \sum_i x_i \times w_i \tag{3.2}$$

The activation function of the output nodes is very similar to the activation function of the hidden nodes. The only difference between the two functions is that in the output nodes a step function is applied to the output of the sigmoid function. This means that whereas the output of the hidden nodes are values in the interval $[0 \ldots 1]$, the output of the output nodes are discrete valued and may only take on the values of 0 or 1. The activation function of the output nodes is shown in Equation 3.3.

$$f(net) = \begin{cases} 1 & \text{if } \frac{1}{1+\exp^{-net}} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

3.4 Evolution

In order to find solutions to the tasks of clustering, patch sorting and annular sorting, a genetic algorithm is used. A genetic algorithm maintains a population of evolving competitive problem solutions [23]. There is thus a space of candidate solutions which the genetic algorithm searches. However, not every possible solution is created and evaluated. The genetic algorithm finds solutions by examining only a small fraction of the possible candidate solutions [29]. A general version of the genetic algorithm used in this work is shown in Figure 3.6.

The genetic algorithm used consists of five major steps that are repeated until a good enough solution is found, or until the algorithm has gone through a specific number of repetitions. The first step in the algorithm is the initialization of the next generation. If this is the first cycle of the algorithm, then every individual in the generation must first be created before they are initialized. After every individual in the current generation has been initialized, the next step is to run each individual in the generation. When an individual is run it will create a proposed solution to the desired task. If for example the task is to cluster the objects, then each individual will attempt to form a cluster of the objects contained in its grid-world. After each individual has run for a specified time, the next step in the genetic algorithm is to evaluate the proposed solutions. The evaluation of the solutions is used to create a mating pool consisting of the best individuals of the current generation. The individuals in this mating pool are then used as parents for the next generation by combining

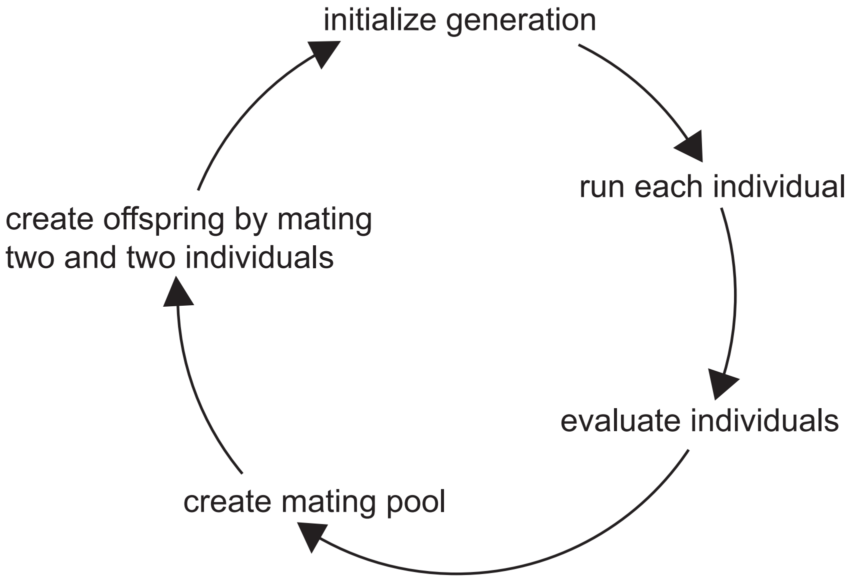

Figure 3.6: The genetic algorithm cycle.

pairs of the current individuals. This concludes the cycle of the genetic algorithm, and the new generation of possible solutions is initialized in the beginning of another cycle of the algorithm. For more details on genetic algorithms, see [16, 20, 29].

3.4.1 Phenotype and Genotype

The individuals of the generations in the genetic algorithm are the environments described in section 3.1. Each individual thus contains a swarm of agents and a collection of objects that are located in a grid-world. When the individuals of a generation are initialized a random positioning of ants and objects is defined. Then each environment is set up such that they all have this same random positioning of objects and ants. When each individual environment in the current generation is run, the environment is stepped a specified number of steps. This causes the ants in the environment to perform pick up and deposition actions, as well as move around in the world. An individual in the genetic algorithm can thus be said to be a swarm of ants that are allowed to move around in a world and move objects for a specified number of steps. The actions of the swarm will cause the objects in the world to have different positions at the end of the run compared to the beginning of the run.

The positioning of the objects constitutes the phenotype of the individuals in the genetic algorithm. After the individuals of the current generation have completed running, they are evaluated in terms of the positioning of the objects in their world. The objects that are currently being carried by the ants are not considered part of the world, and are thus not part of the individual's phenotype. The evaluation is done by giving the positions of the different types of objects to a fitness function. Depending on the task to be solved, this fitness function will give different scores to the various spatial configurations of objects. If the task, for example, is to cluster identical objects, a dispersed configuration of objects will receive a lower score than a densely packed spatial configuration of objects.

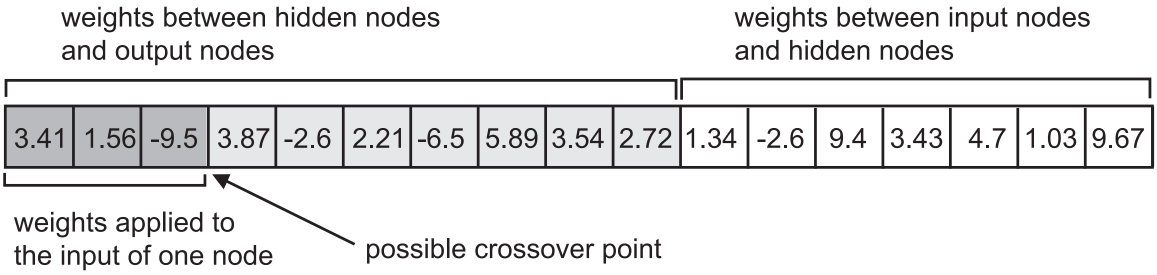

Figure 3.7: The genotype of an individual.

Whereas the phenotype of the individuals is the positioning of their objects, the genotype is what can be said to be responsible for this positioning at the lowest level. The positioning of the objects is formed by a swarm of agents, where each agent is controlled by an identical controller. It is the basis of this controller that is the genotype of an individual. The controllers used are feedforward networks, and the genotype of each individual is an array containing the weights of the networks. An example of this array of weights is shown in Figure 3.7. There are different levels of abstraction at which the array of weights may be seen. At the lowest level the array can be seen as consisting only of real-valued positive and negative numbers. At the next level of abstraction the weights are divided in groups, with each group defining the weights on the inputs to a specific node. The networks are all fully connected such that there are connections between all the input nodes and the hidden nodes, and all the hidden nodes and the output nodes in the network. Each group of weights thus consists of all the weights on the connections from the nodes in the next lower layer and the 'current' node. This level of abstraction views the array of weights as describing the different nodes in the hidden and output layers of the network. The next level of abstraction views the array as being divided in two parts. One part describing the weights on all the connections from the input layer to the hidden layer, and the other part describing the weights from the hidden layer to the output layer.

The array of weights in combination with the topology of the network describes the feedforward networks controlling the individual agents in the environment. There is an intricate path from the genotype to the phenotype of an individual, and this is sketched in Figure 3.8. The genotype of an individual is an array of weights which define a network. This network controls an individual ant which is part of a swarm of identical ants. The swarm of ants resides in a world containing objects, and the ants move these objects around. The actions of the swarm of ants cause a spatial configuration of the objects in the world which constitutes the phenotype of an individual.

3.4.2 Crossover and Mutation

Both the crossover and mutation operators work on the genotype of the individuals. The crossover operator is responsible for creating a new genotype from two existing genotypes. When two genotypes are crossed over the following takes place. The first thing that is done is to calculate the number of possible crossover points. To avoid the crossover operation from being too much of a disruptive force, crossover

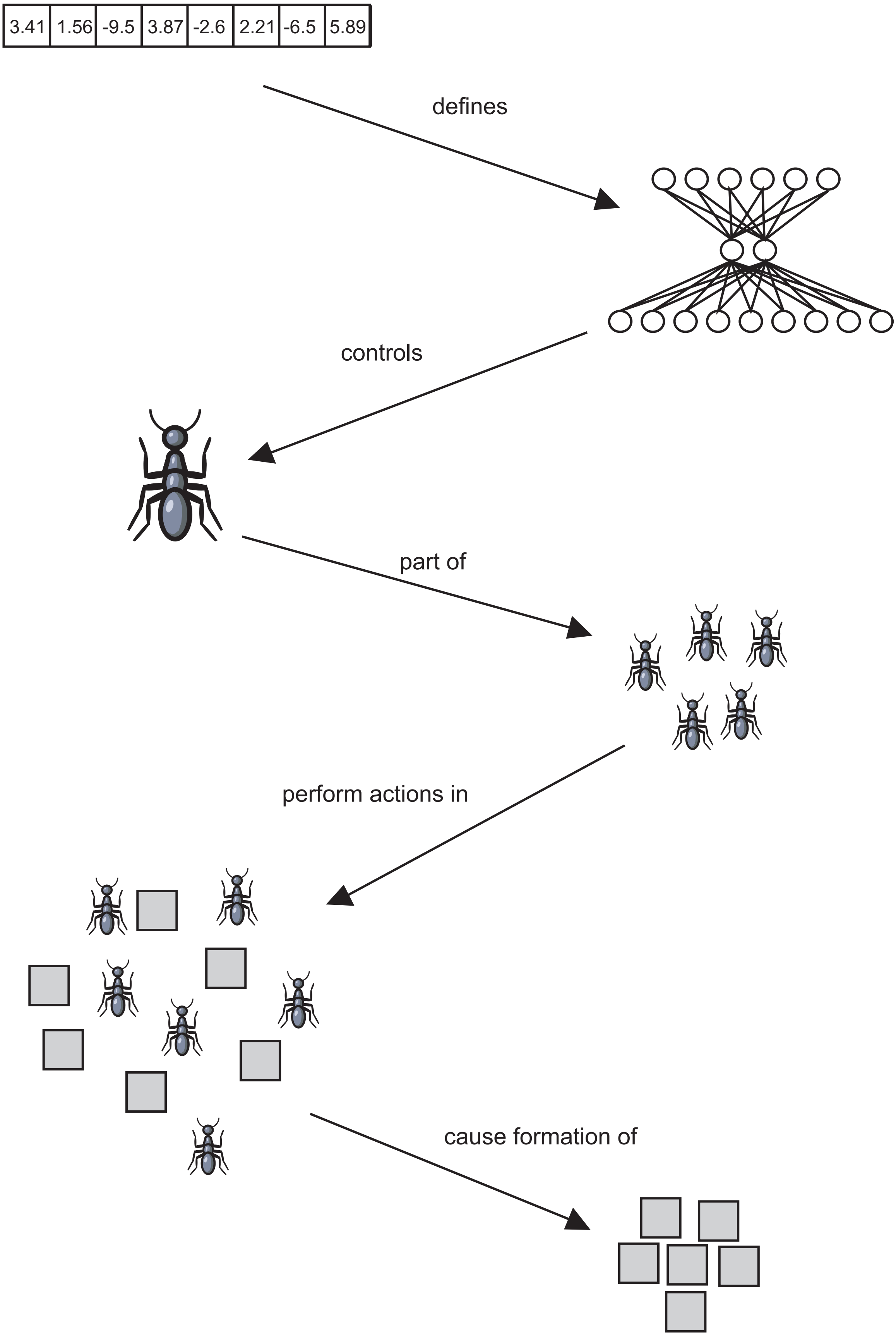

Figure 3.8: The path from an individual's genotype to its phenotype.

is not allowed to split the weights that are applied to the inputs of the same node in the network. Thus crossover can only occur between the groups of weights that belong to the same nodes (see Figure 3.7). After the number of possible crossover points has been determined, one of the points is selected at random. The groups of weights located before this point from the first parent genotype is then combined with the groups of weights located after this point in the second parent genotype. The crossover operator thus uses a one-point crossover with the first part of the resulting genotype coming from the first parent, and the second part coming from the second parent.

The mutation operator may affect every weight of the genotype. It goes through all the weights, and for each weight decides whether it should be altered or not. Each weight is changed with a certain probability, and if the weight is to be changed it is increased or decreased with a random number from the interval $[-1 \ldots 1]$.

3.4.3 Selection Mechanisms

The selection mechanism used in the genetic algorithm is a combination of elitisism and tournament selection. Elitisism forces the genetic algorithm to retain some number of the best individuals at each generation [29]. After the individuals of a generation have been evaluated, the five best individuals of that generation are copied into the next generation, only undergoing mutation in the process. This reduces the chance of good individuals not being selected for reproduction, or being destroyed by crossover.

Tournament selection is the second selection mechanism employed by the genetic algorithm. Five individuals are chosen at random from the current generation. These five individuals then compete for inclusion in the mating pool. The best individual of the five-individual tournament is selected for inclusion in the mating pool with a probability of 0.75. If this individual is not selected, then one of the five individuals (including the best) is selected at random. The number of tournaments that are performed is equal to the size of the current generation minus the number of individuals that were selected by the elitist mechanism.

After the mating pool is created from the tournament selections, pairs of individuals are selected randomly for mating. The two individuals have a certain probability of crossing over with each other. If they are to undergo crossover, then the two are first crossed over with the first parent being first, and then the second parent being first. This results in two offspring, one with its first half from the first parent and the second half from the other parent, and the other with the first part from its second parent and its last part from the first parent. If the two parents are not subjected to crossover they reproduce by cloning themselves. After the children have been created they undergo mutation. The mating process is repeated until the next generation contains the same number of individuals as the current generation, including the individuals that were subjected to elitisism. A summary of the genetic algorithm used in the experiments is given in Algorithm 3.2.

Algorithm 3.2: Genetic algorithm

```
create first generation of populationsize individuals;
for gen = 0 to runlength do
    /*initialize current generation */
    pick random positions for ants and objects;
    initialize each individual;
    /*run current generation */
    foreach individual do
        for i = 0 to runtime do
            randomize list of ants;
            step each ant in the list;
        end
    end
    /*create next generation */
    calculate fitness of all individuals;
    copy the elitistnum best individuals into the next generation;
    /*create mating pool */
    for j = 0 to populationsize - elitistnum do
        select five individuals at random;
        pick a random number p;
        if p < 0.75 then
            select the best individual;
        else
            select an individual at random;
        end
        put the selected individual in the mating pool;
    end
    /*make offspring */
    for k = 0 to (populationsize−elitistnum)/2 do
        select two individuals at random from the mating pool, ind1 and ind2;
        pick a random number r;
        if r < 0.7 then
            crossover the individuals with ind1 as first parent;
            crossover the individuals with ind2 as first parent;
        else
            clone individuals ind1 and ind2;
        end
    end
    expose all individuals of the next generation to mutation;
end
```

Parameter	**Value**	
Number of objects	50	60
Size of world	23×23	25×25
Number of ants	5	6
Individual runtime		4000
Population size		50
Generations		6000
Crossover rate		0.7
Mutation rate		0.05

Table 3.1: Parameter values used in the experiments.

3.5 Parameter Settings

There are several parameters that must be set for each experiment. The parameters can be divided into two classes: those that apply to each individual, and those that apply to the genetic algorithm. The first parameter that must be set for each individual is the number of objects that are to be located in the world. All of the experiments use either 50 or 60 objects. As described in section 3.1, the number of objects can be used to determine the size of the world and the number of ants to use. This means that an environment containing 50 objects will have a 23×23 grid world with 5 ants, and an environment containing 60 objects will have a 25×25 grid world with 6 ants. The last parameter that needs to be set for each environment is how many steps it should be allowed to run. In all the experiments each individual environment is run for 4000 steps.

The remaining parameters apply to the genetic algorithm. All experiments have a population size of 50 individuals, and the evolutionary run is 6000 generations long. The crossover rate is set to 0.7, which means that in 70% of the cases, the parents selected from the mating pool are crossed over to produce offspring. The mutation rate is constant and is set to 0.05 for all the experiments. The parameter settings are summarized in Table 3.1.

Chapter 4

Experiments

In our work we have designed experiments to solve the tasks of clustering, patch sorting, and annular sorting by a swarm of agents. This chapter describes five of these experiments. The first experiment solves the task of clustering. This is the problem that was first addressed within this field of research by Deneubourg et al. [12] in 1991. Clustering is the task of grouping a collection of identical objects in a continuous area. The second and third experiments are concerned with patch sorting. This task involves the grouping of two or more classes of objects such that each class is individually clustered, while at the same time the clusters of each class are separated from each other [27]. The second experiment solves the task of sorting two different types of objects in to distinct clusters, while the third experiment sorts three different types of objects. The last two experiments solve the task of annular sorting. Melhuish et al. [27] have defined annular sorting as "*...forming a cluster of one class of objects, and surrounding it with annular bands of the other classes, each band containing objects of only one type.*" In the fourth experiment an annular structure consisting of two types of objects is formed, and in the final experiment an annular structure containing three types of objects is created.

4.1 Clustering

In this experiment the task to be solved is the grouping of identical objects in a continuous area. To solve this task all the objects in the world must be placed together such that they form a structure that is as compact as possible. This means that there can not be any empty cells within the structure, and that no objects can be isolated. If the definition of the clustering task is viewed on a local scale, it can be rephrased as the requirement that each individual object must be surrounded by as many other objects as possible. If each object in the world has as many objects in its eight surrounding cells as possible, the resulting global structure will be a densely packed cluster of objects. This local view of a cluster is used in the fitness function of the experiment. To compute the fitness of a proposed solution, every object in the final configuration formed by the solution is investigated in turn. For each object the number of objects in its eight-cell neighborhood is counted. The sum of these counts constitutes the fitness of the solution. The fitness function is given in Equation 4.1.

$$\sum_{i=1}^{n_o} c(o_i) \tag{4.1}$$

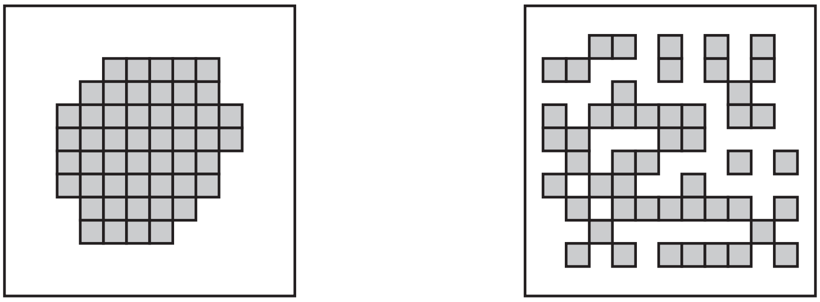

Figure 4.1: The left structure will receive a maximum score from the fitness function, whereas the structure on the right will receive a mediocre score.

In the equation n_o is the number of objects in the world, and $c(o_i)$ is a counting function that returns the number of objects in object o_i's eight-cell neighborhood. Figure 4.1 shows two different structures that will get a maximum and a mediocre score by the fitness function.

The controller of the individual ants in the experiment is a feedforward network with nine input nodes and three hidden nodes. Since there is only one type of object, it is sufficient to have only nine input nodes, that is, one node in each group described in section 3.3. The number of hidden nodes was determined experimentally, and was set to one node more than there are sensory stimuli. For each perception that the ant makes, it can either sense an object or an empty cell. There are thus two different stimuli that may be perceived by the ant, and hence a network with three hidden nodes is chosen as the ant's controller. In this clustering experiment there are 50 objects and 5 ants located in a 23×23 grid-world. Each individual environment containing objects and ants is stepped 4000 times before it is evaluated, and the evolutionary run is 6000 generations long.

4.1.1 The Evolution of a Solution

The fitness of the fittest individual and the average fitness of each generation is shown in Figure 4.2. Only the first 2000 generations have been included in the figure because there is little change in the fitness of the fittest individual after generation 180, and the average fitness of each generation stabilizes around generation 300. It is clear from the figure that a solution to the clustering task is evolved very quickly. The fittest individual of the first generation has a fitness of 64, and after only four generations the fitness of the fittest individual has more than doubled and reached a score of 144. After only 160 generations the fittest individual has reached the maximum achievable fitness of 324, and 20 generations later the fitness of the fittest individual stabilizes around this value for the rest of the evolutionary run.

To see what takes place in the evolutionary run during the early period of rapid change, three of the individuals from this period are selected for an investigation of their sorting process. Because the change in fitness is so fast during the early stages of the evolutionary run, the selected individuals will be very close in terms of the generations that they are from. The first selected individual is the fittest individual from generation 3, which has a fitness of 102. The next individual is from generation 6 and has a fitness of 200, and the final individual is from generation 13

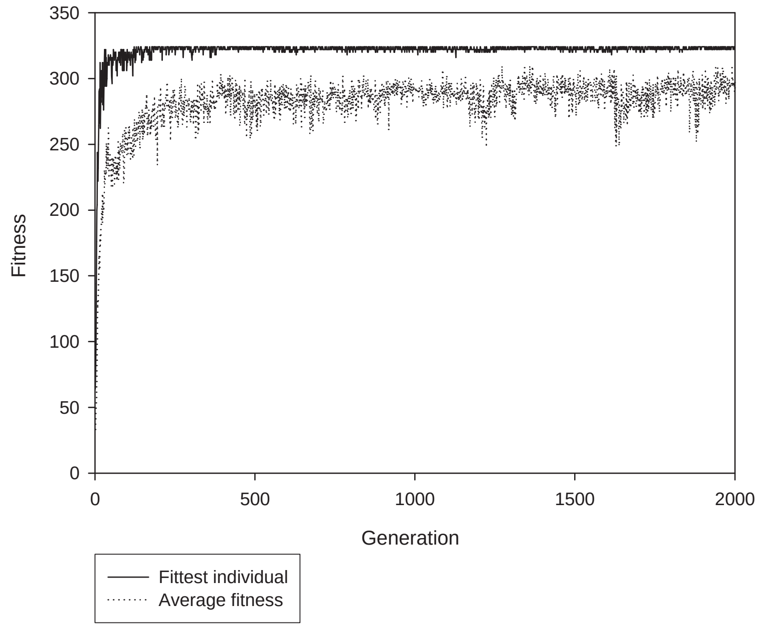

Figure 4.2: Fitness during the evolution of a clustering solution.

with a fitness of 290. By investigating these three individuals[1] it is possible to see how evolution progresses in terms of the individuals becoming increasingly better at clustering objects. The end results of clustering performed by the three individuals is shown in Figure 4.3.

The fittest individual from generation 3 clusters around two thirds of the objects into several small, loose clusters, while the remaining third of the objects are scattered randomly in the world. The ants of the individual preferentially pick up isolated objects, but do also occasionally pick up objects that are part of a group. After the ants have picked up an object they drop it in the next step, irrespective of whether it is in a densely or sparsely populated neighborhood. The preferential pickup behavior causes the formation of small groups, while the constant deposit causes some objects to be isolated. The movement of the ants of this individual is usually in straight paths backward, but when there are objects present in the ants' neighborhood they sometimes turn left.

The movement of the ants of the fittest individual from generation 6 is identical to the movement of the ants of the fittest individual from generation 3. The individual is, however, capable of clustering all of the objects into three to five loose clusters. The formed groups are larger than the ones formed by the previous individual, and there are no isolated objects. This change is mainly due to the ants only depositing objects in neighborhoods where there are other objects. After the clusters are formed the ants continue to remove objects from them, and this sometimes cause parts of the clusters to break off from the clusters. These fragments are, however, quickly picked up and deposited in one of the clusters.

The fittest individual from generation 13 is capable of forming one or two dense clusters. If two clusters are initially formed, one of them is usually much smaller

[1]A detailed description of the sorts performed by the individuals is given in appendix A.1.

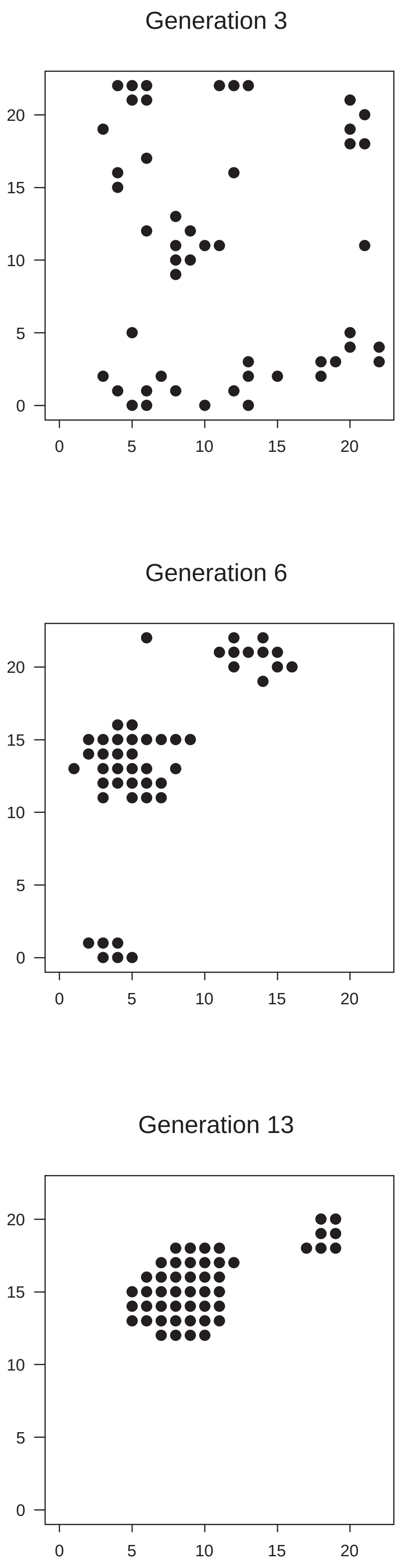

Figure 4.3: Clustering performed by the fittest individuals from generations 3, 6, and 13.

than the other. If the individual is allowed to run for a long time the smaller cluster is eventually broken down, leading to the formation of a single cluster. The ants of the individual first form loose clusters that eventually become densely packed with objects. Then all but one or two of these dense groups are broken down, leading to the final distribution of objects. The only difference in the movement of the ants of this individual and the two previously described individuals, is that the ants sometimes turn right in addition to left when there are objects present in their neighborhood.

From the investigation of the individuals it is evident that the ants of the individuals first evolve a proper pickup behavior, and then a proper deposit behavior. The ants first evolve a preference for picking up isolated objects, and then a preference for depositing objects in groups. These behaviors are refined throughout the evolutionary run leading to progressively better solutions to the clustering task.

4.1.2 The Final Solution

The evolution of a solution to the clustering task was allowed to run for 6000 generations, even though it later proved that successful solutions had evolved after only 160 generations. At generation 6000 all of the five fittest individuals have received perfect fitness scores. In fact, for the last 4000 generations the five fittest individuals of each generation all have received such high fitness scores that they must have clustered the objects into a single cluster of close to perfect shape. The average fitness of the last generation is 293.32 with a standard deviation of 52.44. This indicates that at least one of the individuals of this generation did not perform that well. However, the fittest individual received a score of 324 which constitutes a perfect score on the clustering task when there are 50 objects to be clustered.

Figure 4.4 shows an example of a clustering performed by the final solution. After 2100 steps several groups have been formed, and after 2990 steps one of these groups have grown considerably. A single cluster with a perfect shape is formed after 5840 steps. The way the final solution clusters the objects is as follows. At the beginning of the run of the solution most of the objects are isolated, although there are some small groups of two or three objects. When the ants encounter an object in a sparsely populated neighborhood they will pick up the object. Since most eight-cell neighborhoods only contain one object at the beginning of the clustering process, the ants will pick up the first object they encounter. These objects are carried by the ants for a significant number of steps. The reason for this is that the more objects there are in the ants' immediate environment, the more likely the ant is to deposit its carried object. Since the objects in the world are randomly scattered at this point of the clustering process, the ants rarely encounter an environment in which they are likely to deposit their object. The ants do, however, occasionally deposit their object either next to another object, or in most cases next to a pair of objects. These groups occasionally reach a size of five to six objects. After some time most of these groups are, however, broken down as the ants pick up objects from them.

At the beginning of the clustering process there is thus a phase were the ants are mostly loaded with objects. During this phase small groups of objects are occasionally formed and broken down. This phase lasts until one or more of the groups contains around seven to eight objects (step 2100 in Figure 4.4). Once this happens the group will rapidly increase in size. The number of objects needed to trigger the growth of

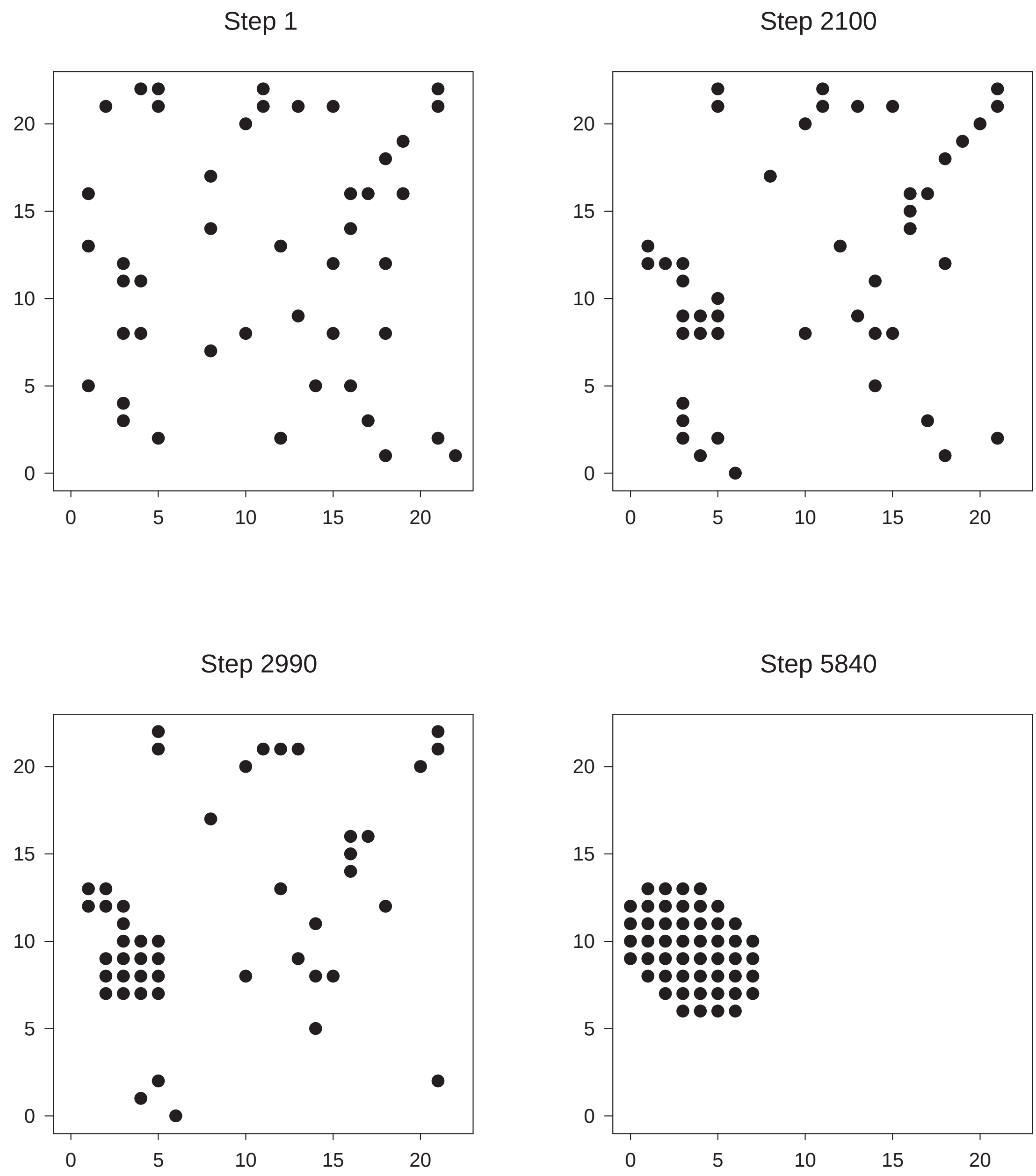

Figure 4.4: An example of cluster formation by the final solution. After 5840 steps a single cluster with a perfect shape is formed.

a group depends on the shape of the group, but a square group of nine objects will always grow rapidly.

The growth of one or two groups of objects signifies the onset of the second phase of the clustering process. In this phase the ants are initially mostly loaded, but as more and more objects become part of the growing clusters, the ants are loaded less of the time. This is due to the fact that the ants do only rarely pick up an object from a large, dense cluster of objects. As more and more of the objects become part of a cluster of this type there are consequently fewer objects for the ants to pick up. At the beginning of the growth phase it is mostly the isolated objects that are deposited in the growing groups (step 2990 in Figure 4.4). After a while all of the objects become part of either the growing cluster or a smaller group. The smaller groups are eventually broken down. Depending on whether there were one or two groups that started growing there may be a third phase of the clustering process. If only one group started growing the second phase will end with only one cluster, but if two groups started to grow the phase will end with two clusters, and a third phase will follow. In this phase the ants are usually unloaded, and the phase may take a long time to complete. The ants do occasionally pick up an object from one of the clusters and deposit it in either cluster. As a result of this one of the clusters will eventually become bigger than the other cluster. The ants will continuously remove and deposit objects in the clusters until the smaller cluster reaches a critical size of seven objects or less, depending on the shape of the cluster. Once this happens the small cluster will rapidly disintegrate leading to the formation of only one cluster.

The clustering performed by the final solution appears to be based on two specific features of the ants pick up and deposit behaviors. First, the ants mostly pick up objects that are located in sparsely populated environments, and rarely remove objects that are part of larger, dense clusters. Second, the ants usually deposit their objects only in densely populated areas, although they may occasionally deposit an object next to pairs of objects. These characteristics of the ants' pick up and deposition behaviors enable the final solution to cluster a set of identical objects.

The movement of the ants of the final solution is usually in the backwards direction regardless of whether an ant is carrying an object or not. In fact, the ants display the same type of movement both when they are carrying an object and when they are unloaded. When there are no objects present in the ants' immediate environment they move in a straight path backwards. However, when they perceive objects in their eight-cell neighborhood their movement changes. The presence of objects in the ants' surroundings cause the ants to turn left or right. In some situations the ants will also move in the forwards direction. The default behavior of the ants can thus be said to be in a straight path backwards, with this behavior being modified by the perception of objects.

The observation of the actions of the ants is not the only way to look at the clustering process. An alternative approach is to evaluate the distribution of objects at each step of the process with the fitness function. The score of the process at each step may then be investigated to see both how the clustering takes place, and to get an indication of the stability of the cluster once it is formed. The score at each step of the clustering process of Figure 4.4 is shown in the graph in Figure 4.5. In the graph it can be seen that there is first a brief period where there is not much increase in the score from step to step. This is the period when small groups of

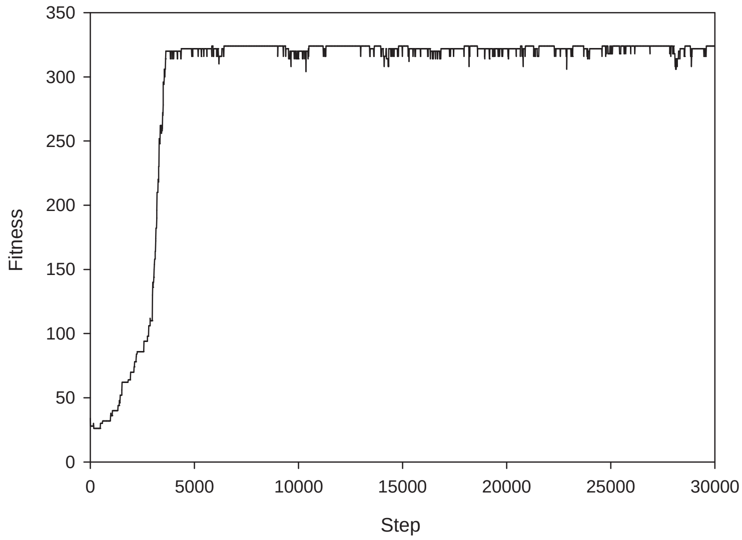

Figure 4.5: The score at each step of the clustering process shown in Figure 4.4.

objects are occasionally formed and broken down. After this brief period the score starts to increase. The increase is slow at the beginning, before it suddenly increases rapidly to the maximum value. At the beginning of the increase one of the clusters has begun to grow. As more and more of the objects become part of the growing cluster the score of the distribution of objects increases. The reason why the increase is slower in the beginning than in the end is due to the fact that at the beginning objects were removed from smaller groups and then deposited in the growing cluster. Then, after some time isolated objects were moved to the cluster. The objects that were earlier located next to other objects do not add as much to the score when they are moved to the growing cluster as isolated objects, and the increase in the score is thus slower at first. The graph also gives an indication of the stability of the cluster once it is formed. After the ants have successfully collected all of the objects in a single cluster, they continue their actions and occasionally remove an object from the cluster. However, as is evident from the graph, once the cluster is formed it never deteriorates and it is thus stable once it is formed.

Testing Sensitivity to Initial Conditions

An important feature of the clustering solution is whether it is stable over a wide range of start configurations (different distributions of objects and ants). To investigate whether the solution is capable of clustering a collection of objects irrespective of their initial configuration, the solution was run on 50 different start configurations. Then the average fitness of the 50 runs and the standard deviation were calculated. To get an indication of how long the clustering process takes on average, as well as to see if the clusters are stable once formed, the 50 runs were repeated with different numbers of steps. The solution was run with run times of 5000, 10000, 15000, and 20000 steps with 50 runs for each runtime. The results of the runs are summarized in Table 4.1. The first thing to be noticed from the results is that the average fitness of the runs that were run for 5000 steps is somewhat lower than for the other run times.

Individual runtime	Average fitness	Standard deviation
5000	310.36	36.34
10000	319.44	9.71
15000	319.00	9.04
20000	321.28	5.43

Table 4.1: Results of 50 runs of the clustering solution with different start configurations and run times.

The reason for this is probably that if two clusters are initially formed, a runtime of 5000 steps is not enough time for the clusters first to form, and then for one of them to be broken down. For the longer run times the time is however long enough for one of the clusters to be broken down completely if two clusters are formed initially. Another thing to notice is that the standard deviation is very low for the three last run times. A score of 319 means that all of the objects have been clustered into a single cluster. The only difference between a cluster with this score and a cluster with a perfect score is the shape of the cluster. The low standard deviation thus means that the solution is capable of clustering the different initial configurations of objects into a single cluster. The high score of the longest run times also means that the clusters are very stable once they are formed, despite the continuous actions of the ants.

4.2 Two-Object Patch Sorting

Patch sorting is the task of creating several clusters, each containing objects of only one type, which are separated from each other. Since there are two types of objects in this experiment, the task to be solved is to create two clusters that are spatially separated. Each of the clusters must contain objects of only one of the two types. As with the clustering task, this task may also be viewed from a local perspective. In the clustering task each object should be surrounded by as many other objects as possible. However, since there is now more than one type of object, restrictions must be applied to the surrounding objects. In a successful patch sort solution each object should be surrounded by as many other identical objects as possible, but at the same time it should be surrounded by as few objects as possible of the other type. To calculate the fitness of a patch sort solution an extension of the strategy used for clustering is used. The objects of the solution can be located in three different general kinds of neighborhoods. An object can either be isolated, it can be in an area with identical objects, or it can be in an area with objects of another type. Objects that are in each of these different situations should contribute to the overall fitness of the solutions in different ways. An object that is located in an area with a high density of identical objects should contribute positively to the fitness, whereas an object that is located in an area with a high density of different type objects should contribute negatively to the overall fitness. Objects that are isolated can be said to be neutral in the sense that they are not pertinent to the overall solution. To compute the fitness of a solution to the patch sort task, each object in the solution is investigated in turn. For each object, the surrounding objects in its neighborhood are considered. For each of these objects that are identical to the object being evaluated a reward is given, and for each object that is of a different type a punishment is given. The function used to calculate the fitness of a patch sort solution is given in Equation 4.2.

$$\sum_{i=1}^{n_o} \sum_{j=1}^{c(o_i)} s(o_i, o_j) \tag{4.2}$$

In the equation n_o is the number of objects in the world, and the function $c(o_i)$ returns the number of objects that are located in object o_i's neighborhood. The function $s(o_i, o_j)$ is a discrete similarity function that returns 1 if object o_i and o_j are of the same type, and -1 if they are of different types.

The feedforward network used as controller for the ants has four hidden nodes and 18 input nodes. As there are now two kinds of objects, there are three distinct sensory stimuli that may be perceived by the ant. These are either one of the two types of objects, and nothing. Since there are three types of sensory stimuli we have chosen to use four hidden nodes in the networks controlling the ants. The increase from one to two types of objects causes a doubling in the number of input nodes of the network. With two types of objects there will be two nodes in each of the nine input groups of the network, making a total of 18 input nodes. In this experiment there is a total of 50 objects with 25 objects of each type. These objects are located in a 23×23 grid-world and are moved around by a swarm of 5 ants. Each individual of a generation is allowed to step 4000 times before being evaluated, and the evolutionary process runs for a total of 6000 generations.

4.2.1 The Evolution of a Solution

The fitness of the fittest individual of each generation and the average fitness of the generations are shown in Figure 4.6. As can be seen in the figure there are two main phases of the evolutionary run. In the first phase there is a rapid increase in both the fitness of the fittest individual and the average fitness of each generation. The fittest individual of the first generation has a fitness of -10.88, but after only 217 generations the fitness of the fittest individual has increased to 166. After a total of 805 generations the fittest individual has a fitness of 284, and this marks the end of the first phase of the evolutionary run. In the second phase there is not much increase in the fitness of the fittest individual. However, the fitness of the fittest individual becomes less variable as the run continues. This is an indication that the fittest individual becomes better at performing a patch sort from a large variety of initial distributions of objects and ants. Although there is not much increase in the fitness of the fittest individual, the average fitness of the generations continues to increase to some extent throughout the evolutionary run.

Of the two different phases of the evolutionary run, it is the first that is most interesting in terms of how the sorting process of the fittest individual changes. During this phase there is a rapid increase in the fitness of the fittest individuals, and it is clear that the fittest individual of each generation quickly improves on the task of sorting two types of objects into two distinct clusters. To see how the sorting process changes during this phase of rapid improvement, the fittest individuals from four generations are chosen for a closer investigation. These are the fittest individuals from generations 77, 158, 550, and 720. These individuals have a fitness of 98, 150, 200, and 250 respectively. The first individual that is selected has a fitness of around

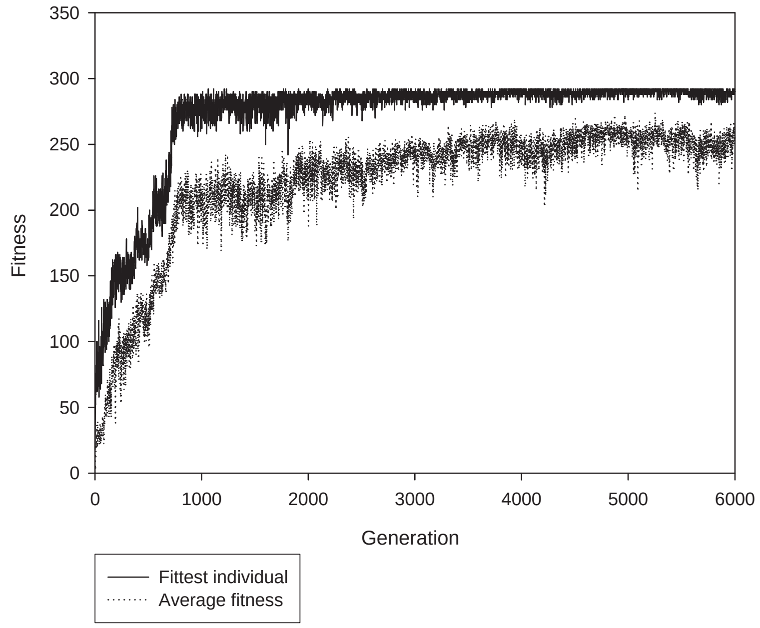

Figure 4.6: Fitness during the evolution of a solution to the two-object patch sorting task.

100 because it is assumed that individuals that have a lower fitness than this do not show a behavior that is interesting in terms of patch sorting. After this first individual, the next are selected for every 50-unit increase in fitness. By examining the sort performed by these four individuals[2] it is possible to get an idea of how a solution to the two-object patch sorting task is evolved. The result of sorts performed by the four individuals are shown in Figure 4.7.

The fittest individual from generation 77 is capable of forming three or more loose groups of type 1 (T1) objects. The type 2 (T2) objects are, on the other hand, only put in groups of two or three, or they are isolated. The ants of the individual carry T1 objects longer than T2 objects before they are deposited, and they are mostly put down in areas with other T1 objects. T2 objects are, on the other hand, deposited in isolated areas or next to other T2 objects. This leads to the individual first forming groups of T1 objects, for thereafter to form small groups of T2 objects. The ants of the individual move in circular paths throughout the sort.

The movement of the ants from generation 158 differs from the movement of the ants from generation 77 when they are carrying a T1 object. Instead of moving in circles the ants move in a straight path backwards, and when objects are present in their neighborhood they turn right. The individual start the sorting process by first forming several small groups of T1 objects. Then all of these are broken down except for two or three groups that grow in size until they contain all of the T1 objects. After these groups begin to form, some groups of T2 objects also emerge, but these are quickly broken down. The end result of the sorting process is two or three groups of T1 objects that are more compact than the groups formed by the previous individual. The T2 objects are, however, only scattered and not forming pairs.

[2]A detailed description of the sorts performed by the individuals is given in appendix A.2.

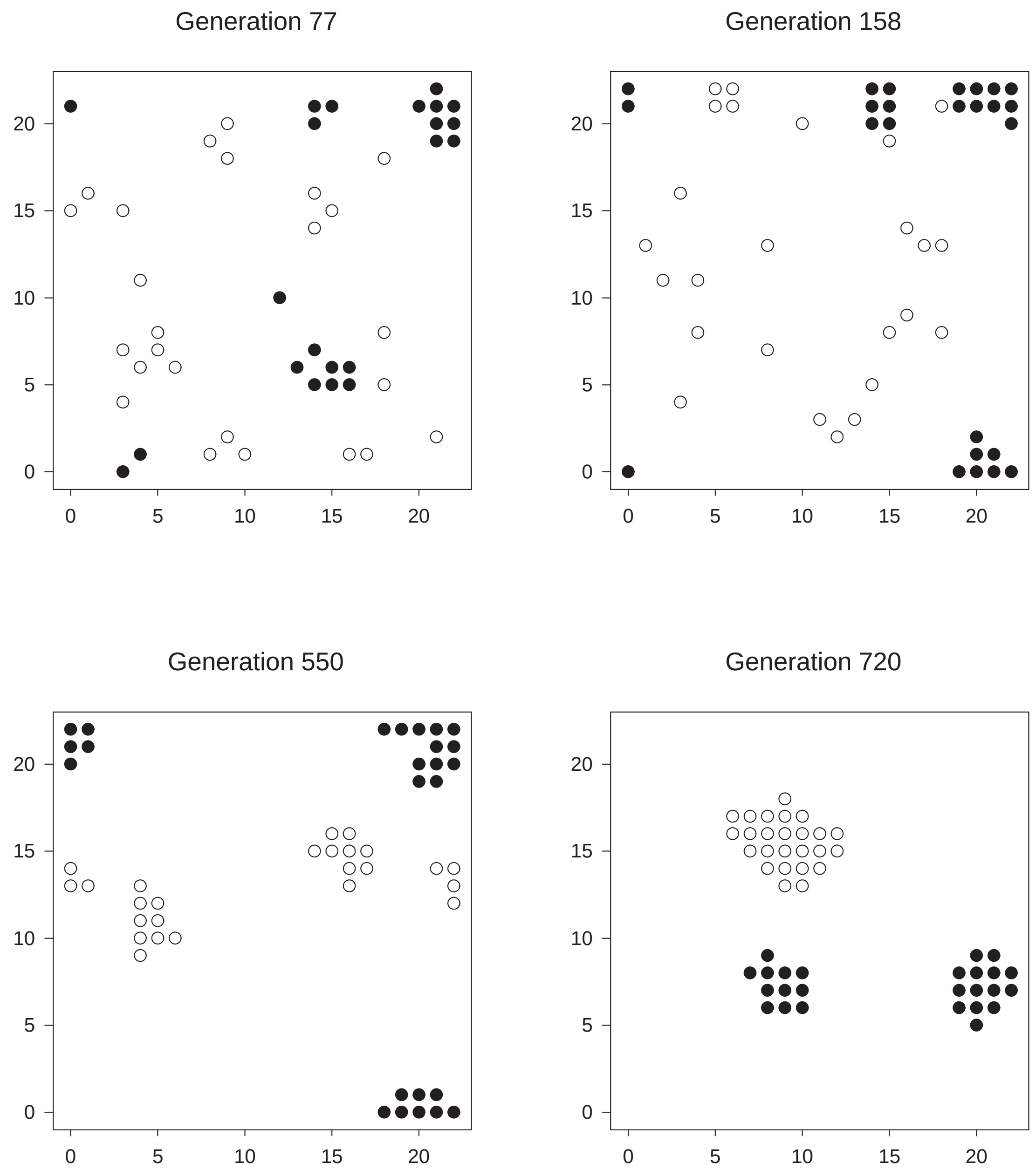

Figure 4.7: Two-object patch sorts performed by the fittest individuals from generations 77, 158, 550, and 720. In the figures filled circles are T1 objects, and empty circles are T2 objects.

The fittest individual from generation 550 forms larger groups of T2 objects, in addition to one or two groups of T1 objects. This was not done by the previously described individuals. Like the individual from generation 158, this individual also begin by forming groups of T1 objects. One of these then increases in size, and as this happens the T2 objects also begin to be grouped. The T2 objects eventually form three or four smaller groups that are not as dense as the T1 groups.

The last investigated individual is from generation 720. This individual is capable of performing an almost perfect patch sort. In contrast to the previous individuals this individual begins by forming several groups of each type of object. All the groups of T2 objects are then broken down, except for one that eventually contains all of the T2 objects. The T1 groups are more stable than the T2 groups and consequently remains for a longer period than the T2 groups. Most of them are, however, eventually broken down except for one or two. In most cases this leads to the formation of a single cluster of T2 objects, and one or two clusters of T1 objects. The movement of the ants has changed from the previous individuals when they are unloaded or carrying a T2 object. Like when they carry T1 objects, the ants move in straight paths, and turn either left or right when there are objects present in their neighborhood.

From the investigation of the individuals it is evident that the evolved individuals first become proficient at clustering T1 objects, and thereafter T2 objects. For both types of objects the individuals first become capable of forming several smaller groups of the types, before finally being able to form a single cluster of the type.

4.2.2 The Final Solution

After a total of 6000 generations the evolutionary run is ended. The final generation has an average fitness of 256.20 with a standard deviation of 29.80. Towards the end of the evolutionary run there is not much variation in the fitness of the fittest individuals from each generation, and it appears that the fittest solutions are capable of performing patch sorts from a variety of different initial configurations. In the final generation all of the five fittest individuals have a fitness of 284 or more. The fittest individual of the final solution, which constitutes the final solution to the patch sorting task, has a fitness of 292. This score is the maximum achievable fitness, and it thus represents a perfect patch sort.

The final solution is capable of performing patch sorts from a wide variety of different start configurations. The time it needs to perform a successful sort does, however, depend on the initial configuration of the objects. If the initial configuration contains a small group of either type of object, then this group will quickly ‘attract’ similar objects, and in these cases a patch sort will be performed very quickly. In other cases the patch sort will take longer time. This may happen if all the objects are initially isolated, but this is not what makes the patch sort take the longest. The patch sort takes the longest time if there, at some point, arise several stable clusters of any type. If this happens then it may take a long time before all but one of the clusters are broken down. Figure 4.8 shows how a two-object patch sort may be achieved. In this particular case the patch sort takes a total of 6610 steps, and the final result has a fitness of 290 which is only two points from a perfect score.

The time the final solution takes to perform a two-object patch sort depends on the initial configuration of ants and objects, but in most cases it takes between 5000

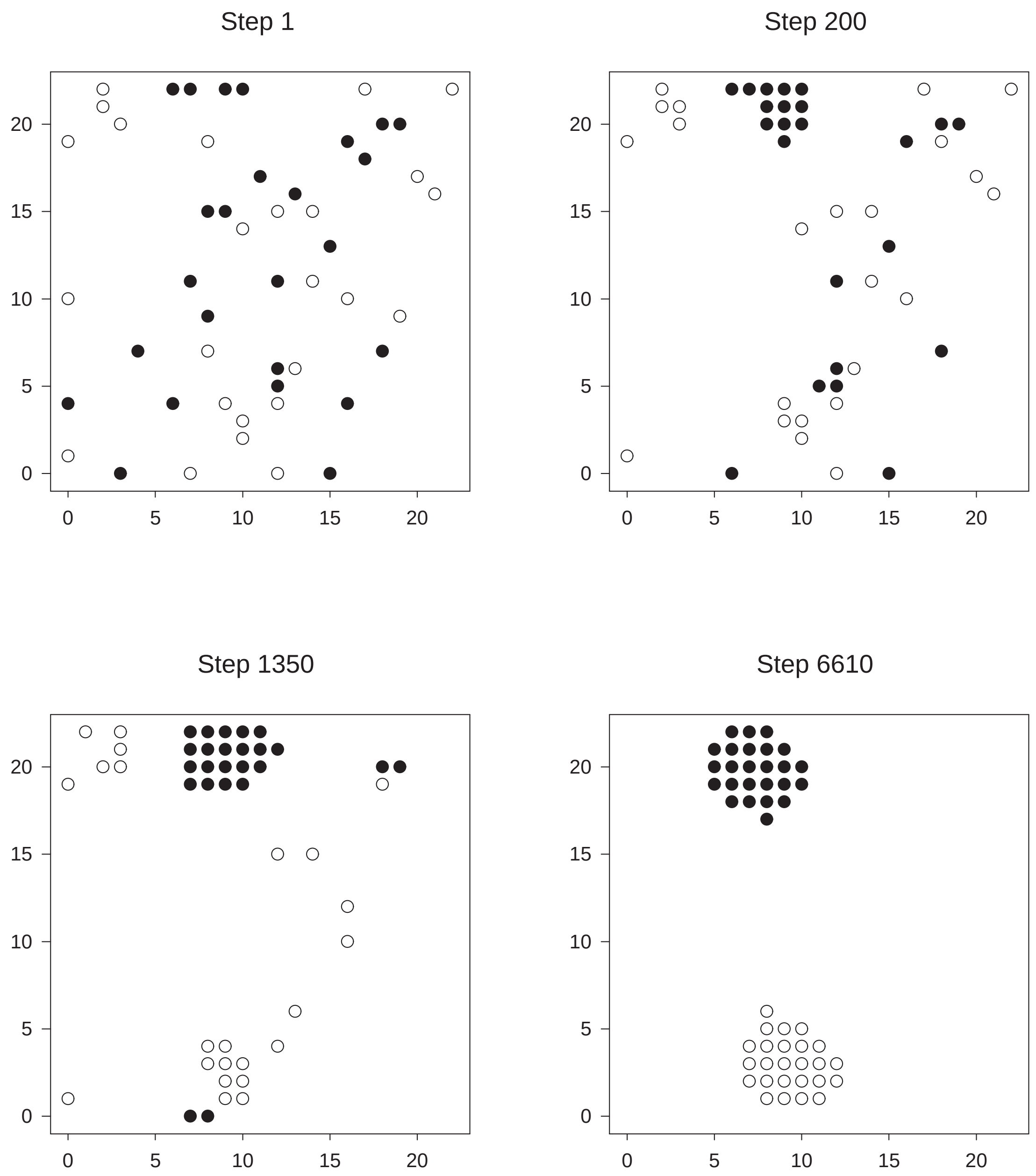

Figure 4.8: The final two-object patch sort solution is able to form two separate clusters in 6610 steps. In the figures filled circles are T1 objects, and the empty circles are T2 objects.

and 7000 steps. At the beginning of a patch sort performed by the final solution, the ants quickly become loaded with either type of object. Which type of object the ants pick up at this point seems to be purely a matter of which object is encountered first. This is because the ants are likely to pick up isolated objects, and at the beginning of a run most objects are isolated. At this point of the run the objects are carried for a relatively long time. The reason for this is that the ants are not inclined to deposit objects in neighborhoods that contain only one, or very few objects, and at this point of the run the majority of neighborhoods in the world are sparsely populated with objects. After around 100 steps a small group of type 1 emerges. Once this group has emerged it does however not increase in size. Then after another 200 steps two to three other small groups of type 1 emerges, and at the same time one group of T2 objects is also formed. Up until this point most of the carried objects have been of type 1, but at this stage the two types of objects are carried about the same amount of time. The groups of T1 objects are very fragile, and they are easily broken down. As some of the groups are broken down, other groups emerge while some of the existing groups increase in size. During the formation of groups of T1 objects, T2 objects are carried a lot of the time but there is no formation of T2 groups. However, after around 1000 steps one of the T1 object groups has increased in size to a point where it contains the majority of the T1 objects. The remaining T1 objects are now either isolated or located in pairs. As this happens a few small groups of T2 objects emerge. One of these groups then increases in size, whereas the others remain about the same size as before. As most of the T1 objects become part of a large cluster, the smaller T2 groups are broken down. The objects removed from these groups are deposited in the large T2 cluster. This eventually leads to the formation of two large clusters, each one containing objects of only one type.

When the ants are unloaded or carrying a T1 object their default movement is in the backward direction only. If the ants perceive any objects in their immediate surroundings they turn either left or right. The movement behavior of unloaded ants and ants carrying a T1 object can thus be described as being in straight paths backwards, with turns being taken if there are objects present in their vicinity. The movement of ants that carry T2 objects differ from the movements of unloaded ants and ants carrying T1 objects in two respects. First of all, ants carrying a T2 object move only in the forward direction. If there are no objects present in the ants' eight-cell neighborhood then the ant will move in a straight path forwards. Second, the underlying cause for the straight movement differs from the reason why unloaded ants and ants carrying T1 objects move in straight paths. These ants move in a straight path because none of the turning actions are active. The ants carrying T2 objects, on the other hand, have both of their turning actions active at the same time. At each step the ants will turn both left and right. These turns cancel each other out and the ant therefore moves in a straight path. When an ant carrying a T2 object encounters objects in its environment it will turn either left or right. Irrespective of whether an ant is carrying an object or not and what object the ant may be carrying there are some configurations of objects in its neighborhood that cause the ants to remain stationary.

The progress and quality of a patch sort may be viewed in terms of the 'fitness' of the distribution of objects at each step of the run of the final solution. By evaluating the quality of the running sort at each step of the sort it is possible to see both how

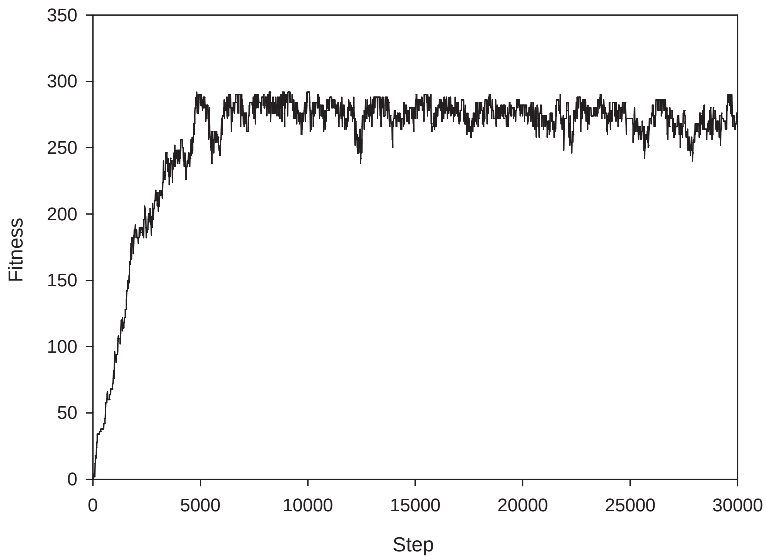

Figure 4.9: The progress of the two-object patch sort shown in Figure 4.8.

the progress of the sort is, and how stable the sort is once the objects in the world have been separated into two clusters. The fitness at each step of the patch sort shown in Figure 4.8 is given in the graph in Figure 4.9. From the graph it is clear that the score of the ongoing sort increases very rapidly at the beginning of the sort. The initial configuration of objects in this particular sort has a score of -2. After 1970 steps the score of the configuration of objects in the world has increased to 192. The reason for this rapid increase is that at the beginning most of the objects are isolated. Then when objects are put next to similar objects, both the deposited object and the neighboring objects will have more neighbors and thus the score will increase. After several groups are formed there will not be much increase in the score when these are broken down to form a single cluster. This is because the objects in the smaller groups already have many similar neighbors. There is thus not a large change in the score of a world that contains a few groups of objects and one that contains one large cluster of each type of object. This is why as the score of the ongoing patch sort gets closer to 200 there is a slower increase in the score. After the patch sort has reached its maximum score after 6610 steps it is evident from the graph in Figure 4.9 that there is not much change in the score. This means that once the solution has performed a patch sort, the sort does not deteriorate despite the ongoing actions of the ants. The evolved solution consequently appears to perform stable patch sorts in the sense that once the objects have been sorted, they are not moved around in a way that destroys the 'sortedness' of the objects.

Testing Sensitivity to Initial Conditions

To further investigate the stability of the evolved solution and its sensitivity to initial conditions, it was run 50 times on different random start configurations. If the average fitness of these runs is relatively high, this is an indication that the evolved solution is stable because it can successfully sort objects with a wide variety of initial

Individual runtime	Average fitness	Standard deviation
5000	270.04	14.16
10000	274.48	10.22
15000	274.76	10.78
20000	275.12	11.08

Table 4.2: Results of 50 runs of the final two-object patch sort solution with different start configurations and run times.

distributions. To see approximately how long a patch sort takes, and if the sort is stable once it is formed, the 50-run trials were repeated with different run times. The evolved solution was run 50 times on different configurations in four turns with individual run times of 5000, 10000, 15000, and 20000 steps. The results of these runs are given in Table 4.2. From the results of the runs it is first of all evident that a two-object patch sort in most cases takes between 5000 and 10000 steps. Even though the average score for 5000 steps is reasonably high, the scores for the other run times are slightly higher and consequently indicate that the runtime for a successful sort is longer than 5000 steps. Apart from the runtime of 5000 steps the remaining results are very similar. This shows that once the objects have been sorted, the formed clusters are very stable despite of the ants occasionally removing objects from them. The scores are also relatively close to the maximum achievable score, and a sort that has a score above 270 will, when inspected visually, in most cases be judged to be a successful patch sort. The results of these four trials of 50 runs thus show that the evolved solution is stable both in terms of the different initial object distributions that it can handle, and in terms of the fact that once a sort has been done it remains stable despite the dynamics introduced by the ongoing actions of the ants.

4.3 Three-Object Patch Sorting

The task to be solved in this experiment is the same as the task to be solved in the two-object patch sort experiment, only now there are three different types of objects instead of two. A successful solution must therefore form three spatially separated clusters with only one type of object in each cluster. The fitness function used to evaluate the proposed solutions is the same as the one used in the two-object patch sort experiment (Equation 4.2). Since there are now three types of different objects the networks used to control the ants have five hidden nodes, one more than the number of different sensory stimuli perceived by the ants. Because there are three different types of objects there are also three nodes in each of the nine input groups of the network, leading to a total of 27 input nodes. In the experiment there are a total of 60 objects with 20 objects of each type. Since there are now more objects than in the previous experiments the objects are located in a 25×25 grid-world with a swarm of 6 ants. The evolution runs for 6000 generations, with each individual in a generation stepping for 4000 steps.

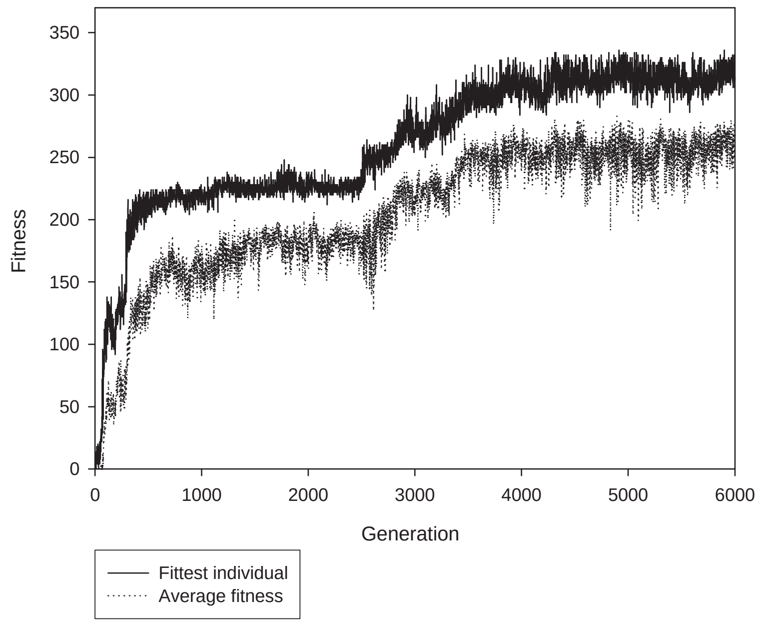

Figure 4.10: The fitness of the fittest individual and the average fitness of each generation during the evolution of a three-object patch sort solution.

4.3.1 The Evolution of a Solution

Figure 4.10 shows the fitness of the fittest individual and the average fitness of each generation during the evolutionary run. From the figure it appears as if the evolution of a solution to the three-object patch sorting task has happened in three major phases. At the beginning of the evolutionary run there is a phase of rapid increase in fitness. This phase begins with the first generation and ends after around 360 generations. In this phase there is a steady increase in the fitness for the first 150 generations. At this point there is a sudden drop in the fitness before it again starts to increase. When the evolution has reached generation 366 the fittest individual has a fitness of 212, and after this there is a phase with only a minor increase in the fitness. During the second phase of only a modest increase in the fitness, there is actually only an increase of 20 points during the 2200 generations that this phase lasts. After a total of approximately 2500 generations the fitness of the fittest individual of each generation again begins to increase. This is the final major phase of the evolutionary run. During the first 2000 generations of this phase the fitness increases at a steady rate, before it levels out and only increases a little for the rest of the duration of the evolutionary run.

To see how a solution to the three-object patch sort task evolves, the fittest individual from four generations during the evolutionary run is selected for a closer investigation. During the first phase it is likely that there is a large change in the sorting process of the individuals because of the rapid increase in fitness. Two individuals are selected from this phase to capture this change. The first individual is the fittest individual from generation 87, which is before the sudden drop in fitness. This individual has a fitness of 108. The next individual is chosen after the fitness has increased another 50 points. This individual is the fittest individual from generation 253 which has a fitness of 156. The third individual is from the point of the run where the increase in fitness slows down, which is where the second phase of the evolutionary run begins. This is the fittest individual from generation 366 which has a fitness of 212. The last individual chosen for a closer investigation is selected from around the middle of the initial growth phase of the third phase of the evolutionary run. This is the fittest individual from generation 3273 which has a fitness of 280. By investigating these four individuals[3] it is possible to get an indication of how a solution to the three-object patch sorting task is evolved. The results of sorts performed by the investigated individuals are shown in Figure 4.11.

The fittest individual from generation 87 is capable of clustering only type 3 (T3) objects. Some of the type 2 (T2) objects are grouped in small groups, whereas the type 1 (T1) objects remain scattered randomly in the world. The individual first forms a cluster of T3 objects, and as this begins to form some of the T2 objects are deposited in small groups that are continuously built and broken down. The T1 objects are deposited immediately after they have been picked up by the ants, and consequently remain scattered in the world. The movement of the ants is the same throughout the evolutionary run. Irrespective of whether they are carrying an object and what type of object they carry, they move in a straight path forward. If there are objects present in their surrounding cells then they turn either left or right, before continuing in a straight path.

The fittest individual from generation 253 also begins by forming a group of T3 objects. After the T3 objects form a single cluster, the T2 objects begin to form small groups. These groups are very fragile and are continuously formed and broken down throughout the remainder of the sorting process. The T1 objects are only carried short distances by the ants, and they are deposited either in isolated locations or in pairs. The end result of a sort by the individual is a single cluster of T3 objects, while the T1 objects mostly remain scattered. Most of the T2 objects form loose groups, which are often in the form of loose chains instead of dense groups.

The ants of the fittest individual from generation 366 are able to sort the T2 and T3 objects into one large cluster each. However, the ants do not carry T1 objects for very long, and as a result of this they remain scattered in the world. The individual start by forming groups of T3 objects, and to a lesser degree, T2 objects. In contrast to the previous individual the T2 groups begin to form before there is a single cluster of T3 objects. After the homogeneous groups are formed, one of each type increases in size while the other groups are broken down. This eventually leads to one cluster each of T2 and T3 objects, while the T1 objects remain scattered.

The fittest individual from generation 3273 forms two or three loose clusters of T1 objects, in addition to one cluster each of T2 and T3 objects. This individual begins by forming groups of T3 objects, then T2 objects, and finally T1 objects. The T3 objects are the first to form a single cluster, and as this happens the T2 objects are first forming loose groups, before being moved to a single cluster. After the T3 and T2 objects are clustered, the T1 objects are grouped into one or two loose groups that are only barely connected.

From the descriptions of the investigated individuals it is clear that during the evolutionary run the individuals first become able to cluster T3 objects, then T2

[3] A detailed description of the sorting processes of the individuals is given in appendix A.3.

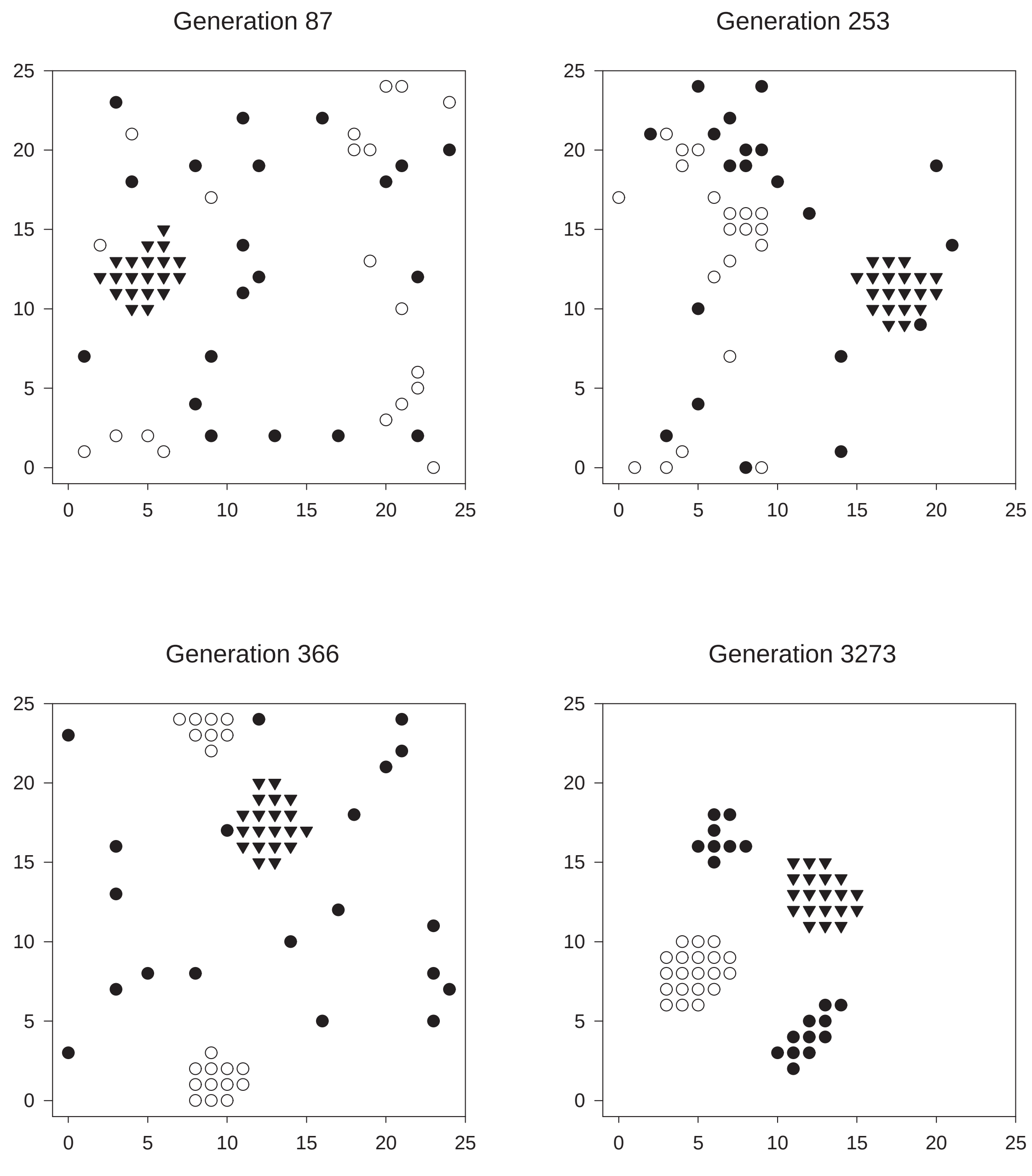

Figure 4.11: Three-object patch sorts performed by the fittest individuals from generations 87, 253, 366, and 3273. In the figures filled circles are T1 objects, empty circles are T2 objects, and triangles are T3 objects.

objects, and finally T1 objects. The sequence in which individuals are evolved that cluster the different types of objects is also reflected in the sequence in which the later evolved individuals cluster the different types of objects. They first cluster T3 objects, then T2 objects, and finally T1 objects.

4.3.2 The Final Solution

The evolution is allowed to run for 6000 generations. At this point the final generation has an average fitness of 270.88 with a standard deviation of 32.97. The five fittest individuals of the final generation all have a fitness of 306 or more. In fact, the five fittest individuals of the generations toward the end of the run are all capable of performing patch sorts that are visually satisfactory, although they do not yield perfect fitness scores. The reason for this is that the fitness function awards extra points for the shape of the clusters and not only for the clusters of each type being separated. The fittest individual of the final generation, which constitutes the solution to the task of three-object patch sorting, has a fitness of 314. This corresponds to 94% of the maximum achievable fitness which is 334.

The final solution is capable of performing a successful three-object patch sort of a large variety of different start configurations. The time it takes to perform the patch sorts does however vary. This varying time is caused by the formation of two equally sized clusters of one type of object. If there are created two clusters of any of the three types of objects that are approximately of the same size, then the solution takes longer time to finish successfully. The reason for this is that the objects in a large cluster of identical objects are not easily picked up, and it therefore takes a long time before one of the clusters is finally broken down, and the objects moved to the other cluster of the same type. Figure 4.12 shows how a patch sort may happen in 7970 steps. The final structure has a fitness of 334, which is the maximum score.

The movement of the ants of the final solution is only in the forward direction. Irrespective of whether or not the ants are carrying an object, and what type of object they carry, they show the same kind of movement. If the ants move in desolate areas of the world they will move forward in a straight path. In the case of the ants being unloaded or carrying a T2 or T3 object, the ants move in straight paths simply because they are neither turning left nor right. However, if the ants are carrying a T1 object they move straight because they are at each step turning both left and right. The two turning behaviors thus cancel each other out and the ants move straight ahead. When the ants perceive any objects in their eight-cell neighborhood their movement is modified. In most cases the occurrence of objects in the ants' immediate surroundings causes the ants to turn either left or right, before continuing in a straight path. However, there are some configurations of objects in the ants' environment that cause them to move in both directions, thereby leaving the ants stationary. The ants are in most cases only stationary for one step because they also turn in addition to moving both forwards and backwards, and the perceptual input of the ants are thus different in the next step allowing the ants to move forward again.

When the final solution is run on a three-object patch sort task, its ants quickly become loaded with any type of object. After only a few steps the majority of the ants are however carrying a T2 or T3 object. The reason for this is that when the objects are scattered, it is mostly the T1 objects that are deposited because these

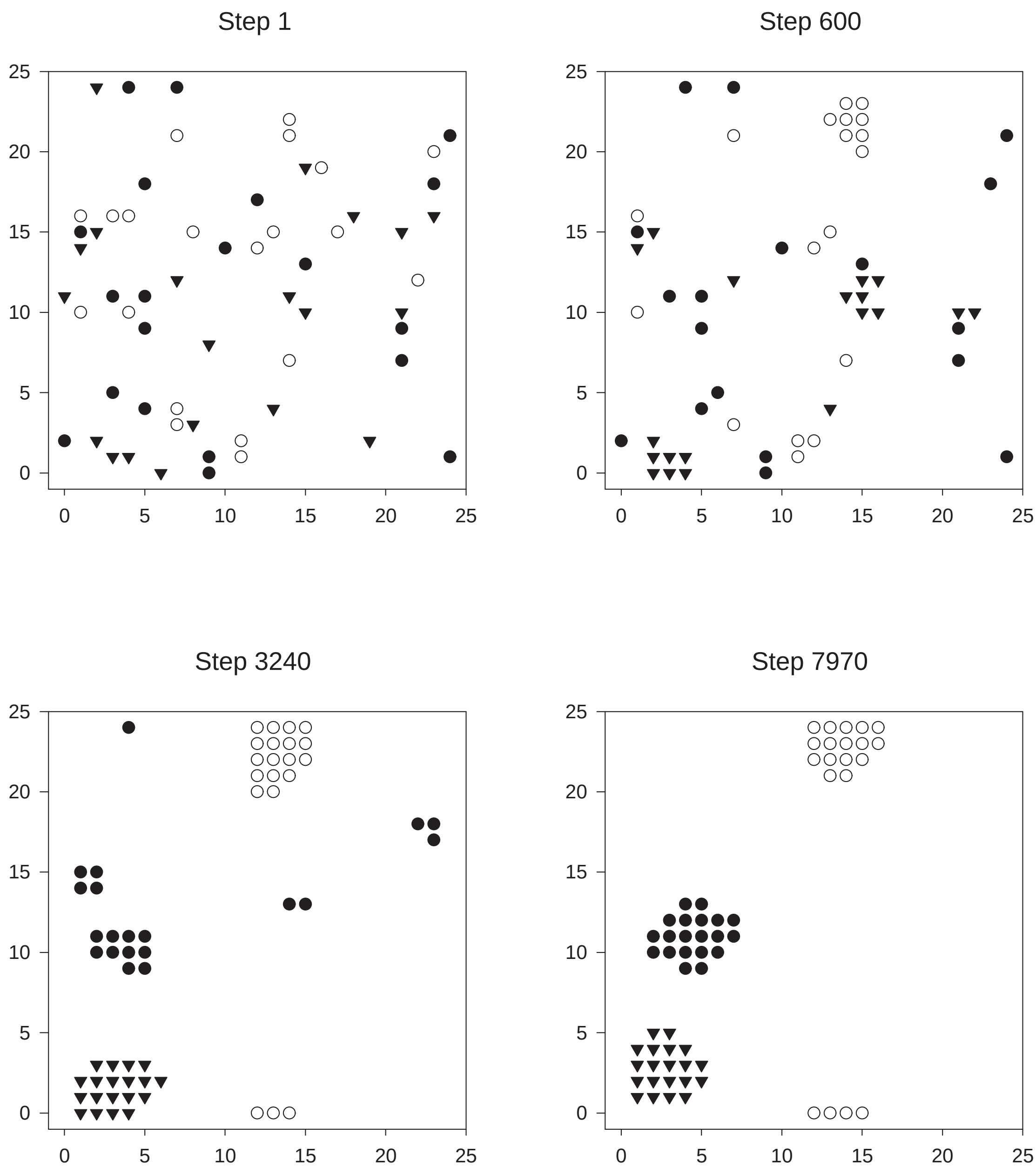

Figure 4.12: The final three-object patch sort solution is capable of forming three separate clusters in 7970 steps. In the figures filled circles are T1 objects, empty circles are T2 objects, and triangles are T3 objects.

are not only deposited in neighborhoods that are densely populated with identical objects. It is not long before all of the T3 objects are part of a small group. Shortly after this happens, one or two of the groups of T3 objects begin to quickly increase in size as the smaller groups of T3 objects are broken down. When this happens most of the T2 objects also become part of smaller groups of identical type. One of these groups then begins to grow such that it is considerably larger than the other groups (step 600 in Figure 4.12). Shortly after this growth occurs the T3 objects are all grouped into one cluster. If there was initially only one T3 group that increased in size, this will eventually contain all the objects of this type, but if it were two that initially began to grow, one of these will increase in size while the other is broken down. As all the T3 objects become part of one large cluster, and one or two of the T2 groups begin to grow, most of the T1 objects are also grouped into small groups. When this occurs the ants are not loaded as much of the time as they were earlier in the run. The run continues with the T1 objects being collected in fewer, larger groups. At the same time the T2 objects are all collected in a single cluster (step 3240 in Figure 4.12). As this happens one of the T1 groups increases in size while the other groups of this type are slowly broken down. The objects will eventually be sorted in three clusters with each cluster containing only objects of one type.

To summarize the patch sorting performed by the final solution one can say that it sorts the three types of objects in order. First the T3 objects are grouped, and then the T2 objects are grouped. Then the T3 objects are clustered into a single cluster, and the T2 objects are grouped in fewer groups. At the same time the T1 objects are grouped in several smaller groups. After the T3 objects are all contained in a single cluster, one of the T2 groups will increase in size until it contains all the objects of this type. As this cluster is formed the T1 objects are grouped in fewer groups until eventually one of the groups grows until it contains all of the T1 objects. The patch sorting thus happens by first forming a T3 cluster, then forming a T2 cluster, and finally forming a T1 cluster.

The patch sort performed by the final solution may be investigated in terms of an evaluation of the current progress at each step of a run of the individual. For each step of the individual the distribution of the objects in the world is evaluated with the fitness function used during the evolution of the solution. Figure 4.13 shows a graph giving the fitness of the progressing sort depicted in Figure 4.12. From the graph it is evident that the sort progresses very rapidly in the beginning. During the first 3200 steps there is a rapid increase in the score of the ongoing sort. After this point the rate of increase in the score slows down somewhat until it reaches its maximum after 7970 steps. The reason for the reduced rate of increase is the following. At the beginning of the sort most of the objects are isolated. As objects are picked up and put down next to other objects of the same type, the score of the ongoing sort increases rapidly. However, after some time most of the objects are located in groups of identical objects. At this point objects that are moved from one group to another do not add as much to the score as isolated objects that were previously added to the groups. Because of this the score of the sort does not increase as rapidly as it did earlier in the sorting process.

The graph in Figure 4.13 also gives an indication of the stability of the performed sort. Even after all three types of objects have been separated into three distinct clusters, the ants continue to move around in the world and occasionally pick up

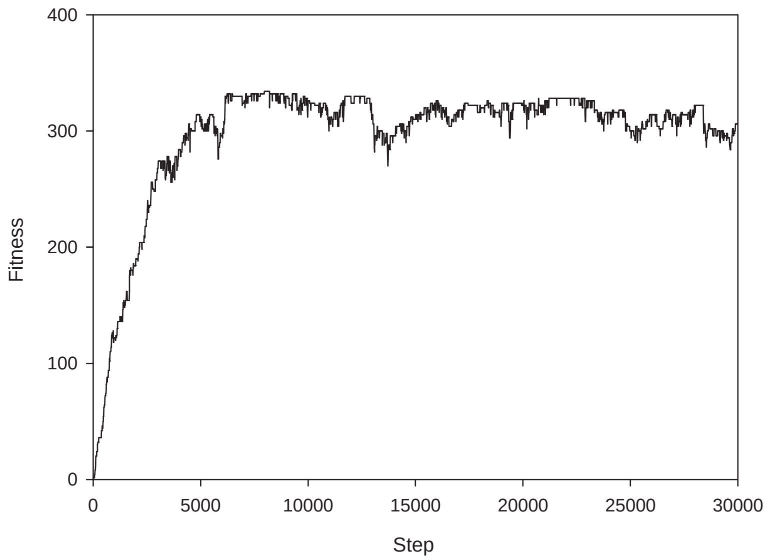

Figure 4.13: The fitness at each step of the sort shown in Figure 4.12.

Individual runtime	Average fitness	Standard deviation
5000	285.02	23.49
10000	298.52	14.91
15000	306.64	14.78
20000	311.92	13.98

Table 4.3: Results of 50 runs of the final three-object patch sort solution with different start configurations and run times.

objects. It is therefore possible for the ants to destroy the performed sort. The graph does, however, show that after the objects have been separated into three clusters the score of the sort remains relatively high. There are some times when the score decreases, but then it goes back up again. It thus appears that the performed sort is stable despite the ongoing actions of the ants.

Testing Sensitivity to Initial Conditions

As mentioned previously the evolved solution uses different times to perform a successful sort depending on the initial configuration of objects. To investigate how long, on average, the solution takes to perform a successful sort, and to ensure that the solution is stable in terms of being able to sort different initial distributions of objects, the solution was run with several different initial distributions of objects and for varying amounts of time. The final solution was run for 5000, 10000, 15000, and 20000 steps on 50 different initial distributions of objects. Then the average score of these 50 runs and the standard deviation were calculated. These results are shown in Table 4.3. When looking at the results of the runs it is first of all evident that the successful sort of three types of objects does on average take quite a long time. The average score of the runs continues to increase as the solution is allowed to run for a longer time. It is evident that a successful sort rarely takes place in 5000 steps, although perfect sorts have been shown to complete in less than this time. It appears that a relatively good sort that is visually satisfactory takes between 10000 and 15000 steps to complete. The results in Table 4.3 also show that the solution is capable of performing relatively well on the different random start configurations since the standard deviation is low. Furthermore, it appears that once a sort is performed successfully it is quite stable despite of the ongoing actions of the ants. Even when the ants are allowed to step 20000 times the sort does not deteriorate even though it probably takes less time to complete. In fact, the average score of the 50 runs that are allowed to run for the longest is higher than for the other runs.

4.4 Two-Object Annular Sorting

The task to be solved in this experiment is that of annular sorting. This consists of forming a cluster of one type of object that is surrounded by bands of the other types of objects, with each band containing only objects of one type. Since there are two types of objects in this experiment a successful solution must create a dense cluster of one type of object that is surrounded by a band of the objects of the other type. The central type objects are hereafter only called *central objects*, and the outermost type objects are called *outermost objects*. To achieve this task a global fitness function is used. This fitness function is a modified version of the metric used by Wilson et al. [33] to judge the quality of annular structures (see section 2.3 for a description of the metric). The metric consists of four components which are compactness, separation, shape, and completeness. However, Wilson et al. [33] argues that the compactness and separation components are the most important when evaluating annular structures. The notion of compactness and separation as being most helpful in describing annular structures was strengthened further when Scholes et al. [31] found that the annular structures formed by real ants got a high score for these two components, but a very low one for the other two. Influenced by these findings we chose to include the compactness, separation, and shape metrics in our fitness function with each component giving a maximum score of 100.

Since the original metric was used to evaluate structures formed by circular objects, it must be adapted to the 'square' objects located in our grid-worlds. The first component of the fitness function is the compactness component. We have chosen to replace Wilson et al's [33] compactness component with a function using the equation used as fitness function for the clustering task (see section 4.1). The task to be achieved in clustering is to create a dense structure containing all the objects, and it will consequently give higher scores for compact structures. The compactness metric takes the score obtained by the fitness function used for clustering (Equation 4.1) and gives this as a fraction of the maximum achievable score with n_o objects in the world.

$$C = 100 \times \frac{\sum_{i=1}^{n_o} c(o_i)}{m(n_o)} \tag{4.3}$$

In the equation $c(o_i)$ is a function that returns the number of objects in o_i's neighborhood, and $m(n_o)$ is a function that returns the maximum achievable score of Equation 4.1 when there are n_o objects.

The second component is the separation component. In a perfect annular structure there are three general types of objects. These are the central objects, the outermost

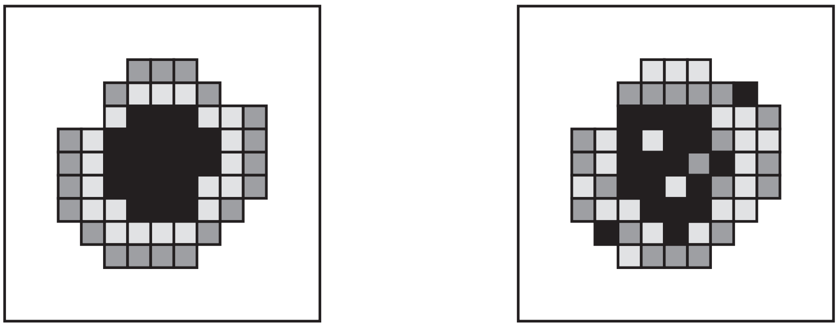

Figure 4.14: The structure on the left has better separation than the structure on the right.

objects, and intermediate objects. These three types should have different positions in the structure. The innermost objects should all be located in the center of the structure, and the outermost objects should be located further from the center of the structure than any other type of object. The center of the structure is taken to be at the centroid of the largest cluster of central objects. For the intermediate objects the situation is a bit more complicated. Every object that belongs to an intermediate band of objects should be closer to the center than objects belonging to bands located further from the center, while at the same time be further from the center than objects belonging to bands that are located closer to the center. It is these notions that the separation component captures by counting the number of objects that infringe on the 'home zone' of another object type. Said simply, the number of objects that are located in wrong positions are counted. Figure 4.14 shows two different structures that have good and only mediocre 'separatedness'.

To count the number of objects that infringe on the home zone of another object type, the first step is to measure the Euclidean distance from the center of the structure for each object. The distances for each type of objects are then sorted and the upper and lower quartiles are found. These quartiles are then used in four different counts. The first count considers the central type objects. These objects should be located closer to the center than objects of other types. To give a measure of this the number of central type objects that have a distance to the center greater than the lower quartile of any other object type is counted, N_c. The second count performed considers the outermost objects. These objects should be located further from the center of the structure than objects of other types. To measure this notion the number of outermost type objects that have a distance to the center that is less than the upper quartile of any other object type is counted, N_o.

The last type of objects to consider are the intermediate objects, which are located in bands between the central and outermost objects. When evaluating these object types both objects located closer to and further away from the center must be considered, and consequently two counts are involved for intermediate objects. The first of these two counts signifies the notion that objects belonging to a band should not be further away from the center than objects belonging to bands located further away from the center. In this count the intermediate type objects that have a distance to the center of the structure that is greater than the lower quartile of any other object type further away from the center are counted, N_i^g. Objects belonging

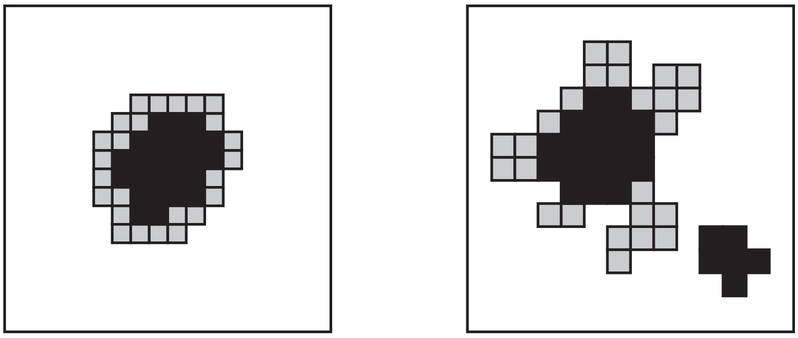

Figure 4.15: The structure to the left will receive a high score on the shape component, whereas the structure to the right will receive a mediocre score. This is because in the structure on the right there is a second cluster of central type objects, and because the outermost objects do not all have the same distance to the center of the structure.

to a band should also not be located closer to the center than objects belonging to bands closer to the center. This notion is realized in the final count. Here objects that have a distance to the center that is less than the upper quartile of any other object type closer to the center is counted, N_i^l. The four counts are combined to form the separation component of the metric as shown in Equation 4.4.

$$Se = 100 \times \left(1 - \frac{N_c + N_o + \frac{N_i^g + N_i^l}{2}}{n_o}\right) \tag{4.4}$$

The last component of the fitness function is the shape component. This component has two parts. The first part considers the central type objects. In a perfect annular structure the central cluster should contain all the objects of this type. The first part of the shape component calculates the fraction of central type objects that are located in the center of the annular structure, f_c. The remaining types of objects should form bands around the central cluster. Every object in each of these bands should have approximately the same distance to the center. This is what is measured in the second part of the shape component. For each non-central type the average Euclidean distance to the center is calculated, $\overline{o}_c$. Then the deviations from this average are summed up for all the objects of the same type, d_c. This summed deviation is then normalized. The equation for the shape component is given in Equation 4.5, where m is the number of object types. Figure 4.15 shows two structures that will receive a high and a mediocre score on the shape component.

$$Sh = \frac{100 \times f_c + \sum_{c=2}^{m} \left[100 \times \left(1 - \frac{d_c}{\overline{o}_c}\right)\right]}{m} \tag{4.5}$$

In a recent study Sendova-Franks et al. [32] found that the brood of the *Leptothorax* ant was first clustered and thereafter spread out to form the annular structure. This might signify that clustering is of particular importance in the formation of an annular structure. Influenced by this we have decided to give the compactness component twice the weight of the other two components in the fitness function when they are added together. The combination of the three components in the fitness function is given below in Equation 4.6.

$$\mathcal{F} = 2C + Se + Sh \tag{4.6}$$

The feedforward controllers used in the annular sorting tasks use the same number of hidden nodes as the controllers in the patch sorting tasks. This means that the feedforward network used for the two-object annular sorting task has four hidden nodes and 18 input nodes. Each individual in the evolutionary run has a total of 50 objects with 25 objects of each type in a 23×23 grid-world, and a swarm of 5 ants to perform the annular sorting.

4.4.1 The Evolution of a Solution

The increase in fitness of the fittest individual of each generation is very rapid in the beginning of the evolutionary run. This can be seen in Figure 4.16. The fittest individual of the first generation has a fitness of 113.26, and after only 21 generations the fittest individual of the generation has a fitness of 246.69. This rapid increase in fitness continues until generation 45 were the fittest individual has a fitness of 299.60. After this generation the increase in fitness is reduced. Although the fitness continues to increase at a relatively high rate, the explosive growth in fitness that occurred in the first 45 generations is no longer evident. This reduced rate of increase in fitness continues until generation 331 where a peak is reached in the fitness of the fittest individual of the current generation. After this point the fitness of the fittest individual drops for a few generations before it again begins to increase. However, this increase is much slower than the previously observed rates of increase. This slower rate of increase in fitness continues until around generation 1000. From this point and through the rest of the run there is not much increase in fitness, and the fitness of the fittest individual varies to a lower degree than earlier in the run. This means that the fittest individual of each generation is capable of performing well on all start configurations that it is given.

During the rapid increase in fitness in the beginning of the evolutionary run, the best individual of each generation is clearly becoming better at solving the annular sorting task very fast. To see how the sorting process of the individuals changes during the first two phases of rapid increase in fitness, four individuals from this stage of the evolutionary run have been selected for a closer investigation. These are the fittest individuals from generations 21, 45, 217, and 331. The fitness of these individuals are 246.69, 299.60, 349.45, and 389.54 respectively. The first individual has been selected because it is from the last part of the period of rapid increase in fitness, and the second individual is selected because it is from the period of transition between the first two growth phases. The third individual is from the middle of the phase of slow increase in fitness, whereas the last is from the peak in fitness. By investigating the sorting behavior of these individuals[4] it is possible to get a notion of how a solution to the two-object annular sorting task is evolved. The result of sorts performed by the investigated individuals are given in Figure 4.17.

The fittest individual from generation 21 start by forming three different types of groups. The first two groups are small groups containing either only central or outermost objects. The third type of group is larger than the other two, and contain both types of objects. As the sort progresses most of the initial groups are broken down, except for two or three groups that contain both types of objects. These groups

[4] A detailed description of the sorting process of the investigated individuals is given in appendix A.4.

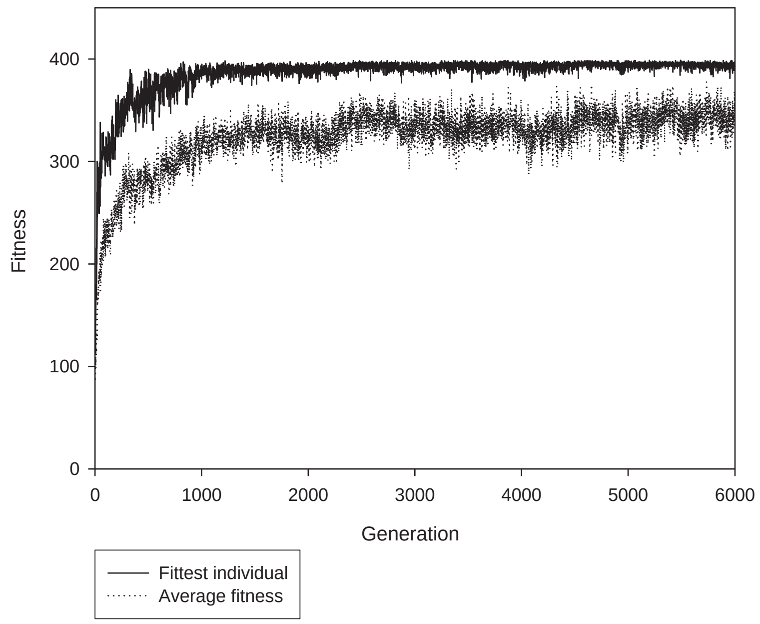

Figure 4.16: Fitness during the evolution of a solution to the two-object annular sorting task.

are not annular in their structure, but the central objects tend to be grouped together in the center of the structure with the outermost objects mostly located on the edges. The movement of the ants of the individual is only in the backwards direction, and if there are no objects in their immediate surroundings they move in a straight path. If the ants perceive any objects they turn either left or right.

The movement of the ants of the fittest individual from generation 45 has changed compared to the individual described above when the ants carry outermost objects. Instead of moving in straight lines, the ants move in circular paths when they carry outermost objects. The individual start by forming two to four groups of central objects. During the formation of these groups some of the outermost objects are moved to their edges, but most remain scattered in the world. However, after the central groups are formed, all of the outermost objects are moved to these groups.

The fittest individual from generation 217 also begins by forming central type clusters. As these begin to form the ants also begin to position outermost objects on the edges of the clusters, or in groups of identical objects. All of the formed clusters are eventually broken down, except for one. The final sort performed by the individual consists of one central cluster with most of the outermost objects positioned along its edges. The remaining outermost objects are either scattered, or located in small groups of identical objects. The movement of the ants has changed back to being in straight paths, irrespective of whether or not they are carrying an object, and what type of object they carry. This movement pattern remains for the remainder of the evolutionary run.

The sorting performed by the fittest individual from generation 331 differs from the sorts performed by the previously described individuals. The individual starts by forming several groups of central objects, before all of these, except one, are broken

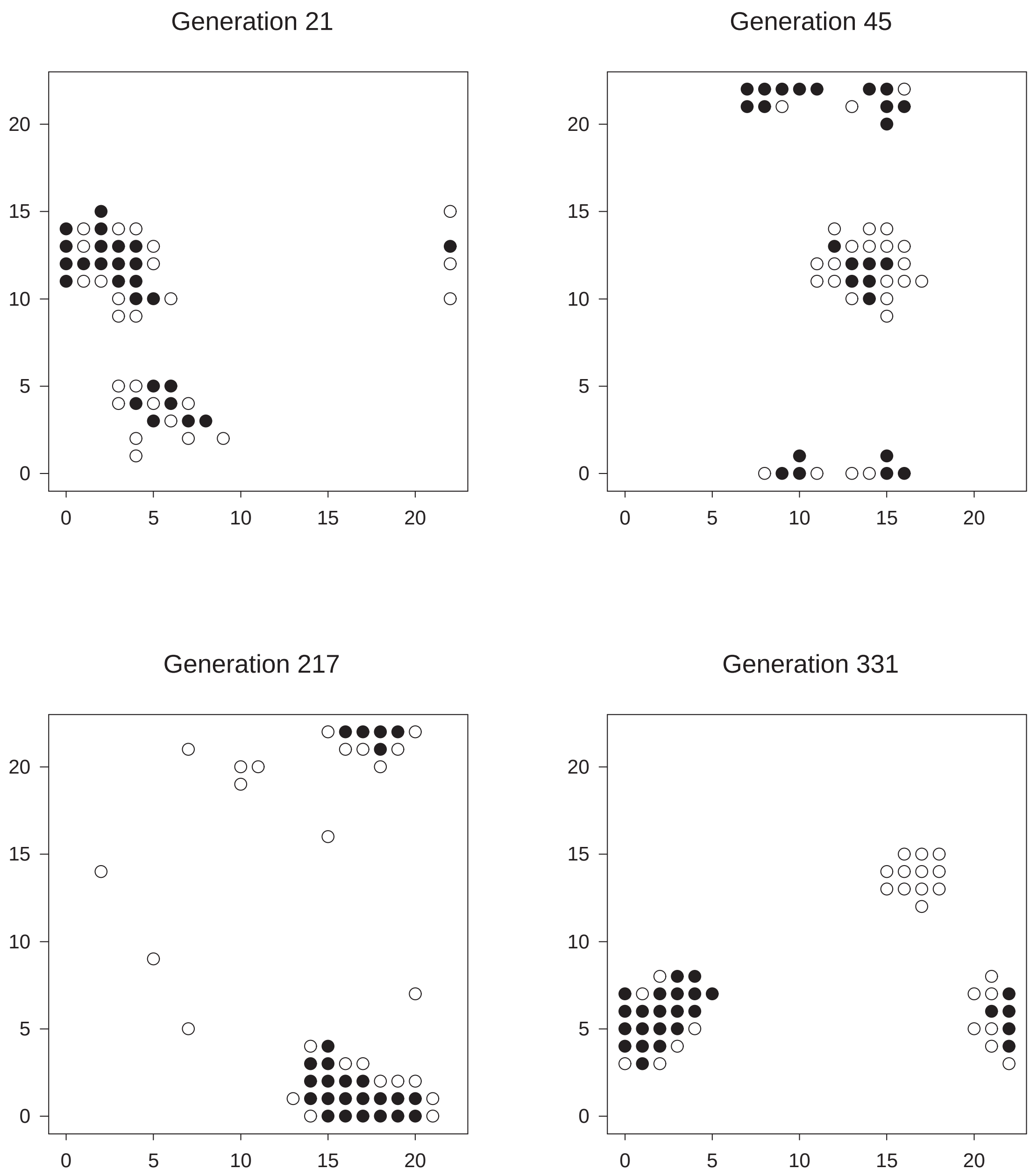

Figure 4.17: Two-object annular sorts performed by the fittest individuals from generations 21, 45, 217, and 331. In the figures filled circles are central objects, and empty circles are outermost objects.

down. It is not until all of the central objects are part of a single cluster that the ants of the individual begin to move outermost objects to a large extent. These objects are either moved to the edges of the central cluster, or deposited next to other outermost objects. The final sort performed by the individual consists of a single cluster of central objects that are surrounded by some of the outermost objects. The rest of the outermost objects form one or more smaller groups.

From the descriptions of the investigated individuals it is evident that individuals that form mixed clusters of objects are evolved first. Then individuals that form several annular-like groups are evolved. As the evolution progresses the evolved individuals form fewer and fewer groups, until they finally form a single annular structure. The later evolved individuals also differ from the earlier evolved individuals in that they cluster the central objects completely before beginning to move the outermost objects.

4.4.2 The Final Solution

The evolutionary run ends after 6000 generations. At this point the final generation has an average fitness of 362.48 with a standard deviation of 31.93. The five best individuals of the final generation all have a fitness above 387. The fittest individual from this generation, which constitutes the final solution to the task of two-object annular sorting, has a fitness of 392.54. This fitness corresponds to 98.14% of the maximum achievable fitness. In practice this fitness score represents a structure that from a visual perspective is perfect. This is due to the fact that the shape metric can never reach a score of 100 in the experiments.

The final solution is capable of forming an annular structure. However, the time that it uses to perform an annular sort depends on the start positions of the objects. Some initial configurations of objects allow the individual to perform an annular sort in a short amount of time, whereas other configurations cause the individual to use longer time before finally achieving an annular sort. Figure 4.18 shows how an annular sort is performed by the final solution in 5080 steps. The annular structure that is formed has a fitness score of 392.90. The compactness component of this score is 100.00, but since it is multiplied by two when the components are added together it contributes a total of 200.00 to the final fitness. The separation component has a score of 100.0, and the shape component has a score of 92.90. The structure is thus perfect in terms of compactness and separation, and a score of 92.90 for the shape is very high since a maximum score for this component is not possible. The final evolved solution is thus capable of performing visually perfect annular sorts that are compact and well separated.

The default direction of movement of the ants of the final solution is backwards, but when the ants are unloaded some configurations of objects in their immediate environment cause the ants to move forward. This forward movement is however very rare. In some situations when the ants are unloaded they also become immobile due to the forwards and backwards actions being active at the same time. Other than these exceptions the ants will move backwards when they are unloaded, and when the ants perceive objects in their neighborhood they will turn either left or right. When the ants pick up an object they will only move backward, and when there are no objects in their neighborhood they will move in a straight line. However,

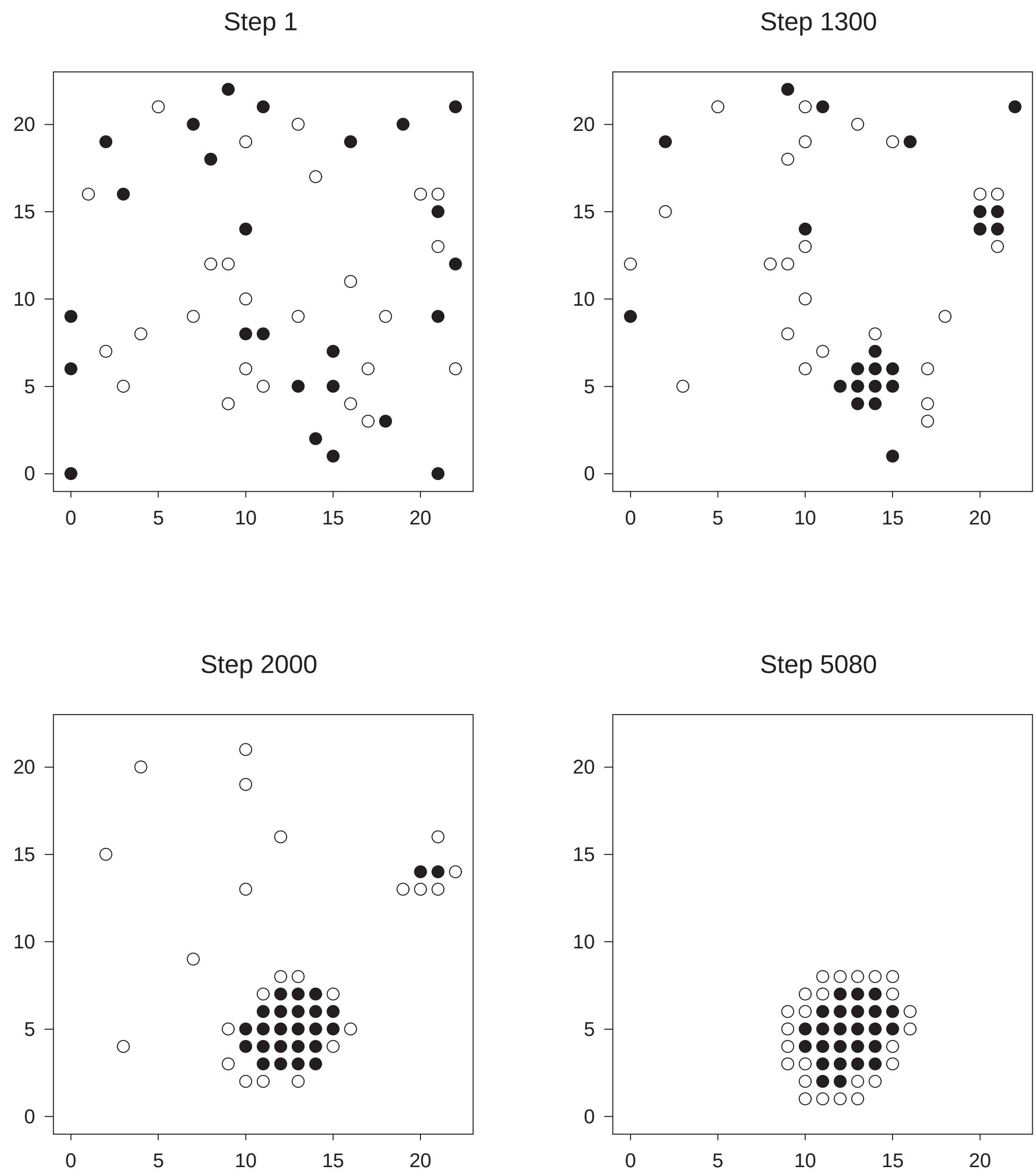

Figure 4.18: A two-object annular sort performed by the final solution. The final structure has a fitness of 392.90 with perfect scores for the compactness and separation components. In the figures filled circles are central objects, and empty circles are outermost objects.

the movement in a line is caused by different basic actions depending on what type of object the ants are carrying. If the ants carry a central object they move straight because none of the turning actions are active. On the other hand, when they are loaded with an outermost object their straight movement is caused by both turning actions being active at the same time, and they consequently cancel each other out. When there are objects in their neighborhood the ants will turn either left of right depending on their sensory stimuli. There are also some sensory stimuli that cause the ants to become immobile by activating both directions of movement at the same time.

At the beginning of a run of the solution most of the objects in the world are in isolated locations. The ants are at this stage unloaded, and when the ants encounter an object of either type they pick up the object. This leads to the ants being initially loaded with both types of objects. However, after a brief period all of the ants become loaded with only the central object type. The reason for this is that the outermost objects are easier to deposit than the central objects. The central objects are deposited in areas with a high density of central objects, and since there are no such areas in the world in the beginning, these objects are carried for a long time by the ants. This leads to the ants becoming loaded with central objects since the ants carrying these objects do not deposit them. The ants that were initially carrying outermost objects will eventually also become loaded with central objects since they easily deposit their outermost object, but once they pick up a central object they are 'stuck' with this object for a long time. After some time with this carrying behavior there is a fluctuation in the world that causes the formation of a small group of central objects. This group quickly increases in size, and once a central object becomes part of this group the ants are not likely to pick it up. This causes the ants to carry the outermost objects more of the time than previously in the run. These outermost objects are for the most part put down along the edges of the central type cluster, but some are put down next to other outermost objects or in isolated positions. The outermost objects that are positioned along the edges of the cluster are rarely picked up by the ants. When all of the central objects and most of the outermost objects become part of the structure, the ants mainly pick up the outermost objects that are scattered around the world in small groups or in isolated positions. All of these objects are eventually moved to the cluster such that it in the end is an annular structure containing all of the objects in the world. After the annular structure is formed the ants will occasionally remove an object from the structure for thereafter to deposit it in the structure again. The final sort that is formed by the solution is thus dynamic in nature in that it never reaches a final static configuration.

The progress of a two-object annular sort may be viewed in terms of the development of the 'fitness' of the distribution of objects at each step of the run. The development of the fitness for the run displayed in Figure 4.18 is given in Figure 4.19. As can be seen in the graph there is first a brief increase in fitness before it suddenly drops. Then there is a rapid increase in fitness that continues until an annular sort has been performed, and the objects of the world form an annular structure. The first structure that contains all 50 objects is formed after 5080 steps. It is evident from the graph that the annular structure created is very stable. After around 3200 steps there is not much variation in the fitness of the structure. Although the fluctuations increase towards the end of the run they are not significant and may very well be the

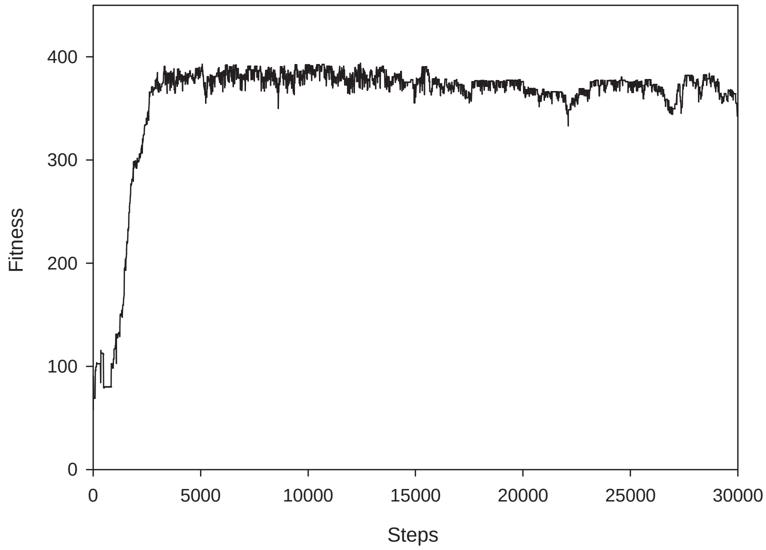

Figure 4.19: Development of fitness during the annular sorting of two types of objects shown in Figure 4.18.

Individual runtime	Average fitness	Standard deviation
5000	362.22	30.63
10000	360.78	30.66
15000	365.33	19.82
20000	361.41	21.47

Table 4.4: Results of 50 runs of the final two-object annular sort solution with different start configurations and run times.

result of misplaced objects due to the random noise added to the ants' perceptions. The low variability in fitness indicates that once the structure is formed it remains, even though the ants are occasionally picking up and depositing objects.

Testing Sensitivity to Initial Conditions

To investigate the stability of the evolved solution and its sensitivity to initial conditions, it was run with 50 different random start configurations. Then the average fitness and standard deviation for these runs were calculated. In addition to investigating the stability of the solution these runs were also used in an attempt to determine approximately how many steps the solution needs to make a good annular structure. To do this we repeated the 50-run evaluations with four different individual run times. The final solution was first run on 50 different start configurations for 5000 steps. Then this was repeated with individual run times of 10000, 15000, and 20000 steps. The results of these runs are given in Table 4.4.

As can be seen in the table the average fitness is always above 360. This is a quite good score since a structure with this score will appear to have an annular structure when inspected visually. The evolved solution is thus capable of forming an annular structure in each of the four run times. In addition the structures formed by the evolved solution appear to be very stable as the score after 20000 steps is about the same as the score after 5000 steps. The standard deviation is lower for the last two run times indicating that when the evolved solution is run for a long time there is less variation in the final formed structures. From these results it is evident that the solution is capable of performing an annular sort on objects that have different initial configurations, and that the sorted structure is stable even though objects are occasionally removed and deposited.

4.5 Three-Object Annular Sorting

The task to be solved in this experiment is the same as in the two-object annular sorting experiment, only now there are three types of objects. This means that a successful solution must have an intermediate band of objects between the central cluster and the outermost band of objects. These objects are called *intermediate objects*. However, when there are the same numbers of each object type, and every object must be part of the structure, it is not possible to form a perfect annular structure. This means that a solution will be successful even if the two outer bands do not surround the structure completely. The fitness function used is the same as in the two-object annular sorting experiment (Equation 4.6). The feedforward network used to control the ants has one more hidden node than the network used in the annular sorting experiment with two types of objects. This means that the network has five hidden nodes, and a total of 27 input nodes with three nodes in each of the nine input groups. Each individual in the evolutionary run has 60 objects that are moved around by a swarm of 6 ants in a 25×25 grid-world. Each individual is run for 4000 steps, and the evolutionary run is 6000 generations long.

4.5.1 The Evolution of a Solution

From Figure 4.20 it can be seen that in the beginning of the evolutionary run there is a rapid increase in both the fitness of the fittest individual of each generation and the average fitness of the generation. This high rate of increase in the fitness ends around generation 200, where the fitness of the individuals begin to increase at a lower rate. After 1600 generations the fittest individual of the generation has a fitness of 373.68. From this point on the fitness of the fittest individuals of the generations appear to remain relatively stable with only minor variations in fitness. These variations are to be expected as the different random start configurations of each generation may present tasks of different complexity. Since each individual is only allowed to step 4000 times it is not certain that the time given to the individuals is enough to form an annular structure from the objects in every situation, if in any at all. The average fitness of the generations shows greater variation than the fitness of the fittest individuals, but neither does it increase much after generation 1600.

To see how a solution to the annular sorting task evolves, the fittest individuals from four generations are selected for a closer examination of how they attempt to solve the task. Since the fitness increases very rapidly in the first generations, the first individual selected is the fittest individual from generation 31 with a fitness of 201.73. Then the fittest individuals from the generations are selected for every 50-unit increase in the fitness. This results in the selection of the fittest individuals from

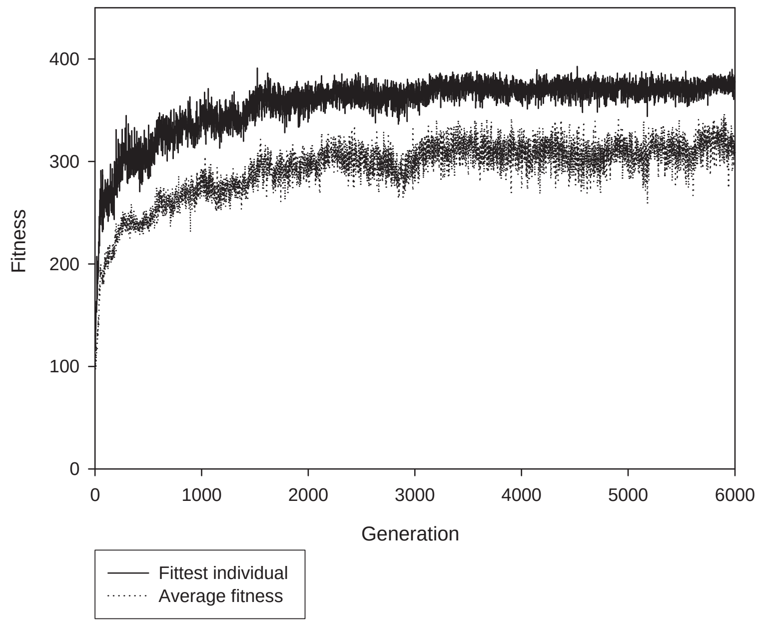

Figure 4.20: Fitness during the evolution of a solution to the three-object annular sorting task.

generations 44, 240, and 1420, with fitness of 250.09, 302.67, and 352.9 respectively. The results of sorts performed by the investigated individuals[5] is shown in Figure 4.21.

The fittest individual from generation 31 starts by forming a cluster containing most of the central objects. As this happens most of the intermediate objects are isolated, but some also form pairs. After the central cluster is formed, it is broken down, only to surface at another location. For the remainder of the sort the central cluster is continuously built and broken down, while the other two types of objects are rarely moved, and consequently remain scattered. The movement of the ants is in straight paths in the forwards direction. When they perceive objects they turn either left or right. This type of movement is evident throughout the remainder of the evolutionary run.

After 44 generations of the evolutionary run, the fittest individual from the generation creates two to three clusters of central objects surrounded by intermediate objects. Some of the outermost objects are also located on the edges of the clusters, but the majority of outermost objects remain scattered. The individual start by forming two or three stable clusters of central objects. These are in contrast to the cluster formed by the previous individual not broken down. After these clusters are formed the ants position the intermediate objects along the edges of the clusters. Finally, some of the outermost objects are also moved to the edges of the cluster, but most remain scattered.

The fittest individual from generation 240 also begins by forming clusters of central objects. After one or two clusters are formed the ants begin to move intermediate objects. These are either moved to the edges of the formed clusters, or they are

[5]A detailed description of the sorting performed by the investigated individuals is given in appendix A.5.

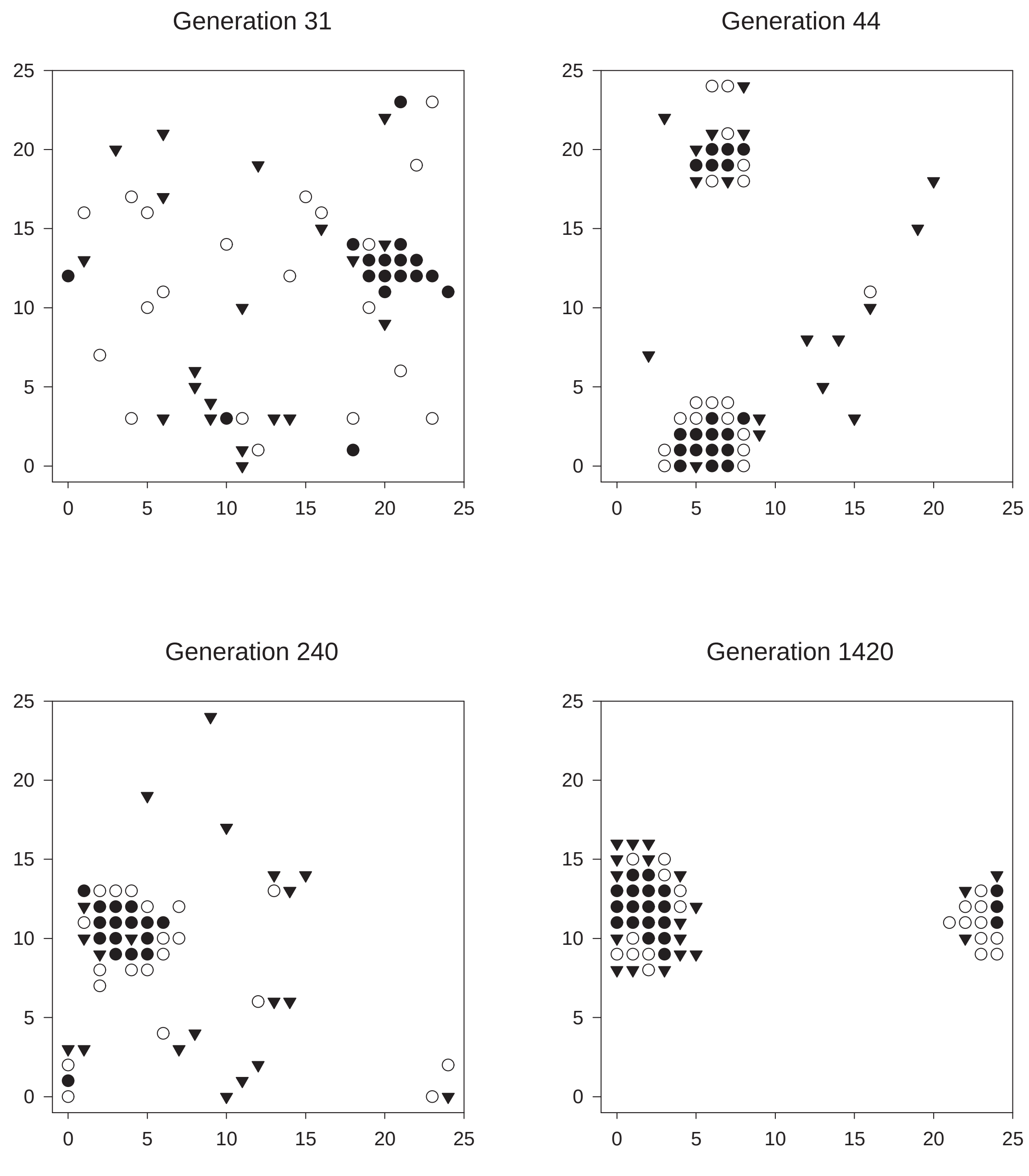

Figure 4.21: Sorts performed by the fittest individuals from generations 31, 44, 240, and 1420. In the figures filled circles are central objects, empty circles are intermediate objects, and triangles are outermost objects.

deposited in small groups that may contain outermost objects. The outermost objects mostly form small groups, but some are also moved to the edges of the clusters. The small groups that are formed are continuously built and broken down, and some of their objects are also dropped in isolated locations. The end result of a sort by the individual is a cluster of central objects, that is surrounded by most of the intermediate objects and a few outermost objects. The objects that are not part of the cluster are either isolated, or located in small groups.

The last investigated individual is the fittest individual from generation 1420. This individual is capable of clustering all the objects into a single cluster. The individual does, like the previously described individuals, start by creating a cluster of central objects. As this cluster is formed most of the intermediate objects are moved to the edges of the cluster, but some also form groups of identical objects. The outermost objects either form groups of only outermost objects, or they are formed around an intermediate object. When most of the intermediate objects have been moved to the edges of the central cluster, the outermost objects also begin to be moved to the edges of the cluster. In the end, all of the intermediate and outermost objects have been moved to the edge of the central cluster. Most of the intermediate objects are located closer to the center of the structure than the outermost objects, and this leads to the formation of annular bands that are somewhat noisy.

From the investigation of the individuals it is clear that individuals that cluster central objects are first evolved. Then individuals that position intermediate and some outermost objects around the central clusters are evolved. As the evolution progresses the evolved individuals first become capable of forming a single cluster of central objects that are surrounded by all of the intermediate objects, and finally forming an annular structure containing all the objects.

4.5.2 The Final Solution

After 6000 generations the final generation has an average fitness of 309.00 with a standard deviation of 48.96. The five fittest individuals of the final generation all have a fitness above 366, and the fitness of these individuals has remained relatively stable for the last 3000 generations. The fittest individual of the final generation, and thus the final solution to the task, has a fitness of 378.38. This score is 94.60% of the maximum achievable fitness.

The final solution is capable of performing a three-object annular sort. The time required for this task is considerably longer than for the two-object annular sorting task, and may vary with different initial configurations of objects to be sorted. Figure 4.22 shows an example of the three-object annular sorting task performed by the final solution. This sort took 14190 steps before the final annular structure was formed, and the annular structure has a fitness of 382.70. The score on the three components making up the fitness of the structure is as follows. The structure has a score of 96.94 on the compactness component. This component thus contributes a total of 193.88 to the total fitness since the component is weighted. The separation component has a score of 96.55, and the shape component has a score of 92.27. As was the case with the two-object annular sort displayed in Figure 4.18, the structure receives a lower score on the shape component than on the other two components.

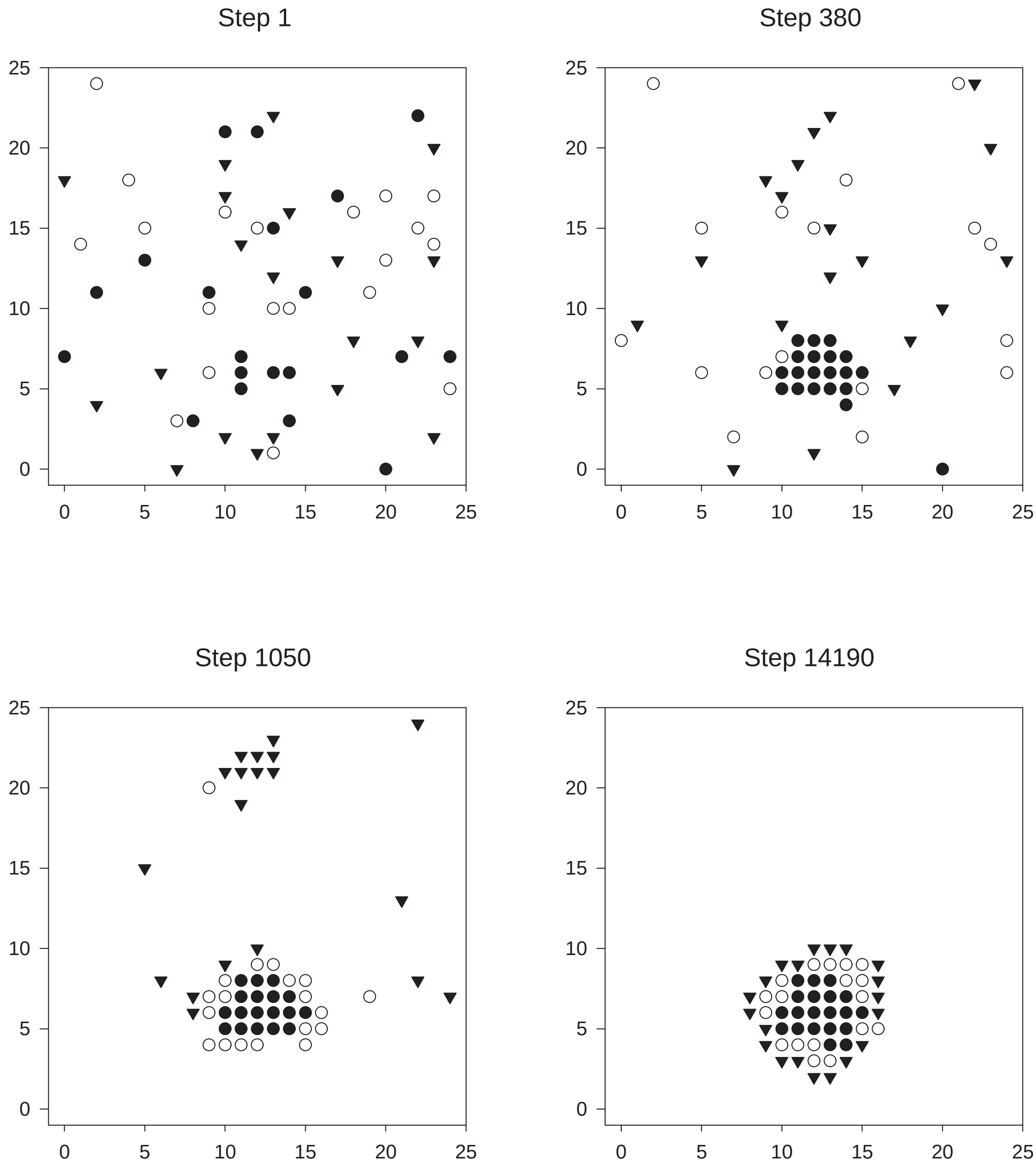

Figure 4.22: A three-object annular sort performed by the final solution. The final structure has a fitness of 382.70, with a score of 96.94 on the compactness component, 96.55 on the separation component, and 92.27 on the shape component. In the figures filled circles are central objects, empty circles are intermediate objects, and triangles are outermost objects.

The collective behavior of the ants of the final solution when performing an annular sort is as follows. At the beginning of the annular sorting run all the ants quickly pick up an object. At this early stage most of the objects are relatively isolated, and the ants pick up any one of the three object types. After around 60 steps the ants mostly carry objects of the central type. The ants carry these objects for a long time, as they appear to require a neighborhood containing several central objects before putting these objects down. As one or two groups of central objects are formed, the ants are not loaded as much of the time as previously, and they also begin to carry the other two object types. This trend continues as a single cluster of the central object type is formed. When all the central objects are located in the cluster the ants begin to put down intermediate objects along the edges of the cluster. The outermost objects are also moved to some extent at this stage, but they are mostly put down in isolated locations. As more and more of the intermediate objects are located along the edges of the cluster, the outermost objects are moved to a larger extent. Some of these objects are still being put down in isolated locations, but the larger part of the objects are put down along the edges of the cluster. At this stage of the formation of the annular structure some of the outermost objects also temporarily form small groups of two or more objects. Eventually all of the outermost objects are moved to the edges of the cluster, and at this time an annular structure has been created.

The formation of an annular structure by the final solution can be summarized as follows. First the ants form a stable cluster of central objects. Then the ants move intermediate objects to the edges of the cluster, before doing the same with the outermost objects. The center of the created structure can be seen as creating an attractive force as objects that become part of the structure are not likely to be removed from the structure. However, some objects are occasionally removed, and the attractive force is thus not absolute. Instead, objects that become part of the structure are unlikely to be removed, and because of this the ants turn to moving the objects that are not yet part of the structure.

The movement of the ants in the final solution resembles the movement that has been observed throughout the evolutionary run. The ants always move forwards, and they turn both left and right with each turn being performed about the same number of times. The ants thus move in straight lines with the direction of movement being influenced by the presence of objects in their eight-cell neighborhood. The presence of objects will cause the ants to turn either left or right before they continue in a straight path.

The progress of the fitness during the three-object annular sorting task shown in Figure 4.22 is displayed in the graph in Figure 4.23. As is evident from the graph the fitness of the evolving structure increases rapidly from the beginning of the sort performed by the ants. This rapid increase continues until the individual has run for 1540 steps. At this point the increase in fitness slows down. After 3600 steps the formed structure has a fitness of 362.70. From this point and until the ants have sorted the objects in 14190 steps the fitness of the structure only goes through minor changes. There are two reasons for the fitness not increasing further during this period of the sorting. First of all, any objects carried by the ants result in a lower fitness of the structure. The difference between the fitness after 3600 steps and 14190 steps is only 20 points, and every object of the structure may be involved in the three components making up the fitness of the structure. If many objects are

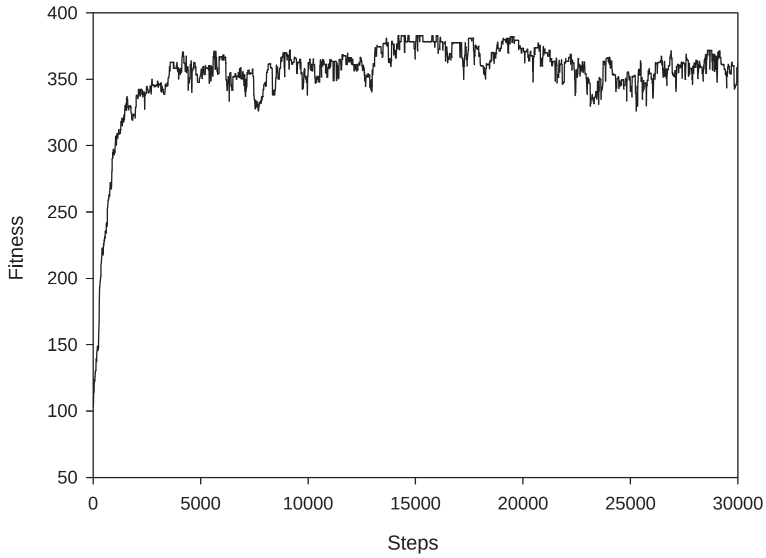

Figure 4.23: Development of fitness during the annular sorting of three types of objects shown in Figure 4.22.

Individual runtime	**Average fitness**	**Standard deviation**
5000	326.99	43.43
10000	329.23	39.09
15000	332.22	37.50
20000	331.65	35.64

Table 4.5: Results of 50 runs of the final three-object annular sort solution with different start configurations and run times.

carried by the ants the fitness of the structure may be lower. Second, as there is such a small difference in the fitness, any misplaced object may have a high impact on the difference. If, for example, an outermost object is located in the center of the structure, then it may take a long time before an unloaded ant walks to this location and picks up the object. It is therefore possible that random misplacements of objects cause the solution to receive a low fitness for a significant number of steps.

Testing Sensitivity to Initial Conditions

To investigate the stability of the final evolved solution and its sensitivity to initial conditions, it was run 50 times on different random start configurations. After these 50 runs were performed the average fitness and standard deviation of the formed structures were calculated. The 50 runs were repeated four times with the individual allowed to run for a different number of steps on each repetition. The results of the four 50-run evaluations is given in Table 4.5. As can be seen in the table the average fitness is approximately the same in all four cases. However, when the runs were repeated a number of times the average fitness achieved when the individual were allowed to run for 5000 steps were considerably lower than for the other three durations. This indicates that, on average, it takes more than 5000 steps to perform an annular sort of 60 objects.

The average fitness of the runs may appear to be low. The reason for this is that in some runs two clusters of the central type objects that are of equal size are formed. When these are formed they are surrounded by a band of intermediate type objects and this structure is not easily broken down. As a result of this there may be two annular structures in the world for a long period before one is finally broken down. The formation of two annular structures is penalized by all three components of the fitness function. The separation and shape components especially penalize the formation of two structures heavily. Because all types of objects are located in both of the structures, there will be large variations in the distances to the assumed center of the structure[6] within each object type. For the shape component this leads to large deviations from the average distance of each object type, in addition to the fact that not all central objects are located in the assumed central cluster (see Equation 4.5). In the separation component the large variability in distances cause the home zones of each type of object to be both wide and to overlap each other. This causes many objects to be considered as infringing on the home zone of another object type. If two annular structures are formed this leads to such a low fitness that they will affect the average fitness considerably.

[6]The center of the structure is the centroid of the largest cluster of central type objects

Chapter 5

Results Analysis and Discussion

The last chapter described the different performed experiments and their evolved solutions. The evolved solutions solve three different types of tasks, yet there are some important similarities among the different solutions. In this chapter we discuss three of these similarities. We also analyze the individual behavior of the ants in the three-object annular sorting solution that enables the swarm of ants to solve the task at the collective level.

5.1 Limited Sensory Capabilities

One of the goals of this work has been to keep the individual agents simple, and the ants have therefore been equipped with a limited sensory apparatus. The ants only have nine perceptual inputs, where the first eight inputs come from the eight cells surrounding the cell where the ant is currently located, and the last sensory input comes from a perception of the object that the ant is possibly carrying (see Figure 3.4). These nine perceptual inputs may at a first glance seem sufficient, but there is one important perception that has not been included in the ants' sensory apparatus. This is a perception of the object located in the same cell as the ant. This perception was not included because we wanted the ants' sensory apparatus to be as simple as possible. Deneubourg et al. [12] have suggested that the sorting of different objects can be done either with a pick up rule or a deposit rule. From this we inferred that it was sufficient for the ant either to perceive the object it was about to pick up, or the object it wanted to deposit. Since it seems natural that the ant will know what it is carrying, we selected a perception of this object. The ants are consequently not able to sense if there is an object in the current cell that they can pick up, and what type this object is.

When an ant is able to pick up an object it can not already carry an object, and therefore the pick up decisions of the ants are only dependent on objects located in the ants' eight surrounding cells. In the clustering task the limitation of the ants not being able to sense the object that they may pick up is not important, because all the objects are of the same type in this task. The ants can simply make a decision of whether an object should be located in their current position or not, depending on the density of surrounding objects. In the tasks of patch sorting and annular sorting the ants' inability to sense the object they are considering whether to pick up or not is more important. This is due to the fact that there is now more than one type of object. Given the same configuration of surrounding objects, the object that is located in the ant's current cell may either be located in a proper position, or in a wrong position, depending on the type of the object. In a patch sorting task, for example, an object that is different from the objects in its eight surrounding cells is located in the wrong position, whereas an object that is identical to its surrounding objects is located in the correct position. However, the ants are unable to sense the type of the object in their current cell, and they are therefore unable to make this decision. The only thing the ants can decide is whether an object located in the current position is located in a wrong position regardless of the type of the object. In the case of patch sorting this means that an object that has many different types of objects as neighbors is located in a wrong position, since all the objects in a successful patch sort must be located in homogeneous neighborhoods.

Because the ants can not sense the type of the object that they are considering picking up, the deposition behavior of the ants becomes the most important behavior with respect to a successful solution of the tasks. When the ants are picking up an object they may only make a decision that holds for the general case. They can only decide if an object located in the considered position would generally be located in a wrong position irrespective of the type of the object. Whereas the ants can only make pick up decisions on general terms they can be specific in their deposition behavior depending on the type of object that they are considering whether to put down. This means that it is only when the ants are depositing an item that they can decide whether an object of a specific type would be in a wrong or correct position if it is located in the current position. In the case of patch sorting this means that it is only when the ant considers whether to deposit an object that it can tell that an object that is different from all the surrounding objects does not belong in the current position. If the ant was considering whether to pick up an object in this situation it would not be able to sense that the considered object was of a different type than the other objects, and so it would leave the object even though it is in a wrong position.

The deposition actions of the ants are even more important in the annular sorting tasks than in the patch sorting tasks. When performing an annular sort all the objects will eventually be located in a single cluster with several types of objects. It is thus extremely difficult for the ants to decide whether an object with many surrounding objects should be picked up or not, even if the objects are of several different types. The deposition behavior thus becomes extremely important in this case because it is so hard to decide if an object should be removed once it becomes part of the growing cluster. The ants must thus only deposit objects in the cluster if they are confident that the carried type of object belongs in their current position. It is clear that the deposition behavior is more significant than the pick up behavior because only the deposition behavior may be influenced by the type of the object that the ant is considering to act on.

The ants' inability to sense the type of the object that they are considering to pick up makes two characteristics of the solutions extremely important in terms of error correction. These two characteristics are the ants' ongoing actions after they have sorted the objects into clusters, and the noise that is added to the ants' perceptions. If, for example, an object is placed in a cluster of a different type of objects then it is clearly in a wrong position, but it will not be picked up by the ants because the surrounding objects are identical. However, because the ants are continuously walking around in the world the ants will continue to walk over the wrongly placed object. Also, because noise is added to the input in 1% of the perceptions, it is probable that the wrongly placed object will eventually be picked up if the sort is allowed to run for enough time. Once the object is picked up the ant may deposit it in the correct position because it is able to sense the type of the object. The occasional picking up of objects regardless of whether they are in the presumably correct position consequently enables the solutions to correct errors that are made during the sorting of the objects.

5.2 Emergent Phases in Sorting

In both of the patch sort solutions, the ants start by sorting one or two of the types of objects. They first create small groups of one type, before all of these groups except one are broken down. The remaining group eventually contains all the objects of the type. After this the ants begin to form small groups of the next type of object, before finally a single cluster of this type is formed as well. This continues until all the object types have been clustered in separate clusters. In the two annular sorting solutions the ants begin by clustering the central type objects. When the central type objects are all part of the same cluster, the ants begin to focus on the objects that are to surround the central cluster. The surrounding objects are picked up by the ants and deposited along the edges of the central cluster, with intermediate objects being moved before the outermost objects.

It is evident that both patch sorts and annular sorts happen in distinct phases. During these phases the ants carry mostly one type of object, and position the objects of this type in the proper locations. Since the emergent phases occur for all the sorting tasks, there must be some kind of commonality in the behavior of the ants of the solutions. This common behavior must be evident in either the pick up or deposit behaviors, or in both of these behaviors. We begin by investigating the pick up behavior. When looking at the sorts while running, only two obvious similarities in the pick up behaviors of the solutions are observed. First of all, when the sorts begin the ants quickly become loaded with any type of object before they turn to focusing on only one type of object. Second, in all the solutions the ants pick up isolated objects when they encounter them in an unloaded state. These two similarities are due to the same characteristic of the pick up behavior, and that is that isolated objects are likely to be picked up by the ants (when a sort begins the ants are unloaded and most objects are isolated).

The ants act based on the sensations they receive from each of their nine perceptual inputs. Depending on what the ants perceive they respond with a corresponding behavior. The controllers of the ants consequently map different combinations of sensory stimuli into appropriate actions. To see if there are some 'hidden' commonalities between the pick up actions of the various sorting solutions, it is possible to look for commonalities in the combinations of sensory stimuli that cause the ants to pick up, or not to pick up, objects when they are unloaded. When looking at these sensory combinations there are no obvious similarities between all of the solutions that can account for the observed phases in the different sorts. There are, however, some important similarities between the solutions to the three-object patch sort and annular sort. First of all, when counting the different combinations of sensory stimuli that either cause the ants to pick up or not to pick up an object, there are a lot more combinations that cause the ants not to pick up an object than to pick it up. This indicates that the ants are restrictive when they are considering whether to pick up an object. The other similarity between the two solutions is evident when looking at the different sensory combinations leading to the two pick up decisions and computing, on average, how many of each type of object that are present in the combinations. The sensory combinations that cause the ants to pick up an object, on average, contain fewer sensations of each type of object than the combinations that lead the ants to not pick up an object. This is what causes the isolated objects to be picked up, because the ants will pick up objects when many of their eight surrounding cells are empty.

It is not surprising that it was not possible to find any similarities in the pick up behavior of the ants that could account for the observed phases in the different sorts. As discussed in the previous section, the deposition behavior is considered to be the most significant when performing the sorts. It is therefore likely that there are similarities in the deposition behaviors of the ants of the different solutions that cause the emergent phases. When all the solutions are run the ants quickly pick up the first isolated object they encounter irrespective of the type of object. However, after only a few steps the ants that are not carrying the objects on which they will focus on first (*primary objects*) deposit their objects. In the annular sort solutions, for example, the ants will first pick up any type of object, but after a short period the ants that did not carry a central type object will usually have deposited their object. The quick deposition of non-primary objects appears to be a common characteristic of all the runs throughout the duration of the runs. During any phase of a sorting run the ants will pick up objects that are not already clustered or part of the emerging sort. However, the objects that are not currently focused on at the particular stage of the sort will quickly be dropped by the ants. For example, in the three-object sorting task this behavior is evident when the ants are focusing on the second type of objects. At this point of the sort they rarely pick up T3 objects as these are mostly part of a single cluster, whereas the ants continously pick up T1 and T2 objects. However, the ants that pick up T1 objects will quickly deposit their object, and the majority of the ants will thus carry T2 objects. In the light of these observations it appears as if the deposition behavior may be responsible for the observed phases of the run.

A similarity between all the sorts is found when investigating the combinations of sensory stimuli that cause the ants to deposit the different objects. This is that there are fewer sensory combinations that cause the ants to deposit their object when they carry primary objects, than when they are carrying the other types of objects. In fact, there is a clear ordering of the number of sensory combinations that cause the ants to drop their objects when carrying each of the different types. There are fewest combinations that cause the ants to deposit the primary objects. For the objects on which the ants focus last, there are the largest numbers of sensory combinations that cause the ants to drop the object. If there is an intermediate stage in which the ants focus on a third type of object, the number of sensory combinations that cause the ants to drop this type of object will be between the numbers for the other two types of objects. The number of sensory combinations that cause the ants to deposit their object when carrying each type of object for all the sorting solutions are summarized in Table 5.1. In the table it can, for example, be seen that in the three-object annular

Two-object patch sort		**Two-object annular sort**	
Type 1	310	Central type	802
Type 2	421	Outermost type	3207
Three-object patch sort		**Three-object annular sort**	
Type 1	2470	Central type	10054
Type 2	1881	Intermediate type	19079
Type 3	7085	Outermost type	35465

Table 5.1: Number of combinations of sensory stimuli that cause the ants to deposit different types of objects in the different sorts.

sort there are 10054 sensory combinations that cause the ants to deposit central objects, whereas for the intermediate and outermost type objects there are 19079 and 35465 combinations, respectively, that cause the ants to drop their objects. There is however one exception to this numbered ordering of sensory combinations. In the three-object patch sort solution there are far more combinations of sensory stimuli that cause the ants to deposit the primary objects than for the remaining two types of objects. However, if the actual combinations are studied it is evident that these are combinations that are rarely encountered, and that will never be found in the world at the beginning of a sort. Examples of such combinations includes situations in which the ants are surrounded by objects, which will not occur at the beginning when the objects are scattered.

To summarize, we have made three different observations that may help to explain the emergence of phases. First, the ants will pick up any type of object, but the objects on which the ants are not currently focusing will be quickly deposited. Second, the numbers of combinations of sensory stimuli that cause the ants to drop their different objects are correlated to the stages in which the ants focus on the different types of objects. There are fewer combinations leading to deposition of primary objects, and then increasing numbers of combinations for the objects on which the ants focus in later stages. Third, the three-object patch sort contains a contradiction to this numbered ordering. In this solution there are far more combinations that cause the ants to deposit the primary objects than there are for the remaining two object types. However, the object structures causing these combinations of sensory stimuli are encountered at the beginning of a sort.

Based on these three observations the following explanation for the occurrence of stages may be given. At the beginning of the sort the ants will pick up any type of object. They will then carry this object around in the world and look for a suitable position at which to deposit the object. However, for the primary objects there are fewer combinations of sensory stimuli, and hence fewer structures of objects in the world, that cause the ants to deposit their object. Because of this the ants will not be able to deposit the primary objects, while at the same time being able to deposit the other objects, as these are deposited in a larger number of situations. Every time the ants pick up a primary object they will thus be unable to deposit their object. The ants consequently become ‘stuck’ with their object. This causes the ants to carry these objects for the most time, thus focusing on these objects, simply because they are unable to deposit this type of object. Then after some time a structure emerges by chance that causes the ants to deposit their objects in the vicinity of this structure.

As most of the primary objects become part of this structure the ants will not pick up these objects. This is because these objects are now located in their proper positions, and objects located in their correct location are not likely to be picked up. Because of this the ants will only pick up the other types of objects. The ants will now focus on the second type of object because it is relatively more difficult to get rid of these objects than the third type of objects. The ants thus have various difficulties with depositing the different types of objects, with the objects focused on earlier in the runs being harder to deposit than the objects focused on later in the runs.

The observed phases in the sorts in which the ants focus on different types of objects are consequently the result of a combination of both the pick up and deposit behaviors of the ants. The deposit behaviors cause the ants to have varying difficulties with depositing the different types of objects, and as a result the ants will at each stage carry the object that they have the greatest difficulty with getting rid of. The deposit behavior thus causes the ants to focus on one of the object types at each stage of the run. The pick up behavior does, on the other hand, limit the types of objects on which the ants can focus. Once all, or most, objects of an object type are located in the correct positions, the ants will no longer pick up these objects. As a result of this the ants will be limited in the types of objects they can pick up and carry, and thus there will be fewer types of objects on which the ants can focus.

5.3 Individual Behavior in the Three-Object Annular Sorting Solution

The different sorting solutions emerge from the combined actions of the ants. The groups of ants are capable of clustering and sorting the objects which are quite complex tasks, yet at the individual level the ants only have a simple set of actions that they can perform. These simple actions of the ants are what lie at the bottom of the different sorting solutions. It is therefore interesting to analyze the behavior of the individual ants to see what kind of individual behavior that gives rise to the collective behavior of the ants as a group. The behavior of the individual ants is controlled by a neural network controller, and it is the actions of this controller that must be analyzed in order to say anything about the behavior of the individual ants. Of the actions that the ants perform it is the pick up and deposition actions that are most central to the sorts that are performed. It is therefore only these two actions that are investigated. The solution to the three-object annular sort is the most interesting solution, and we have therefore decided to investigate the neural network controller of the ants in the evolved solution to this task.

There are several ways of investigating the neural network controller. We have chosen to investigate the controller using two different methods that look at the controller from different perspectives. Both of the methods give the same kind of result. What the controller is doing is to take perceptions from the ants and convert these perceptions into actions. The perceptions of the ants are observations of what type of object is located in each of the ant's perceptual fields. What the controller does is thus to look at the configuration of different types of objects in the ant's eight-cell neighborhood and its carried object, and then decide on which action it is appropriate to take. The results of the two methods are a numbered evaluation of how much each type of object located in each of the perceptual fields biases the controller towards taking the action in question. A central type object located directly in front of the ant may, for example, bias the ant more strongly towards depositing its currently carried object than a central type object that is located behind the ant. The two methods look at different aspects of the controllers in their analysis. The first method uses the different combinations of sensory stimuli causing the various actions as the basis for its analysis, whereas the other method looks at the weights of the neural network controller. If both methods yield similar results then the results are likely to be reliable. The method that looks at the sensory combinations also shows clearly how empty perceptual fields influence the behaviors.

The first method looks at the different combinations of sensory stimuli that cause the ants to perform the action that is to be investigated. This is done by first calculating all the different combinations of sensory stimuli in which the ant is in the appropriate state and decides on the investigated action. For example in the case of pick up behavior, all combinations of sensory stimuli that are possible when the ant is not carrying an object and that leads to a pick up action are extracted. The combinations of sensory stimuli are investigated by looking at the percentages of the combinations of each sensory stimulus in each of the perceptual fields. The higher this percentage is, the more likely it is that the corresponding sensory stimulus will bias the ant into taking the considered action. If for example a sensory stimulus corresponding to a central type object has a high percentage of occurrence in perceptual field two, which is directly in front of the ant, then the ant will be biased towards taking the investigated action when it encounters a central type object directly in front of itself.

The second method looks at the weights of the neural network controller. The neural network is a fully connected feedforward network. This means that each input to the network is given as input to each of the nodes in the hidden layer, whose values are given to each of the nodes in the output layer. Each node in the output layer corresponds to a specific action, and each input to the network will contribute in some way toward firing each of these nodes. What this method of investigation does is assign a value between 0 and 100 to each possible input to the network. This value indicates how much each input contributes to the firing of a specific output node, and hence to a specific action. Each input to the network corresponds to the perception of a specific type of object in a specific perceptual field, and the assigned values consequently indicate how much each type of object in each of the ant's perceptual fields bias the ant toward taking the considered action. This is also what was done in the other method, although this method inferred the bias directly from the combinations of sensory stimuli leading to the different actions. How much each input to the network contributes to the firing of a specific output node is determined as follows. If the network is viewed as a graph then there are several paths between each input node and the output nodes. For each pair of input node and output node it is possible to go via each of the hidden nodes. Each of these paths is weighted differently. The raw score of each input is calculated by summing the products of the edges of each path. This is illustrated in Figure 5.1.

In the case of the controller for the three-object annular sort there are five hidden nodes. This means that for each input there are five different paths that each consist of two edges. There is one edge between the input and the hidden node, and one

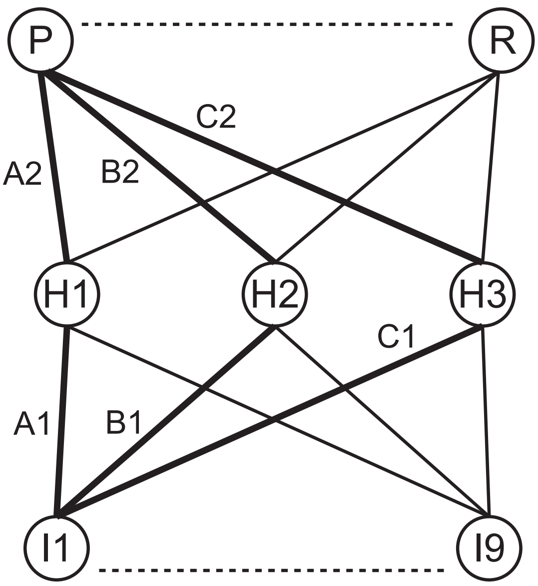

Figure 5.1: To calculate how much node I1 contributes to the firing of node P, the edges of each path between the two are multiplied together, and then the results are summed. In this case node I1 contributes to the firing of node P with $A1 \times A2 + B1 \times B2 + C1 \times C2$.

between the hidden node and the output node. The weights of each pair of edges are multiplied before these products are added together. The raw score of each input consequently looks at how each input propagates through the network, and assigns a score showing how it influences the final outcome of the investigated output node, which corresponds to a specific action. After the raw scores have been calculated they are normalized to a value between 0 and 100 as follows. First the lowest score is set equal to 0 by adding the distance between this number and zero to all the values (the lowest score is negative). Then each score is divided by the largest score and multiplied by 100. By doing this each score will be normalized in the range $[0 \ldots 100]$.

5.3.1 Pickup Behavior

When looking at the pick up behavior of the ants during three-object annular sorting it is evident that the pick up behavior serves both a disruptive and a conserving role. The ants pick up objects that are generally in the wrong positions, and hence disrupt the existing distribution of objects. However, the behavior also conserves existing local distributions of objects that have a correct structure by not removing objects from these local areas. As discussed in section 5.2 the pick up behavior also contributes to the sorts occuring in phases by limiting the types of objects that the ants pick up, and consequently move around.

We start by looking at the weights of the network that influence the pick up action of the ants. The normalized scores obtained from the appropriate weights are given in Table 5.2. The table gives the scores for each type of object occuring in each of the ant's eight surrounding cells (the numbering of the cells is shown in Figure 5.2). There are no entries for objects carried by the ant because when the ant picks up an object it can not currently carry an object, and hence the input from this perceptual

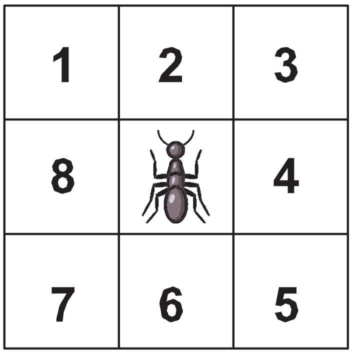

Figure 5.2: Locations of the ant's perceptual fields.

	Central	**Intermediate**	**Outermost**
1	65.07	**86.99**	72.98
2	0.00	36.27	49.07
3	45.28	78.86	**100.00**
4	40.87	66.14	**89.50**
5	7.70	53.96	39.03
6	**97.55**	67.07	78.44
7	16.48	69.35	44.97
8	37.78	48.36	60.69

Table 5.2: Pick up weight scores for each type of object in perceptual fields 1 through 8.

field will always be zero when the ant is considering whether to pick up an object or not. The most obvious feature of the scores is that in all of the perceptual fields, except field 6, the score achieved by the presence of a central type object in the field is lower than the scores for the presence of the other two types. This means that if the ant perceives central type objects around itself that are not located directly behind it (perceptual field 6) the ant is not likely to pick up an object. The presence of central type objects thus inhibits the ant from picking up objects. This behavior is clearly appropriate because when there are many central type objects in the ant's surroundings it is located in, or near, a cluster of central type objects. A cluster of central type objects is one of the features of the desired structure, so when the ant senses the presence of this cluster it should not remove objects because they are likely to be located in their correct position. Another result of the scores being higher for the presence of intermediate and outermost objects is that if there are many of these present in the ant's surroundings then they will be picked up. This is appropriate because there should be no dense groups of either, or both, intermediate or outermost type objects. The presence of intermediate type objects bias the ants most towards picking up objects if they are located in perceptual fields 1 and 3. Outermost type objects bias the ants most towards picking up objects if they are present in perceptual fields 3, 4, and 6.

An analysis of the combinations of sensory stimuli shows how the presence of empty cells around the ant affects the pick up behavior of the ant in addition to the presence of various types of objects. Table 5.3 shows for each combination of sensory

	1	2	3	4	5	6	7	8
Nothing	19.79	**52.64**	26.42	22.25	**48.32**	17.41	29.73	**38.31**
Central type	30.86	9.33	18.31	22.59	6.31	**60.94**	1.41	29.84
Intermediate type	30.14	14.33	17.73	20.30	10.30	7.44	**44.58**	23.07
Outermost type	19.20	23.71	37.54	34.86	35.06	14.20	24.28	8.79

Table 5.3: Percentage of occurrence of each combination of stimulus and perceptual field in the sensory stimuli that cause the ant to pick up an object.

stimulus and perceptual field, the percentage of times this combination is present in the combinations of sensory stimuli that cause the ant to pick up an object. A central type object is for example located directly in front (position 2) of the ant in only 9.33% of the situations where the ant will pick up an object. From Table 5.3 it is clear that the ant is likely to pick up objects if there are empty cells surrounding it. The absence of objects in the perceptual fields 2, 5, and 8 bias the ant particularly towards picking up objects. For the different types of objects the ant will be biased towards picking up objects if they are present in the following perceptual fields. Central type objects will bias the ant towards picking up objects if they are located in perceptual fields 1, 6, and 8. Intermediate type objects will bias the ant towards picking up objects if they are present in perceptual fields 1 and 7. And finally, outermost type objects will bias the ant towards picking up objects if they are present in perceptual fields 3, 4, and 5.

The results of the two different methods of analysis can be combined to draw the following conclusions about the pick up behavior of the ant. First, the ant will pick up objects if there are many empty cells surrounding the ant, particularly if these are directly in front of the ant, directly to the left of the ant, or back and to the right of the ant. Second, the ant is not likely to pick up objects if there are many central type objects in the ant's surroundings. Central type objects located directly in front of the ant or to the sides behind the ant will inhibit pick up behavior most. However, if there is a central type object located directly behind the ant then the ant will be biased towards picking up an object. Third, the ant will pick up objects if there are several intermediate or outermost type objects in the ant's surroundings. These are the most clearly observed features of the ant's pick up behavior.

5.3.2 Deposit Behavior

The deposit behavior of the ants is considered to be the most important behavior in terms of being able to solve the different sorting tasks successfully. This is because it is only when the ants are depositing objects that they can make a decision of whether an object of a specific type belongs in a specific location or not. In the three-object annular sort there are three types of objects present in the environment. Because of this the deposition behavior can be seen as consisting of three different deposit actions. There is one action for each type of object. For each type of object the ant is carrying it must be able to decide whether the object should be deposited or not, depending on the presence of other types of objects in its immediate surroundings.

Even though the deposition behavior is considered to consist of three different actions depending on the type of object that the ant is currently carrying, the weights

	Central	Intermediate	Outermost
1	46.80	50.03	57.79
2	**84.55**	62.66	68.81
3	41.68	29.57	24.07
4	56.10	50.04	42.83
5	**100.00**	36.91	57.07
6	28.69	50.59	36.07
7	42.92	47.99	**96.06**
8	35.61	58.45	48.15
A	0.00	21.33	39.91

Table 5.4: Deposit weight scores for inputs from the ant's eight surrounding cells (1–8) in addition to the object the ant is carrying (A).

of the network can only be analyzed for all three actions together. The result of the analysis of the network controller for the deposition behavior of the ant is given in Table 5.4. In contrast to the analysis of the pick up behavior, the input from the object that the ant is carrying is also included in this analysis in addition to the perceptions from the cells surrounding the ant. In fact, the scores for the different types of object that the ant is carrying are very important for the deposition behavior. This is because one of these scores will always be 'active' when the ant is carrying an object. These scores consequently function as a weighted baseline that determines how large the other scores must be for the ant to engage in deposit behavior. When the ant is carrying a central type object the score for the perception of the carried object is 0, and hence this must be counter-balanced by the scores for the other perceptual fields being higher in order for the ant to engage in deposit behavior. When the ant is carrying an outermost object the baseline is highest, and when carrying an intermediate object the baseline is halfway between the other two baselines. Another way to see this is that it takes more in order for the ant to deposit its object when it is carrying a central type object than when it is carrying intermediate and outermost objects, respectively. This correlates with the reason for the observed phases discussed in section 5.2. The ants have greater difficulties with depositing central type objects because of the low baseline, and hence focus on these objects first. Then the ants focus on intermediate and outermost objects because these have gradually higher baselines.

Since the deposition behavior can be said to consist of three different actions depending on which type of object the ant is carrying these are analyzed in turn below.

Deposition of Central Type Objects

The central type objects of an annular sort form a cluster in the center of the structure. This means that the ants should only deposit central type objects in areas that contain other central type objects, and possibly intermediate objects. There are in other words a restricted number of situations in which the ants should deposit their object when they are carrying a central type object.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Nothing	23.95	9.65	27.74	12.74	12.28	29.67	14.68	25.10
Central type	29.50	**51.18**	25.77	**56.82**	**64.89**	14.60	30.38	23.74
Intermediate type	21.37	25.07	21.40	19.74	5.95	19.16	26.69	26.65
Outermost type	25.17	14.09	25.08	10.69	16.88	36.57	28.26	24.51

Table 5.5: Percentage of occurrence of object types in each of the ant's surrounding cells when depositing a central type object.

When analyzing the weights of the network with respect to the deposit behavior, it is seen that when an ant is carrying a central type object the score for the perceptual field of the ant's carried object will always be zero. This means that when carrying a central type object the ant is initially biased towards not depositing its object. In order for the ant to deposit the object there must either be many other objects located in the ant's surroundings, or objects must be present in perceptual fields that yield high scores. The higher the total sum of scores the more biased the ant becomes towards depositing its object. When looking at the weights it is seen that the scores for central type objects are particularly high when these are present in the perceptual fields 2, 4 and 5 (see Table 5.4). This means the ant is likely to put down its object if there are central type objects located directly in front of the ant, directly to the right of the ant, and behind and to the right of the ant. Apart from this, the main conclusion that can be drawn from the weight analysis is that it is less likely that the ant will deposit a central type object than one of the other types of objects given the same configuration of surrounding objects.

The analysis of the combinations of sensory stimuli that cause the ant to deposit its central type object sheds more light on the ant's deposit behavior when carrying central type objects. The results of the analysis are shown in Table 5.5. When there are empty cells around the ant it is not likely to deposit its central type object. None of the cells surrounding the ant is empty in more than 30% of the situations where the ant will deposit a central type object. The perceptual fields 2, 4, 5, and 7 have particularly low occurences of empty cells. This means that if there are no objects in front of the ant, to the right of the ant, or behind the ant on either side, then the ant is not likely to deposit its object. This observation seems logical since the ant should not deposit central type objects in isolated areas. The ant is most likely to deposit its object if there are central type objects located in its perceptual fields 2, 4, or 5. The ant will, in other words, be likely to put down a central type object if there are central type objects directly in front of the ant, or to the right of the ant. When it comes to the presence of intermediate and outermost type objects, intermediate objects are mostly present directly in front, or behind and to the left of the ant in the situations where the ant deposits a central type object. The outermost type objects are mostly present directly behind the ant when the ant deposits a central type object. However, they are also present either in front and to the side of the ant, or behind and to the side of the ant.

If the two methods of analysis are combined the following observations can be made concerning the ant's deposition of central type objects. First, there are fewer situations in which the ant deposits a central type object than the other two object types. Second, the ant is most likely to deposit a central type object if there are

	1	2	3	4	5	6	7	8
Nothing	24.90	14.75	28.95	17.61	18.55	29.15	19.38	30.22
Central type	22.99	**40.02**	17.12	**45.24**	**59.99**	13.02	22.38	15.69
Intermediate type	22.85	26.95	21.59	22.34	6.92	18.83	26.31	27.65
Outermost type	29.25	18.28	**32.34**	14.81	14.53	**39.00**	31.93	26.43

Table 5.6: Percentage of occurrence of object types in each of the ant's surrounding cells when depositing an intermediate type object.

central type objects located in front of the ant, or to the right of the ant. Third, the ant is likely to deposit its central type object if there are intermediate objects on the ant's left side. And finally, the ant will be biased towards depositing its object if there is an outermost object directly behind the ant.

Deposition of Intermediate Type Objects

When the ants have created a cluster of central type objects they deposit intermediate type objects along the edges of this cluster. This means that the intermediate type objects are mostly put down next to central type objects in the final structure. The analysis of the weights of the network first of all shows that when carrying an intermediate type object, the ant is more biased towards depositing its object than when carrying a central object, but less biased towards depositing the object than when carrying an outermost object (see Table 5.4). The ant consequently deposits an intermediate object more easily than a central object. Because the weight score for the occurrence of central type objects is so high when they are located directly in front of the ant and behind to the right, it is likely that the ant will deposit its intermediate type object if central type objects are encountered in these positions. The intermediate type objects receive the highest weight scores if they are present in the perceptual fields 1, 2, 4, 6, 7 and 8. This means that the occurrence of intermediate type objects in all of the ant's surrounding cells, except for the cells to the right and in front or behind the ant, will bias the ant towards depositing its object.

It is easier to see what influences the deposition of intermediate objects when looking at the analysis of the combinations of sensory stimuli that cause the ants to deposit their intermediate type object. The results of this analysis is given in Table 5.6. In similarity with the deposit of central type objects, the intermediate type objects are usually not deposited if there are empty cells around the ant. However, the percentage of occurrence of empty cells is generally higher than was the case for the deposition of central type objects. Central type objects are located either directly in front or to the right of the ant, or behind and to the right of the ant, in most of the situations where the ant deposit an intermediate type object. When it comes to other intermediate type objects, they do not occur very often in any of the ant's surrounding cells. This means that the ant will not deposit intermediate type objects in groups of the same type. When intermediate type objects do occur in the ant's surroundings when depositing an intermediate object, they are mostly located in front and to the sides of the ant, and behind to the left. Outermost objects occur more frequently than intermediate objects in the situations where the ant deposits its intermediate object. These objects are either located on the sides in front of the ant, or behind and to the left of the ant.

	1	2	3	4	5	6	7	8
Nothing	24.62	18.30	28.22	21.49	24.53	28.63	22.16	29.04
Central type	22.48	**33.24**	18.35	**35.35**	**43.34**	15.71	21.70	17.04
Intermediate type	23.83	26.87	22.76	24.13	12.44	21.16	25.59	27.38
Outermost type	29.07	21.59	**30.67**	19.04	19.69	**34.50**	**30.55**	26.54

Table 5.7: Percentage of occurrence of object types in each of the ant's surrounding cells when depositing an outermost type object.

Based on the analysis of the sensory stimuli it is possible to determine in which situations the ant is most likely to deposit an intermediate type object. The deposition of intermediate type objects seems to be particularly influenced by the occurrence of central and outermost type objects in specific locations. All of these combinations of object type and position in relation to the ant occur in more than 25% of the situations in which the ant deposits its intermediate type object. The ant is most biased towards depositing its intermediate type object if there are central type objects located in the perceptual fields 2, 4, and 5, and outermost type objects located in the perceptual fields 1, 3, 6, 7, and 8. This means that the ant for example will deposit its object if there is a central type object in front of the ant, or to the right, and outermost type objects behind and to the left of the ant. The ant will thus deposit its intermediate type object between the central and outermost objects, which is where the band of intermediate objects should be located in the annular structure.

Deposition of Outermost Type Objects

The outermost type objects form a band surrounding the intermediate type band of objects in a three-object annular structure. This means that all the outermost objects are located between a layer of intermediate objects and empty cells. The objects should generally not be located next to central type objects. When looking at the analysis of the weights of the network in Table 5.4, it is evident that when the ant is carrying an outermost object it is biased towards depositing its object irrespective of the presence of other objects in its surroundings. The outermost type objects are therefore the most easily deposited since the ant have an initial bias towards depositing this type of object. This bias is caused by the score of 39.91 from a perception of an outermost object carried by the ant. Outermost objects located behind and to the left of the ant give the highest scores in the weight analysis of the outermost type object scores. This is followed by outermost objects located in front, in front and to the left, and behind and to the right of the ant. The most important information that may be inferred from the analysis of the weights of the network with respect to the depositing of outermost type objects is that these objects are deposited most easily of the three types of objects.

The results of the analysis of the combinations of sensory stimuli that cause the ant to deposit an outermost type object is given in Table 5.7. When comparing the results with the results in tables 5.5 and 5.6, it is clear that there is a gradual increase in the percentage of combinations where fields 2, 4, 5, and 7 are empty when the ant deposits an outermost object, compared to when depositing the other two types of objects. However, even though outermost objects will be located next to empty cells in the annular structure, it is evident that the ant prefers to deposit the outermost objects next to other objects instead of empty cells. Central objects are in similarity with the deposition of intermediate objects most often encountered in the perceptual fields 2, 4, and 5 when the ant deposits its object. However, central type objects are not present in as many of the sensory stimuli combinations that cause the ant to deposit outermost objects as for the other two object types. This is not surprising as the outermost objects should generally not be deposited next to central type objects. Of the ant's perceptual fields, intermediate type objects are encountered most often in fields 2, 4, 7, and 8 when the ant deposits an outermost object. In fact, the intermediate objects are frequently located in all of the ant's surrounding cells, except the cell behind on the right, when the ant deposits an outermost object. Other outermost objects are most frequently present in the perceptual fields 3, 6, and 7, and least frequently present in fields 4 and 5 when the ant deposits an outermost object. This means that the ant will deposit outermost objects next to other outermost objects if these are located in front and to the right, or behind and to the left of the ant.

When looking at what combinations of objects in the perceptual fields of the ant that bias the ant most towards depositing an outermost object, a somewhat surprising fact is discovered. The ant is most biased towards depositing its outermost objects if there are central objects in the perceptual fields 2, 4, and 5, and outermost objects in the fields 3, 6, 7, and to a lesser degree 8. Intermediate objects are also frequently present in fields 2, 7, and 8 when the ant deposits an outermost object. The surprising finding is that the presence of central objects bias the ant towards depositing its outermost object, because the outermost objects should not be positioned next to central type objects. However, when the ants carry outermost objects the central objects are located in a single cluster surrounded by intermediate objects. The ants will consequently not be able to deposit their outermost object when sensing central type objects. The weight genes responsible for this behavior may therefore have a low selection pressure, and as a consequence simply drifted to high values. From these observations it is clear that the ant will deposit outermost objects if outermost objects are located behind and to the left of the ant. Intermediate objects will bias the ant towards depositing the outermost object when encountered in all positions except for behind to the right of the ant.

5.3.3 Key Mechanisms in the Annular Sort

The previous sections have given a detailed description of the pick up and deposition actions of the ants. Based on these descriptions and the observations of annular sorts in progress, it is possible to summarize the key mechanisms that lead to the formation of an annular sort: 1) Objects are picked up if they are isolated, or there are several outermost objects surrounding the object. 2) Objects are not picked up if they are located in a neighborhood that is densely populated with central objects. 3) Outermost objects are most easily deposited, followed by intermediate objects. Central objects are deposited in fewer situations than the other to object types. 4) Central objects are deposited when there are central objects in front or to the right of the ant. 5) Intermediate objects are deposited when there are central objects in front and to the right of the ant, or outermost objects behind and to the left of the ant. 6) Outermost objects are deposited when there are other outermost objects behind and to the left of the ant. They are also deposited when there are intermediate objects in the ant's surroundings.

The key mechanisms described above are involved in the formation of an annular sort in the following way. At the beginning of the sort most objects are isolated, and they are therefore picked up by the ants. Intermediate and outermost objects are more easily deposited than central objects. This leads to some of these being put down next to other objects, whereas the ants carrying central objects are unable to deposit their objects. The ants will carry the central objects until there is a random fluctuation in the world that leads to a small group of central objects being formed. Because central objects are only deposited next to groups of central objects, all of the central objects will eventually be deposited in this growing group.

Since objects are not picked up when they are in a neighborhood densely populated with central objects, the objects in the emerging group will not be picked up once they become part of the cluster. This leads to the ants only picking up intermediate and outermost objects once all the central objects have become part of the cluster. Outermost objects are more easily deposited than intermediate objects, and the ants therefore carry intermediate objects most of the time. The intermediate objects are deposited when there are central objects in front and to the right of the ant, and as a consequence they are deposited along the edges of the central type cluster. Once they are located on the edge of the cluster they are not picked up because they are in a neighborhood with several central objects. All of the intermediate objects eventually become part of the growing structure, because once they are part of it, they are not removed. This leads to the outermost objects being the only objects that are picked up by the ants. These objects are most easily deposited when there are intermediate objects in the ant's surroundings. The only place where there at this point are intermediate objects is on the edge of the growing structure, and the outermost objects are consequently moved to the edges of the structure. Once they are part of the structure they are not likely to be removed. This is because objects are most easily picked up when they are isolated or only have outermost objects in their surroundings. The presence of intermediate objects consequently inhibits the pick up of outermost objects from the edge of the formed structure. Because objects are generally likely to be deposited in the structure, but not removed from the structure, the emerging structure will eventually contain all the objects. At this point the annular sort is complete, and the emerged structure will have an annular pattern.

5.4 Self-Organization in the Evolved Solutions

Self-organization has been defined by Bonabeau et al. [6] as: "*... a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. The rules specifying the interactions among the system's constituent units are executed on the basis of purely local information, without reference to the global pattern, which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence.*" From this definition of self-organization it seems likely that self-organization is present in the evolved solutions to the clustering and sorting tasks. The evolved solutions create structures on the global level, in the form of sorted distributions of objects, and this is caused by the interactions of the swarm of ants. The ants have only access to local information when deciding on which action to perform, and they have no knowledge of the global pattern that signifies a successful solution to the task that they are solving. It is evident that the properties of the evolved solutions closely resemble the definition of self-organization. Based on this closeness it is likely that the evolved solutions are self-organizing systems where pattern formation occurs through interactions internal to the system, without intervention by external directing influences [9]. If the evolved solutions are in fact self-organizing systems then they must have four specific ingredients, which are positive feedback, negative feedback, amplification of fluctuations, and multiple interactions [4, 6].

The first ingredient that exists in self-organizing systems is positive feedback, or amplification, which promotes the creation of structures. Positive feedback is present in all of the evolved solutions in the form that growth of groups of objects encourages further growth of the groups. This is most evident in the clustering solution. During a run of this solution there is at some point formed a group of objects. Once this is formed the ants will quickly deposit the remaining objects that are scattered around the world in this growing group. Positive feedback is thus involved because as a group of objects grows, its attractive force on the other objects in the environment increases. In other words, the larger the cluster the more likely the ants are to deposit their object in this cluster. This form of positive feedback is also present in the sorting solutions. In the patch sort solutions a group of identical objects will encourage the ants to deposit objects of the same type in this group. As the different groups of objects grow, their attractive force on the other objects of the same type increase. In the annular sort solutions, positive feedback is also present, although its occurrence is slightly different because the growing structure will contain several types of objects. However, the growth of the structure still encourages the ants to deposit their object in the structure, and so there is a positive feedback mechanism involved. Positive feedback is consequently a key ingredient in all of the evolved solutions.

The second ingredient of self-organization is negative feedback. Negative feedback counterbalances the positive feedback and helps to stabilize the collective pattern. It may take the form of saturation, exhaustion or competition, and in the evolved solutions it takes the form of exhaustion of objects. In all the solutions the growing structures will continue to grow until there are no more objects that are not already part of the structures. In the clustering solution the ants will continuously move objects to the growing structure, until there are no more objects in the environment that are not already located in the structure. If the 'building blocks' of the structure were never exhausted the ants would continue to move objects to the cluster, and it would increase in size indefinitely. In the two patch sort solutions the various groups of identical objects also stop growing when negative feedback kicks in. The different homogeneous groups that are formed in these solutions will continue to grow until there are no more objects of the same type present in the environment. Negative feedback is also present in the annular sorting solutions. In these solutions the ants will stop moving one type of object when there are no more objects of this type that are not already part of the emerging structure. The carrying and deposition of one type of object is thus stopped when this object type is exhausted, because there are no more of these objects that are not already in their correct positions. Negative feedback is strongly involved in creating the emergent phases that are observed in the sorting solutions (see section 5.2). Negative feedback forces the ants to start working on other types of objects when an object type is exhausted. An object type is exhausted when all of its objects are located in positions from which the ants do not pick up objects, for example in a cluster of central type objects.

Self-organization relies on the amplification of fluctuations, and this is the third ingredient of self-organizing systems. When all of the evolved solutions are run the objects are at first scattered randomly in the environment. After some time, one or more of the objects are deposited next to another object, or in a small group, such that there is a fluctuation in the random distribution of the objects. Once this small group is formed, it quickly increases in size. The initial fluctuation is consequently amplified through the growth of the created group. Because it is the amplification of fluctuations that causes a group of objects to be formed, the global pattern is rarely the same in two different runs of the solution. The fluctuations that are amplified rarely happen in the same place of the environment, and thus the formed structures will be located at different positions of the world in the different runs of the solutions.

The final ingredient of self-organization is multiple interactions. This is clearly present in all of the evolved solutions as they all contain a swarm of ants that create the various structures. In addition to interacting with each other, the individual agents in a self-organizing system should be able to make use of the results of their own and other agent's activities [4]. This is also the case for all of the evolved solutions as the ants make use of the emerging structures independently of whether it is themselves or other ants that have created the structures. When there are multiple interactions among the agents in the system, they clearly must interact in some way, either directly or indirectly. Section 1.3 introduced stigmergy, and this is the way in which the ants in the evolved solutions interact. The sensory apparatus of the ants does not enable them to sense each other, only the objects in the world. The ants can thus not interact directly, but only indirectly through the positioning of the objects in the environment. Each of the ants modifies the environment by picking up and depositing objects, and the other ants respond to the changed state of the environment at a later time. The ant also responds to the changes that it itself made to the environment.

In addition to the four ingredients that must be present in self-organizing systems, there are some key properties that usually characterize self-organizing phenomena. These three properties are the creation of spatiotemporal structures in an initially homogenous medium, the possible coexistence of several stable states (multistability), and the existence of bifurcations [3, 4].

The two first properties are characteristics of all the evolved solutions. Spatiotemporal structures are formed in an initially homogeneous medium. At the beginning of a clustering or sorting task there is no grouping of objects of any type, if this does not happen by chance, and the objects are scattered randomly in the environment. The environment can therefore be said to be a homogeneous medium because there is a random distribution of objects. As the different solutions are run on the initial distributions of objects, structures eventually emerge. In the clustering task identical objects are initially scattered in the environment, for thereafter to be clustered by the evolved solution. This also happens in the annular sorting tasks, but in these there are several types of objects that are initially scattered, and the final cluster has a specific internal structure. In the patch sorting tasks there are initially several types of objects scattered in the world, and when the solutions are done the objects have been clustered in groups containing only one type of object.

Multistability is also a property of the evolved solutions. The structures emerge by amplification of random deviations, and any such deviation may be amplified. The systems can thus converge to any one of several possible configurations, depending on the initial distribution of objects and the fluctuations that occur. For the different evolved solutions this means that when they are run, the location of the clusters may be in any one of a variety of possible locations. Depending on where in the environment the initial groups of objects are formed, the final structures will be located in different positions of the world. There are thus several possible final states of the evolved solutions because each time they are run, the emerging structures may form at many different locations of the environment depending on the initial positioning of objects and ants.

Bifurcations when parameters are varied have not been observed in any of the evolved solutions because it has not been investigated. It is however possible that bifurcations may occur if the ratio between the number of ants and objects are varied.

From the discussion above it is clear that the evolved solutions are all self-organizing systems. The solutions all have the key ingredients of a self-organizing system, and they have two of the three typical properties. This shows that artificial evolution can be used successfully to create such systems. The fitness of the individual behaviors is evaluated by looking at the global pattern formed through the interactions of a group of agents with the same behavior. It is thus possible to evolve the individual behaviors by specifying characteristics that should be present at the global level. This is a promising alternative to the hand-coding of individual behavior, because when modeling self-organizing systems the global patterns are known, whereas the individual behaviors leading to the patterns are unknown. It is therefore favorable to start by defining properties of the known patterns, rather than attempting to define the rules governing the unknown individual behavior.

Chapter 6

Conclusion

This thesis has presented our work with evolving swarms of homogeneous agents capable of clustering and sorting objects. Our work has been inspired by social insect societies, and in particular ant colonies where ants cluster dead bodies outside the nest, or sort their brood either in separate piles or in a pattern of concentric rings. In these societies the individuals are simple, yet at the collective level of the colony complex patterns emerge. Based on these observations we have designed agents with limited sensory capabilities that can pick up and deposit objects as well as move around. The behavior of the agents has been evolved by evaluating the patterns that are formed at the global level when swarms of agents interact with different types of objects. With this as a basis we have been able to evolve swarms of agents that can perform clustering, patch sorting, or annular sorting of simple objects. Earlier research dealing with the clustering and sorting seen in insects has been successful in creating swarms of agents that can perform clustering and patch sorting. However, there has up to this date only been limited success with creating swarms capable of forming an annular pattern. The swarm of agents and the collection of simple objects can be seen as a self-organizing system as described in section 5.4. The agents interact with each other and the objects at the individual level, and at the collective level of the system a pattern in the distribution of objects emerges. Thus, we have created self-organizing systems in which clusters, patches and annular patterns emerge.

One of our main motivations for performing this work has been to show that artificial evolution is a viable approach to creating swarms of agents capable of clustering and sorting. The power of artificial evolution in finding solutions to problems has been shown in a wide range of different research fields, but its application to the creation of swarms of agents has been limited. Within the field of research concerned with creating swarms of agents capable of clustering and sorting, artificial evolution has rarely been used at all. The normal approach within this field of research has been to observe the individual behavior of insects and then form a set of rules, or a mathematical model, that describes the individual behavior. After having defined the individual behavior, the system is run in a simulation and evaluated in terms of whether the desired emergent patterns occur or not. If the desired patterns do not occur, the individual behavior of the agents is modified and the simulation is repeated. The traditional approach to the creation of swarms capable of clustering and sorting is consequently to hand-code the individual behavior and then tweak this until it leads to the desired patterns. Our approach is in a way the opposite of this approach. We define measurable characteristics of the desired patterns, and then evolve the individual behavior by evaluating it in terms of the patterns that emerge. Thus, instead of starting by defining the individual behavior we start by defining properties of the desired patterns. This is a favorable approach because the characteristics of the desired patterns are known in advance, whereas the behavior of the individuals is unknown. By being able to create a solution to such a complex task as that of annular sorting we have shown that this approach is viable, and that artificial evolution also is a viable approach within this field of research.

Another of our main motivations has been to show that simple individual behavior can lead to systems that are capable of solving complex tasks. Previous research has shown that swarms of agents with simple behaviors are capable of tasks such as clustering and sorting objects into separate clusters. However, it has not been shown earlier that swarms of individuals with simple behaviors are capable of solving a task as complex as annular sorting. Annular sorting is considered to be a much more difficult task to accomplish than clustering and patch sorting. By evolving swarms that are capable of annular sorting we have shown that swarms of agents with simple sensory apparatus and behavior are able to solve complex tasks. The formation of annular clusters by a swarm of robots has previously been described by Wilson et al. [33]. The annular sort performed by the group of robots that they describe is however much less compact, and the different types of objects are not as clearly separated as in the solutions we have created.

Our second motivation above is closely connected to one of the main motivations within the field of swarm intelligence. This is to gain a better understanding of nature. By replicating phenomena found in nature in computer simulations a better understanding of their underlying mechanisms may be gained. In our work we have shown what is needed at the individual level in order for the tasks to be solved at the global level. By doing this we have also, by omission, shown what does not need to be inferred about the observed phenomena. First of all, we have shown that for a swarm of agents to solve the given tasks there is only a limited amount of information that is needed by each individual. The agents in our work only have information about the objects that are in their immediate neighborhood. Thus, the only information the agents have about the distribution of objects in the environment is which objects are surrounding them. Consequently, at the individual level, there is no need for knowing anything about the overall distribution of objects, or the distances to the closest objects at each side of the agent. Also the agents do not need to know what they pick up, only what they put down. This further underlines the point that very limited information is needed at the local level in order for the tasks to be solved at the global level of the system.

The simplicity of the individual agents that are needed in order to solve the tasks is further emphasized by the use of simple feedforward networks as controllers for the agents. These networks only map input patterns to output patterns, and do not retain any knowledge of previous activity. The successful use of this type of network consequently indicates that the individuals do not need an explicit knowledge of their previous actions. By using only information from their current neighborhood and a feedforward network to control their behavior, the individual agents do not need any form of internal memory to solve the different tasks [12, 18, 24].

The clustering and sorting tasks have been solved by agents that only have access to limited local information about the distribution of objects, and that are capable of only simple behaviors. This indicates that we do not need to infer anything else about the natural systems that show the same emergent patterns. It is, for example, not necessary to attribute the annular brood sorting to templates either in the form of pheromones or mechanical structures [15].

The most important discovery during our work is the emergent phases that occur in the sorting experiments. In both the patch sorting and the annular sorting experiments the ants sort the objects in phases. The ants first position one type of object in the correct position according to the desired pattern, then the next type and so on. The occurrence of the emergent phases in all of the experiments indicates that there must be some advantage with this phenomenon. When considering an annular sort the occurrence of emergent phases seems logical. It does not make sense to start positioning of objects further out in the annular structure before the inner parts are formed. Before the central cluster is formed it is not possible to create a band surrounding this cluster and so on. In an annular structure it therefore seems plausible that the sorting of objects should occur in phases, where objects closer to the center of the structure will be positioned before objects further out in the structure. This sorting in phases presents an alternative to the two phases that have been recently observed in the *Leptothorax* ant by Sendova-Franks et al. [32]. They have observed that when migrating to a new nest, the ants will first create a dense cluster of brood before they spread out the cluster. It is only when the brood are spread out that the annular structure of brood is formed. The sorting of objects in distinct phases presents an alternative to this observed clustering and spreading out technique. Even though the sorting in phases seems logical in the context of annular sorting, its significance in patch sorting is less apparent. It is possible that the sorting in phases is an efficient way of organizing the work to be done, but this needs to be investigated further. It is however clear that there must be some adaptive significance to the organization of sorting in distinct phases since this is present in all four of the sorting experiments.

The first motivation of swarm intelligence, to achieve a better understanding of nature, was described above. The second motivation does, in a sense, work in the opposite direction. This motivation is that the study and application of characteristics of social insects to computer science may lead to better techniques. Algorithms based on social insects have, for example, lead to improved solutions to optimization, scheduling, and data analysis problems [13, 18]. The collective behavior of social insects has attractive properties such as robustness, scalability, and adaptability. The solutions to the tasks described in this thesis are both reliable and robust. All of the evolved solutions are capable of forming the desired patterns from a wide variety of initial distributions of objects and agents. The solutions are consequently both reliable and adaptable in that they successfully form the desired patterns in a wide variety of situations. Another attractive feature of social insects is robustness through redundancy. In all of the evolved solutions the desired patterns will still be formed even if some of the agents are removed. The solutions are thus robust in that they will still solve the tasks successfully even if some of the agents stop functioning.

6.1 Future Work

The work in this thesis can both be improved and extended. A problem with the work that is performed is the dependency of the neural network controllers on the number of different types of objects in the tasks to be solved. The number of different objects determines both the number of input nodes and the number of hidden nodes in the neural networks. An improvement to the achieved solutions would therefore be to remove these dependencies. The easiest improvement is for the networks to separate different object types quantitatively instead of qualitatively. In the current form the networks separate different object types by associating each type of object with specific input nodes. The structure of the network must consequently be altered according to the different types of objects present in a task. If the networks separated object types quantitatively instead, this alteration of the structure would not be necessary. A way to do this is for the network to receive different valued inputs depending on the type of object present. It is more difficult to remove the dependency between the hidden nodes and the types of objects. The number of hidden nodes is related to the dimensionality of the problem, and the more types of objects present, the higher this dimensionality will be. It is consequently difficult to remove the dependency between the types of objects and the number of hidden nodes, though a possible solution might be to use a different type of network than a feedforward network. If the two mentioned dependencies were removed then the evolved solutions might be applicable to a greater number of tasks. That is, the same swarm of agents might, for example, be able to perform annular sorts of both two and three types of objects.

A possible extension to the work described here is to allow the agents to handle complex objects that are more or less similar, instead of simple objects that are either identical or different. One way to do this is to make the objects arrays of properties, and then define a similarity metric to calculate a real-valued similarity between objects. The network controller can be modified to handle this extension by changing the inputs to the network to be the result of a similarity measure between the perceived object in the corresponding perceptual field, and the object carried by the ant. If this modification is made, it is possible to extend our work to areas such as sorting and visualizing data, and classification of cases.

By letting the arrays represent elements in a database it is possible to sort or visualize the data in the database. Objects in a database often have a high dimensionality (many properties) and are therefore difficult to visualize. Thus, the sorting by ant-like agents is attractive because they group the data in two dimensions, and consequently perform a sort of topographic mapping [18]. Lumer and Faieta did, for example, extend their algorithm to visualize commonalities in banking data [5]. By extending our research, complex data may be visualized either in the form of separate clusters, or in the form of an annular structure.

If the objects are made to represent different cases, it is possible to extend our research to classification tasks. The evolved agents continue with their sorting activities even after all of the objects have been assigned to the correct clusters. A possible future line of research is therefore to investigate what happens if new objects are introduced after the initial collection of objects is sorted. Will the introduced objects, for example, only be positioned in one of the existing clusters, or do they cause a reorganization of objects and new clusters to form? By sorting the initial collection of objects, and then observe where new objects are positioned, it is possible to extend the research to classification tasks.

A different direction of future research is to investigate whether it is possible to implement the annular sorting behavior in swarms of physical robots. Previous robotic experiments have only managed to produce a loose cluster with a separation that is inferior to the one produced by our evolved agents. It would therefore be interesting to see if robots equipped with approximately the same sensory apparatus as our agents, could be evolved to create annular structures of the same quality that was observed in our experiments.

Finally, even though we have identified the key mechanisms involved in an annular sort by our agents (see section 5.3.3), it is not likely that these are the ones that are responsible for the brood sorting performed by ants. This is because we have not replicated the exact characteristics of the biological systems, but rather formed a system that loosely models some of the properties of the natural systems. However, we have shown that the evolution of individual behavior is a promising alternative to the previous hand-coding of behavior. It might therefore be possible to discover the underlying biological mechanisms by making a model that is biologically plausible, and then evolve the behavior of the agents in this model. If the model is successful, then the mechanisms involved in the sorting in the model may very well be the same mechanisms that exist in nature, or at least provide a better understanding of these mechanisms.

6.2 Concluding Remarks

When we started our work it was from an inspiration of social insects and how they are able to solve complex tasks in spite of their individual simplicity. Each individual insect is capable only of a limited set of simple actions, yet the insect societies as a whole are capable of solving complex tasks that exceed the abilities of each individual insect. We found it fascinating that systems modeling the characteristics of social insect societies could be implemented and applied successfully to problems in computer science. These systems are able to exploit the promising features of insect societies, such as flexibility, error tolerance, distributed control, and indirect communication.

As we explored the field of swarm intelligence we were surprised to find that most of the research to this date has used hand-coded rules to control the individual agents in the developed systems. The problems that are addressed by swarm intelligence are characterized by a knowledge of the properties of the final solution, whereas the individual behavior of the agents that is responsible for the emergence of the solution is unknown. It therefore seems strange that the norm so far has been to attempt to capture the unknown mechanisms at the individual level without exploiting what is known about the final solution. As we have mentioned, this has been done by tweaking the individual behavior until the desired solutions emerge. The characteristics of the problems addressed by swarm intelligence suggests that an evolutionary strategy might be appropriate, and a promising alternative, to create the behavior of the individual agents. By implementing numerical measures of the properties of the final solutions in a fitness function, the individual behaviors of the agents in a swarm can be evolved such that they lead to the emergence of the desired solutions. With this approach the knowledge about the final solution is exploited to create the unknown individual behavior. In our work we implemented numerical measures of properties of different patterns that insects create. These patterns gives rise to what is referred to within computer science as clustering, patch sorting, and annular sorting.

By ultimately solving the annular sorting task, which is a well known problem within the artificial intelligence community, we have further contributed to the claim that ant-based systems can solve complex problems. We have also shown a promising alternative to the traditional approach of hand-coding a set of rules controlling the agents in these systems. This is the evolution of the individual behavior with a genetic algorithm, and the use of alternative controllers for the agents, such as a neural network. However, even though we have identified a promising alternative to the traditional approach of programming swarm-intelligent systems, the task of programming such systems is still difficult. This is because of the inherent property of the systems whereby solutions emerge due to interactions both between the individuals in the system, and between the individuals and their environment. Before evolving the behaviors of the individual agents one must therefore not only know what types of behaviors it must be possible to control, but also what types of interactions that are needed between the elements of the system for the solution to emerge.

Swarm intelligence is a young field of research that is yet at a theoretical level [13], and there has been few practical applications of the theories developed within the field. However, as the theoretical research continues, the understanding of the theory within the field increases, and this will hopefully eventually lead to the creation of algorithms that are practically applicable. Even though the field is still at a theoretical stage, some of the research shows promising results. This includes work within ant colony optimization [13], which is the largest and most developed research area within swarm intelligence, and the clustering algorithm created by Handl et al. [18]. Through extensive comparisons with standard clustering algorithms, Handl et al. [18] have shown that their algorithm has a better performance than the standard techniques on some sets of data.

Even though research within the field of swarm intelligence has not yet found many practical applications, we believe that the field has a promising future. The ant-based algorithms have many attractive features, such as adaptivity to changing conditions, error tolerance, distributed control, scalability, and indirect communication. These features will become more important in the future as the problems to be solved increase in complexity. An ant-based system that sorts large collections of data can, for example, be implemented on relatively simple hardware due to low demands on processing power, and can easily be scaled to different sized data collections by simply varying the number of agents in the system. Ideas from swarm intelligence may also find new applications in newer research fields, such as nanorobotics. Research within this field attempts to create microscopic robots that manipulate elements at the molecular level. Because of the size of the robots they must be simple, and when they operate in groups communication is an issue. Swarm intelligence is highly appropriate to the design of such systems as it deals with swarms of relatively simple agents capable of solving complex tasks through indirect interaction. Swarm intelligence thus has a promising future both in the development of the current theoretical research field, and in the application to new research fields.

Appendix A

Descriptions of Investigated Individuals

A.1 Clustering

Generation 3

The fittest individual from generation 3 achieved a fitness of 102 during the evolutionary run, and when it is run it clusters around two thirds of the objects into several small, loose clusters while the remaining third of the objects are scattered randomly in the world. At the beginning of a run of the individual, most of the objects are located in isolated positions. During this stage of the run the ants frequently pick up objects, for thereafter to drop them in the next step. This causes a sensation of drift of the objects. The objects appear to glide around in the world, although their movement is mediated by the ants. When the objects are moved one step, their new location may be either an isolated position or next to other objects. If the object is located in a densely populated neighborhood then it is less likely to be picked up by the ants. The ants consequently move mostly the isolated objects in the world. These objects will eventually be deposited next to other objects, and combined with the ants' preferential pickup of objects this causes a formation of groups of objects. When the world has reached a state containing several groups of objects, the objects are not moved as much as before, although the ants never cease to move objects completely. The continuous movement by the ants causes some of the groups to be broken up and others to be formed. Some objects are also occasionally deposited in isolated positions. The result of the clustering by this individual is a dynamic state of the world where about two thirds of the objects are part of groups, whereas the remaining third are isolated.

The movement of the ants is the same regardless of whether the ant is carrying an object or not. The ants only move in the backwards direction, and they usually move in a straight path. If there are no objects present in the ant's eight-cell neighborhood, then the ant will move in a straight path backwards. However, when there are objects present in the ant's neighborhood the ant will sometimes turn left, but never right. The movement of the ants is thus in straight paths backwards with an occasional turning to the left.

Generation 6

The movement of the ants of the fittest individual of generation 6 is identical to the movement of the ants of the fittest individual from generation 3. The sorting behavior does however differ significantly. This individual is capable of clustering all the objects into three to five loose clusters. There are thus no isolated objects, and the groups of objects are larger than the groups formed by the previous individual. Soon after the beginning of a run of the individual, all of the ants become loaded with an object. At this stage of the run most of the objects are isolated, and the ants pick up the first object they encounter. Once the ants have picked up an object they carry this object for some time. It is not until they encounter a group of objects that they deposit their object, and since there are not many such groups at the beginning of the run, the ants carry their objects for some time during the early stages of the run. After a while several groups of objects emerge. Some of these groups are quickly broken down, while other groups increase in size. Once these groups emerge it does not take long until all the objects in the world are part of one of the groups. As this happens the ants are not loaded as much of the time as they were earlier. The formed groups are loose in their structure and very fragile, and as time goes by all but a few of the groups are broken down. The end result of a run of the individual is a dynamic state consisting of a few loose clusters. In the dynamic state objects are continuously removed from the clusters for thereafter to be deposited in the same, or another, cluster. The loose character of the clusters cause parts of them to occasionally break off from the cluster leaving isolated objects or small groups next to the larger clusters. These broken off objects are, however, quickly moved to one of the clusters again.

Generation 13

The fittest individual from generation 13 is capable of forming one or two dense clusters of objects. When two clusters are formed, one is usually much smaller than the other. If the individual is allowed to run for a long time the smaller cluster is in most cases eventually broken down, and the individual is capable of placing all of the objects in only one dense cluster. The movement of the ants is the same as the movement of the ants of the previously described individuals with one exception. The ants of this individual also turn right in addition to left.

A run of the individual that leads to the formation of one or two clusters happens in the following way. At the beginning of a run most of the objects are isolated, and the ants usually pick up the first object that they encounter. They then carry these objects around until they encounter a group of objects. As objects are mostly isolated at the beginning of the run the ants carry objects for the longest time at this stage of the run. After a short time several groups of objects emerge. These groups are first loose in their structure, but as time goes by they become more densely packed with objects. While most objects become part of a group at this time there are still some objects that remain isolated in the world. When most of the objects have become part of groups the ants are not loaded as much of the time anymore. In fact, the fewer and larger groups of objects there are, the less time the ants spend carrying objects. This is because the ants do not usually pick up objects from neighborhoods that are densely populated with objects. The clustering process thus seems to slow down as fewer groups of objects are formed. The dense groups that were formed by the ants soon contain all the objects in the world. After this happens two or three groups increase in size while the other groups are broken down. As time goes by one or two of these groups are also broken down leading to a final distribution of objects in one or two clusters. If the individual is allowed to run for a sufficient amount of time the end result is in all cases the formation of a single dense cluster.

A.2 Two-Object Patch Sorting

Generation 77

The ants of the fittest individual from generation 77 are capable of forming three or more loose groups of type 1 (T1) objects. The type 2 (T2) objects are, on the other hand, only put in groups of two or three, or they are isolated. The collective behavior of the ants that leads to these formations is as follows. At the beginning the objects are scattered in the world, and the ants soon encounter an object that they pick up. This leads to most of the ants quickly becoming loaded with either of the two types of objects. However, the T2 objects are deposited quicker than the T1 objects, and this result in the majority of the ants being loaded with T1 objects. After some time most of the ants are carrying T1 objects, with the remaining ants being unloaded. At this point an unloaded ant may occasionally pick up a T2 object, but this is quickly deposited. When the individual has run from between 1000 to 2000 steps, a few small groups of T1 objects emerge. These groups do initially only contain two or three objects, but as the run of the individual continues they may increase in size. In fact, if they do not increase in size they are rapidly broken down by the ants. This causes the formation of a few larger groups of T1 objects. As these groups arise the ants will carry T2 objects more of the time. These objects are mostly put down next to other T2 objects, leading to the formation of several pairs of T2 objects. As the run continues all of the T1 objects eventually become part of three or more large groups of T1 objects. When these groups grow the ants are not loaded as much of the time as they were previously, and they also carry T2 objects more than earlier. As a result of the increased carrying of T2 objects, these objects are grouped in several small groups of two or three. However, even though most of the T2 objects eventually become part of a small group, several of them remain isolated in the world. This positioning of T2 objects together with the few large groups of T1 objects constitutes the final 'patch sort' performed by this individual.

The movement of the ants is only in the backwards direction, and much of the time they move in circles that are sometimes displaced. The movement of the ants is identical irrespective of whether an ant is loaded or not, and with what type of object it is loaded with. The circular movement is caused by the ants only moving backwards and turning right at each step. The circles the ants move in are displaced by forward and backward movement occasionally being active at the same time, causing the ants to remain stationary for one step. Also the ants sometimes remain stationary for extended periods when there are certain configurations of objects in their neighborhood.

Generation 158

The movement of the ants of the fittest individual from generation 158 is in some respects similar to the movement of the ants of the fittest individual of generation 77. The ants only move in the backward direction, and when they are unloaded or carry a T2 object they move in circles. However, the behavior of the ants differs markedly when they are carrying a T1 object. If an ant is carrying a T1 object it moves in a straight path backwards, and when there are objects present in its neighborhood it will turn right. However, irrespective of what type of object the ants are carrying, there are still some configurations of objects in their environment that cause the ants to remain immobile.

At the beginning of the run of the fittest individual from generation 158, some of the ants quickly become loaded with objects of type 1. None of the other ants pick up objects of type 2, but rather remain unloaded. After a few hundred steps a few small groups of T1 objects emerge. When these emerge some of the ants do occasionally pick up T2 objects, but these are quickly deposited by the ants. Most of the ants are thus either loaded with a T1 object or they are unloaded. As the run continues some of the T1 groups are broken down whereas others increase in size. This eventually leads to the formation of two or three large groups of T1 objects. As these groups emerge more and more of the ants become unloaded, rather than picking up T2 objects that are still randomly dispersed in the world. When all of the T1 objects have become part of one of the large groups there is not much change in the distribution of objects in the world. Both types of objects are occasionally picked up, and the T1 objects are quickly deposited in one of the large groups. The T2 objects, on the other hand, are either put down next to other T2 objects or in isolated locations. This leads to the emergence of a few groups of T2 objects. These groups are, however, easily broken down, such that the final distribution of T2 objects is in a few small, unstable groups with the rest of the objects scattered around the world.

Generation 550

The movement of the ants of the fittest individual from generation 550 has many similarities with the movement of the ants of the fittest individual from generation 158. When the ants are unloaded they move in circles by moving backwards and turning right at each step. When they are carrying a T1 object they move in straight paths backwards, and turn right when there are objects present in their eight-cell neighborhood. However, the behavior of the ants when carrying T2 objects differs significantly. For the first time the ants move in the forward direction. When an ant picks up a T2 object it moves in circles by moving forward and turning right at each step. The ants now have three different characteristic patterns of movement depending on whether they are carrying an object or not, and what type of object they are carrying.

The fittest individual from generation 550 is, like the previous two investigated individuals, capable of forming large groups of T1 objects. In addition to this it is also capable of forming larger groups of T2 objects, which was not done by the previously described individuals. The collective behavior of the ants that results in a rough type of patch sorting is as follows. At the beginning of a run all of the ants quickly become loaded with either type of object. These objects are then put down next to other objects of the same type. However, it appears that after an ant has put down an object next to a similar object, it picks it up again in the next step if it remains stationary. That is, objects are put down next to objects of the same type, but objects that are only located next to a similar object are also picked up by the ants. As a result of this there is a period where pairs of similar objects are continuously formed and broken up. After some period where this type of behavior is dominant a larger group of T1 objects emerges. Shortly after this happens a group of T2 objects also emerges. The group of T1 objects then quickly increases in size. Although the group of T2 objects also grows, its rate of growth is nowhere near the growth of the T1 object group. At the same time as these two groups of objects increase in size other groups of objects are also formed. The other T1 objects are typically grouped in one or two large groups, whereas the other T2 objects begin to form a few small groups. The T1 objects eventually form one or two dense groups of objects. The T2 objects, on the other hand, are not grouped to the same extent as the T1 objects. The T2 objects form three or more smaller groups that are not as dense as the T1 object groups. This is the final sorting that is performed by this individual.

Generation 720

The fittest individual from generation 720 is capable of performing an almost perfect patch sort of two types of objects. The individual is capable of forming a single dense cluster of T2 objects, but it creates one large and one smaller cluster of T1 objects. The collective behavior of the ants that leads to the formation of these clusters is as follows. At the beginning of a run, all of the ants quickly become loaded with either type of object. These objects are then deposited next to objects of the same type. This leads to the formation of several small groups of objects of the same type. As the run continues some of these groups are broken down, whereas other groups increase in size. When this occurs the ants become unloaded more of the time as more and more of the objects become part of larger groups of identical objects. At this point there are several groups of each type of object, and each consists of between four and eight objects. The T2 groups appear to be less stable than the T1 groups, and as a result these are broken down. The objects from these groups are deposited in the largest group of T2 objects causing this group to increase in size until it contains all the objects of type 2. The groups of T1 objects are also broken down as time goes by, but at a much slower rate than the T2 groups. A square group of four T1 objects seems to be particularly stable, and as a result of this a few groups with this shape are formed, while the remaining objects of type 1 form two large groups. As the run of the individual comes to an end there are typically one or two clusters of each type, although there is more often a single cluster of T2 objects than of T1 objects.

The movement of the ants of the fittest individual from generation 720 differs from the movement of the ants of the fittest individual from generation 550 when they are unloaded or carrying a T2 object. When the ants are carrying a T1 object the behavior is the same. The default behavior of the ants is to move in a straight path backwards unless there are objects present in its eight-cell neighborhood. If there are objects in the neighborhood then the ant will turn right. This type of behavior is now also present when the ants are unloaded, or when they are carrying a T2 object. When the ants are unloaded they exhibit the exact same behavior as when they are carrying a T1 object. The ants move in a straight path backwards, but when they perceive one or more objects in their environment they turn left. When the ants carry a T2 object the movement is in the opposite direction. The ants move in straight paths forward, and when there are objects located in their environment they turn right. There are now also fewer configurations of objects in the ants' immediate environment that cause the ants to remain immobile for extended periods of time.

A.3 Three-Object Patch Sorting

Generation 87

The fittest individual from generation 87 is capable of clustering only type 3 (T3) objects of the three object types in the world. Some of the type 2 (T2) objects are grouped in small groups, whereas the type 1 (T1) objects remain scattered randomly in the world, although the ants do occasionally pick them up. The collective behavior of the ants during a run of the individual is as follows. At the beginning of the run some of the ants pick up an object. The ants may pick up any of the object types, although they carry T3 objects for the most time at this stage of the run. The objects that are picked up are quickly deposited by the ants. Throughout the run of the individual only around half of the ants are loaded with objects. At some times most of the ants are loaded, but at other times the majority of the ants are not carrying any objects. After a short period of time the ants have deposited some of the T3 objects in small groups. These groups are, however, very fragile and are soon broken up by the ants. After this some of the T3 and T2 objects are collected in pairs of the same type. As the run continues a larger cluster of T3 objects emerges. When this cluster reaches a size of around ten objects it appears to have reached a stable size as it is not easily broken down. Even so, objects are occasionally removed from this cluster and deposited either in isolated locations or in small groups. At the same time the T2 objects are continously deposited in small groups or in isolated locations. The small groups of T2 objects are extremely fragile and are broken down and rebuilt at a steady rate. As the run of the individual continues all of the T3 objects do eventually become part of a single large cluster. The T2 objects are continuously deposited in and picked up from small groups or isolated locations, whereas the T1 objects remain scattered throughout the world. The ants of this individual thus appear to be concerned mainly with T2 and T3 objects, since the T1 objects remain randomly dispersed in the world. Despite this, the ants do actually pick up each type of object. But, the T1 objects that are picked up are deposited at once by the ants such that it appears that they are not being carried by the ants at all.

The movement of the ants of the fittest individual from generation 87 is only in the forwards direction. If the ants are unloaded or carrying a T2 or T3 object they show the same kind of movement. The ants will move in a straight path forwards if there are no objects present in their eight-cell neighborhood, but if they perceive any objects they will turn either left or right. If the ant is carrying a T1 object the behavior is significantly different. When carrying a T1 object the ants will move forward and turn left at each step, thereby moving in a circle. If there happens to be any objects present in the vicinity of this circular path, then the ant may displace the path either by not turning or by turning right. However, the movement behavior of ants carrying T1 objects only occur under artificial conditions because the ants will deposit their T1 object at once, and thereby never travel any distance while carrying a T1 object.

Generation 253

The ants of the fittest individual from generation 253 are loaded more of the time than the ants of the fittest individual from generation 87. At the beginning of a run of the individual the ants quickly become loaded with any of the three types of objects. However, T1 and T2 objects are deposited quicker than T3 objects, and this causes the majority of the ants to carry T3 objects after a short period of time. The T1 and T2 objects that are put down by the ants at this time are deposited next to similar objects, leading to a few groups of two or three objects of each of these types. After some time a small group of T3 objects emerges. This group then grows in size until it eventually contains all the objects of this type. As more and more of the T3 objects become part of this group they are not carried as much by the ants any more. This leads both to a larger portion of the ants being unloaded, but also to the ants carrying more of T1 and T2 objects. These objects are initially deposited in pairs. When all of the T3 objects are part of the large cluster, the ants that carry objects carry T2 objects for the most part. The T1 objects that are picked up by the ants are deposited quickly, and are put down both in isolated positions and next to other identical objects in pairs. When all the T3 objects are grouped in a single cluster the ants continually create and break down loose groups of T2 objects. Because these groups are so fragile they never reach a considerable size, and the largest groups of this type of object take the form of loose chains instead of dense clusters. However, in contrast to the T1 objects it is clear that the T2 objects are clustered to some degree. Through the remainder of the run the T2 objects are continuously clustered in loose groups that are later broken up, while the T1 objects occasionally form pairs but for the most part remain isolated and randomly distributed in the environment.

The movement of the ants of the fittest individual from generation 253 is identical to the movement of the ants described previously. Irrespective of what the ants may carry they display the same kind of movement. The ants will move in a straight path forward in the absence of objects in their eight-cell neighborhood. However, if they do perceive any objects in their surroundings they will turn either left or right.

Generation 366

The ants of the fittest individual from generation 366 move in straight paths and turn when they perceive objects. The direction of turning does however appear to differ slightly depending on what type of object the ants are carrying. If the ants are carrying a T2 object they will turn more often to the right, but if they are carrying a T3 object they will turn mostly to the left. The T1 objects are carried for such short periods by the ants that it is not possible to describe any movement behavior of the ants while carrying these objects.

The ants of the fittest individual from generation 366 are able to sort the T2 and T3 objects into one large cluster each. However, the ants do not carry T1 objects for very long, and as a result of this they remain scattered in the world. Soon after the beginning of a run of the individual the ants pick up an object. Since the ants carry T1 objects for only short distances, the ants that become loaded mostly carry T2 and T3 objects. The T3 objects that are carried by the ants are put down in areas where there are objects of the same type. Because of this several small groups of T3 objects emerge after some time. T2 objects, on the other hand, are deposited both in the vicinity of other T2 objects and in isolated locations. There is thus less grouping of this type of object. A short period after the groups of T2 objects and T3 objects are formed, one or two of each type of group increase in size while the other groups are broken down. Whether there are one or two groups that increase in size seems to depend on whether the two groups both have reached a critical size of more than five objects. Once the groups begin to increase in size they grow rapidly, and soon they contain all the objects of their respective types. When this happens the ants are mostly unloaded. This is due to the fact that the ants are not likely to pick up T2 and T3 objects that are located in neighborhoods that are densely populated with objects of the same type. However, objects are occasionally removed from the clusters for thereafter to be deposited in the same cluster, or if there are two clusters of the same type, in the other cluster. This occasional removal and deposition of objects already contained in a cluster will, in the case where there are two clusters of the same type, eventually lead to one cluster decreasing in size while the other cluster increases in size. After varying periods of time the growing cluster will eventually ‘absorb’ all the objects of the type, and the other cluster vanishes. However in some cases this does take a significant period of time. The end result is the formation of one cluster each of T2 and T3 objects, while the T1 objects are randomly scattered throughout the world.

Generation 3273

The fittest individual from generation 3273 is capable of forming one or two loose clusters of T1 objects in addition to one cluster each of T2 and T3 objects. For the first time of the evolutionary run the ants are now carrying T1 objects for a significant amount of time. As a result of this it is for the first time possible to describe the movement of the ants when carrying a T1 object. The movement in this situation is identical to the unloaded ants and ants carrying T2 or T3 objects. The ants will move in a straight path forwards in desolate areas, but when they perceive objects in their eight-cell neighborhood they turn either left or right.

A run of the individual that leads to a nearly perfect patch sort is as follows. Soon after the beginning of a run of the individual all of the ants become loaded with objects. The ants then deposit these objects in the vicinity of identical objects. After a short period of time all of the ants are carrying either a T2 or T3 object, and after another few steps most of the ants are carrying a T3 object. It appears that the reason for this is that the T3 objects are only deposited in neighborhoods with a high density of objects of the same type. T2 objects do not require as many identical objects in their vicinity in order to be put down by the ants, and the T1 objects require the least identical objects in their environment to be put down. Ants that pick up T3 objects are thus stuck with their objects for a longer period of time than ants that pick up T2 or T1 objects. Groups of each type of objects emerge after a few hundred steps. T3 objects are grouped in two or three groups that contain all the objects of this type. The groups of T1 and T2 objects are on the other hand smaller and do not contain all the objects of the type. As time goes by all the T3 objects become part of one or two dense clusters of this type. At the same time the T2 objects form a few loose groups, with the T1 objects being either isolated or located in groups of objects that are barely connected. As the run continues the T2 objects are eventually clustered together in one large cluster. If there were formed two clusters of T3 objects earlier in the run, one of these will be broken down with the other cluster eventually containing all the T3 objects. The T1 objects are grouped into one or two clusters, but these clusters are very loose in their shape and they are only barely connected. In the final distribution of objects there is one cluster each of T2 and T3 objects, and two loose clusters of T1 objects.

A.4 Two-Object Annular Sorting

Generation 21

The fittest individual from generation 21 is not capable of forming a structure resembling an annular structure. At the beginning of a run of the individual some of the ants become loaded with an object. At this stage most of the objects in the world are isolated, and the ants pick up both central and outermost objects. The central objects that the ants pick up are in most cases deposited next to other central objects. However, the central objects are also occasionally deposited next to an outermost object. The outermost objects are both deposited next to outermost and central objects. This causes the formation of three different types of groups early in the run of the individual. The first two groups contain either only central objects or only outermost objects. These groups typically contain two to three objects. The third group contains mostly central objects, but also a few outermost objects. These groups are typically larger than the other two groups. As the individual keeps on running there are fewer and fewer objects that remain isolated in the world. When most objects are part of a group, the ants will pick up objects from these groups and deposit them in the same, or in another, group of objects. This leads to the formation of several groups containing both types of objects. As time goes by most of the groups are broken down, leading to the formation of two or three groups of objects. These groups are not annular in their structure, but the central objects tend to be grouped together in the center of the structure with the outermost objects mostly located on the edges of the structure. Objects are constantly picked up and deposited even after the groups of objects have emerged. Some of the objects that are removed from the groups are deposited in isolated locations, and these are mostly of the outermost type.

The movement of the ants is only in the backward direction. When the ants perceive objects in their eight-cell neighborhood they will turn either to the left or to the right, depending on what type of objects are in each of their perceptual fields. There are also some situations in which the ants remain stationary. In these cases both forward and backward behavior are active, causing the ants not to move in any direction.

Generation 45

The movement of the ants of the fittest individual from generation 45 resembles the movement of the ants of the individual from generation 21. The ants only move backwards, and when there are objects in their immediate neighborhood they turn either left or right. There are also some configurations of objects in the neighborhood that cause the ants to remain stationary. There is, however, a marked change in the behavior of the ants when they have picked up an object. If the ants pick up a central type object they still move in a straight line backwards and turn either left or right depending on the occurrence of objects in their neighborhood. However, if the ants pick up an outermost object they move in circles. This behavior is caused by the ants turning to the left at each step. The circular paths in which the ants move are occasionally displaced due to alternate behavior caused by the noise in the sensor input. When ants pick up an outermost object they thus move in circular paths.

At the beginning of a run of the individual the ants quickly pick up an object. The ants pick up both types of objects, but the majority of the objects carried by the ants are of the central type. After some time groups of central objects appear. These groups do occasionally have outermost objects located next to them. The remaining outermost objects appear to be randomly scattered in the world. As time goes by all of the central objects become part of two to four groups of objects. These groups do initially only contain some of the outermost objects, but as time passes all of the outermost objects are moved to the existing groups of objects. Eventually all of the objects have become part of a cluster of objects. The ants continue to pick up and deposit objects, and as a consequence of this clusters do occasionally disappear. A cluster disappears if all of its central objects are removed. If this happens there is a short period of time where the cluster only contains outermost objects, but these are quickly picked up and deposited on the edge of a cluster containing central objects. The final sort performed by the individual consists of several groups containing both types of objects.

Generation 217

In the fittest individual from generation 217 both the movement behavior and the collective sorting behavior of the ants have changed. The ants now display the same movement if they are unloaded, or if they carry either type of object. They always move backwards, and they turn either left or right depending on the types and the configurations of objects that are perceived.

At the beginning of a run of the individual all the ants quickly become loaded with a central object. These objects are carried by the ants until they encounter another central object. The ants then deposit their object next to this object. This causes the formation of several groups of central objects. Objects are occasionally removed from these groups and deposited next to other groups. This leads to the elimination of some groups while other groups increase in size. As most of the central objects become part of a few large clusters, the ants begin to pick up the outermost objects. These objects are either deposited next to other outermost objects or on the edge of a central type cluster. At the same time some of the central type clusters are broken up leading to the existence of only one, or sometimes two, central type clusters. When all the central objects are part of a large cluster, the ants only occasionally pick up an object. The objects that are picked up at this point are usually outermost objects that are either isolated, or in a homogenous group of two or more outermost objects. These objects are deposited next to other outermost objects, on the edge of a central type cluster, or in isolated locations. Because the outermost objects that are located along the edges of a central type cluster are picked up at a lower rate than outermost objects in other locations, the final configuration of the world is a central type cluster surrounded by most of the outermost objects. The remaining outermost objects are either isolated or form small groups of identical objects.

Generation 331

The ants of the fittest individual from generation 331 displays the same movement behavior as the ants of the fittest individual from generation 217. However, the collective sorting behavior is different. When the individual is run the ants quickly become loaded with either type of object. At this point of the run the outermost objects appear to be easier to deposit than the central objects. As a result of this most of the ants will be loaded with central objects, because when an ant picks up a central object it will not deposit the object. The ants do not deposit a central object unless they encounter a group of central objects. After some time a few small groups of central objects are formed. One of these groups then begins to increase in size as the ants deposit central objects along the edges of the group. At this point the ants start to carry more of the outermost objects. These objects are deposited next to either type of object. When the individual has run some more, the central objects are either part of the large cluster, isolated, or part of a group where the other objects are solely of the outermost type. As the run continues the central objects are removed from the small groups and deposited in the large cluster. This causes the formation of groups containing only outermost objects. The final configuration of the world consists of a single cluster of central objects. This cluster is surrounded by some of the outermost objects while the rest of the outermost objects are part of one or more smaller groups of only outermost objects.

A.5 Three-Object Annular Sorting

Generation 31

The behavior of the ants of the fittest individual from generation 31 is the first to be examined. In the beginning of a run to form an annular structure the ants quickly become loaded with objects. It appears that the ants are equally likely to pick up any type of object at this time. After around 600 steps a cluster of the central object type start to form. At this time the intermediate objects are either isolated or located in groups of two, while the outermost objects remain isolated. The central object type cluster does not contain all the central objects, and as time goes by the cluster disappears only to surface at another location. It is evident that central objects continually form clusters that are then broken down, only to form a cluster at another location. When the first central type cluster begins to form the ants no longer carry the other two types of objects. The picking up of intermediate and outermost objects seem to happen only in the beginning of a run. Because of this the dominant behavior of the fittest individual from generation 31 is the continuous generation of central type clusters that are later broken down. The remaining two types of objects are hardly moved at all, and therefore remains randomly scattered in the world.

When it comes to the movement of the ants they always move forwards, but the direction of movement is influenced by the presence of objects in their eight-cell neighborhood. The general movement of the ants is in a straight line, but when there are objects present in their neighborhood they will turn either left or right, for thereafter to continue in a straight line.

Generation 44

The next individual that was examined is the fittest individual from generation 44. This individual had a fitness of 250.09 in the evolutionary run. At the beginning of the sorting task the ants all become quickly loaded with objects of any type. This situation does however only last for a very brief period. After an initial period the ants only occasionally carry objects of the outermost type. As a result of the ants rarely moving objects of the outermost type, these objects remain randomly dispersed in the world for a long period of the run. The ants are thus mainly concerned with moving objects of the central and intermediate types. After only around 400 steps, one or two clusters of the central object type are formed. In contrast to the central type clusters formed by the individual from generation 31, these clusters are stable. In fact, after two or three clusters of the central type are formed they are never broken up, and it is thus impossible to create only one annular structure containing all the objects. After the central type clusters are formed, the ants start gathering the intermediate objects and drop these around the edges of the central clusters. After all the intermediate objects are located along the edges of the clusters, some of the outermost objects are moved to the clusters as well, although a larger portion of them remain randomly scattered. A final 'annular sort' performed by this individual consists of two or three clusters with central type objects surrounded by intermediate and outermost objects, with most of the outermost objects remaining scattered around the world. The movement of the ants is still only in the forward direction with left and right turns caused by the presence of objects in the vicinity of the ants.

Generation 240

After another 200 generations the fittest individual from generation 240 has a fitness of 302.67. In the beginning of the run of the individual the ants are more restricted in what kinds of objects they carry compared to the individuals from generations 31 and 44. The ants carry mostly central objects, and only occasionally do they pick up an intermediate or outermost object. This type of behavior continues until one or two stable clusters of central type objects are formed. After this it is the central objects that are only carried occasionally. The intermediate objects are now for the most part moved to two different locations. Most of the intermediate objects are moved to the edges of the central type clusters, but some also form groups of three to five objects that may contain outermost objects. Some of the outermost objects are moved to the edges of the central type clusters, but most are moved such that they form small groups of objects, either containing only outermost objects, or a combination of outermost and intermediate objects. After the groups of outermost and intermediate objects are formed they remain for a period of around 100 steps. Then the ants engage in a behavior that continues indefinitely. Most of the intermediate objects are now moved to the edges of the central type clusters, but only a few of the outermost objects remain along the edges. At the same time the small groups of objects are continuously broken down and rebuilt. Objects from the band around the central type objects are also picked up and moved. In this phase intermediate and outermost objects are to a larger degree than previously dropped in isolated locations, causing them to appear to be randomly dispersed in the world.

Compared to the previously described individuals the movement of the ants has changed in one aspect. The ants no longer turn to the left and right at the same rates. The ants turn mostly to the left, and only occasionally do they turn to the right.

Generation 1420

The next individual examined is from generation 1420 and has a fitness of 352.90. The behavior of the ants at the beginning of the run is considerably different from the previously examined individuals. The ants are in the beginning no longer concerned almost exclusively with the central objects. Although the ants start by forming a central type cluster, they also move the other two types of objects around while forming this cluster. This did not happen in previous generations. As the clusters of central type objects grow, more and more of the other two types of objects are moved. The intermediate objects are mostly moved to the edges of the central type clusters, but they also form groups of intermediate objects. The outermost objects are not moved to the edges of the clusters, but are brought together to form groups of outermost objects. These groups consist either solely of outermost objects, or they are formed around isolated objects of the intermediate type. The next marked change in the solution occurs when most of the intermediate objects have been moved to the edges of the clusters. When this has happened the outermost objects begin to get moved to the edges of the clusters. However, the outermost objects also continue to form small groups. The final solution to the annular sorting task created by this individual consists of one or two clusters of central objects surrounded by intermediate and outermost objects. Most of the intermediate objects are located closer to the center of the structure than the outermost objects, but not all. This results in the creation of annular bands around the central cluster that are somewhat noisy.

The movement of the ants is still only in the forward direction, but the turning behavior of the ants has again changed back to what it was in the beginning of the evolutionary run, with the ants turning to the left and right at approximately the same rates.

Appendix B

Weights of the Neural Network Controllers

B.1 Weights of the clustering solution

	H1	**H2**	**H3**
I1	3.1107	−14.4967	−15.9730
I2	0.3222	−11.6929	−1.5600
I3	6.2391	8.5052	−14.3601
I4	−0.4779	3.1421	−9.9979
I5	2.0499	−12.2403	2.6699
I6	−2.9943	9.3006	6.9935
I7	7.5171	−0.9134	−8.2900
I8	11.9525	0.5346	−9.4779
A	−28.7961	7.9842	8.3222

Weights between the input nodes and the hidden nodes of the neural network controller for the clustering solution.

	P	**D**	**F**	**B**	**L**	**R**
H1	−1.2403	3.3515	−3.5095	11.1076	0.3159	1.2931
H2	−1.1508	−3.2327	11.1198	−4.6414	−15.3891	4.7596
H3	5.8120	−12.6881	−15.0424	27.4108	16.1974	−3.5708

Weights between the hidden nodes and the output nodes of the neural network controller for the clustering solution.

B.2 Weights of the two-object patch sorting solution

	H1	H2	H3	H4
I1	5.6002	−16.2402	14.0999	23.9295
	7.9841	−5.5929	6.6927	−2.7108
I2	4.3165	−2.2771	1.0747	−13.6771
	17.5028	0.6621	2.1644	12.1704
I3	−11.9105	−17.0927	11.0559	−15.7878
	1.6364	1.7711	−4.0369	8.5820
I4	6.8636	−8.7017	−8.3461	7.8752
	−0.6096	−3.4830	6.1018	18.5047
I5	7.3927	−2.0107	−1.0524	−14.8289
	−3.3592	5.3683	0.4717	0.7588
I6	−17.1602	7.6619	16.0639	7.2301
	−10.4314	11.2137	13.5808	−2.7221
I7	−13.8524	7.9977	−1.7952	−1.5079
	−1.1849	9.0065	6.4820	0.7849
I8	−27.6461	−6.1788	21.3263	−22.3737
	1.0733	6.4512	15.3626	7.5397
A	5.9107	19.4707	0.2596	0.9555
	−18.3946	−13.0619	−4.6963	19.5859

Weights between the input nodes and the hidden nodes of the neural network controller for the two-object patch sorting solution. The first weight of each input group is from a node that fires if a type 1 object is present in the corresponding perceptual field, and the second weight corresponds to a type 2 object.

	P	D	F	B	L	R
H1	17.7389	9.7945	−19.5107	11.8007	8.7227	−15.7849
H2	3.8238	−21.3878	2.8795	10.4059	−5.5041	0.8660
H3	4.3083	−0.0241	1.0680	4.1980	−18.9684	11.7870
H4	−7.1523	−3.0377	0.8704	−13.2677	2.1745	2.2805

Weights between the hidden nodes and the output nodes of the neural network controller for the two-object patch sorting solution.

B.3 Weights of the three-object patch sorting solution

	H1	H2	H3	H4	H5
I1	3.3541 12.8193 11.5273	−25.4446 3.8095 −10.8833	13.6689 1.2592 9.9939	16.0428 −3.6374 −5.4113	13.8740 −7.6841 −20.0440
I2	9.3146 −4.4775 −1.1605	4.1321 −3.3402 11.7657	−1.0011 −0.2462 −3.3369	12.1906 5.7080 3.0961	3.8887 −9.0826 1.2245
I3	7.9293 16.5778 5.0681	−2.9274 4.4187 −11.2353	−19.4614 2.9386 −13.4852	5.6394 −7.3148 12.4980	−10.5460 −7.3458 12.4890
I4	0.0525 1.1557 −2.3234	2.5331 −22.4689 2.2153	0.2181 −0.0628 −7.3777	13.4991 0.9610 −20.9970	2.4232 2.5305 10.3447
I5	−6.3472 16.3485 0.5268	−3.1651 0.1885 −7.8854	4.5932 3.5683 −18.1666	2.7281 −17.7841 −14.9891	−19.4205 1.3746 8.7779
I6	−6.5117 20.6424 −3.1483	6.3174 −12.7488 −12.0396	−1.3137 1.0705 3.7478	−10.2336 −1.8666 −2.8741	5.1309 6.8733 16.0958
I7	2.7148 −3.4126 6.0847	0.2848 0.5025 3.8313	0.2023 12.8690 −14.9403	4.7393 −6.6377 22.9183	10.4021 −4.1878 9.2738
I8	−7.3006 −4.1824 −1.5886	10.3460 −10.0057 −1.5862	1.2295 −11.1513 −15.4475	−1.0570 5.9993 −12.0426	−5.9386 −16.5323 6.3114
A	9.1325 30.6163 −13.3674	−5.9095 4.0826 13.2600	5.2893 −1.9228 −25.5827	4.6507 12.4924 9.3581	−21.2847 8.2391 −13.5895

Weights between the input nodes and the hidden nodes of the neural network controller for the three-object patch sorting solution. The first weight of each input group is from a node that fires if a type 1 object is present in the corresponding perceptual field, the second weight corresponds to a type 2 object, and the third weight corresponds to a type 3 object.

	P	D	F	B	L	R
H1	−5.1515	−21.5614	4.8220	−7.7488	16.5151	−13.5899
H2	24.4677	−3.1167	13.3974	−12.8607	2.0561	−21.6119
H3	−0.8339	17.4578	5.0999	−4.2554	−3.6840	9.4964
H4	3.5128	−6.4436	−8.5323	−0.8453	−11.5349	7.4893
H5	−0.9027	7.9916	0.2067	−5.9687	−18.6608	12.0358

Weights between the hidden nodes and the output nodes of the neural network controller for the three-object patch sorting solution.

B.4 Weights of the two-object annular sorting solution

	H1	H2	H3	H4
I1	8.1890	−9.1868	3.1409	4.8142
	1.0076	−1.7416	4.5289	4.8240
I2	5.1977	1.9047	9.4553	−19.3064
	−4.1217	2.9905	−0.0484	−2.6161
I3	15.3271	−11.0520	23.8277	−13.3209
	13.8338	4.4095	14.0208	4.9465
I4	6.0292	−14.9325	7.8478	−5.4735
	−7.4794	1.8048	−8.2678	−8.3053
I5	−7.6470	−14.4518	−24.2674	−3.5971
	5.7927	1.2805	9.2334	−25.5180
I6	9.3056	−8.7271	−9.8991	−8.2958
	6.4006	−1.2731	−13.6622	5.7557
I7	2.1711	−6.3158	−12.5694	−19.1546
	13.6253	−0.4342	−6.5302	−8.6460
I8	−1.5564	−9.2240	0.7326	15.5749
	−1.4222	−1.1981	−5.3620	−5.4486
A	−3.7348	29.7273	4.7268	5.4405
	13.3194	1.6192	−0.8239	−6.1060

Weights between the input nodes and the hidden nodes of the neural network controller for the two-object annular sorting solution. The first weight of each input group is from a node that fires if a central object is present in the corresponding perceptual field, and the second weight corresponds to an outermost object.

	P	D	F	B	L	R
H1	−6.1060	2.6558	−23.0238	4.5430	−0.6813	11.6831
H2	16.2657	−20.9214	−6.6839	20.6548	−2.7601	−6.1537
H3	6.2403	−5.8251	−6.4594	−10.1856	16.0479	−13.4615
H4	2.8049	−2.3875	2.3464	−7.6824	−13.7999	−4.4325

Weights between the hidden nodes and the output nodes of the neural network controller for the two-object annular sorting solution.

B.5 Weights of the three-object annular sorting solution

	H1	H2	H3	H4	H5
I1	−8.8554 0.0636 4.9090	−6.9773 −10.2601 −7.1958	4.7543 2.6005 −6.6436	−5.0067 −6.5896 21.3629	−5.6242 −4.6053 3.5585
I2	−10.4066 −4.4742 40.0854	−12.4964 −7.8706 −14.6717	−20.6907 −11.3700 −2.6005	9.3474 −5.6217 17.8471	10.5577 10.8974 9.4462
I3	−13.3812 −0.4145 2.8677	−7.7892 1.5788 9.6823	16.4772 8.8592 −7.2485	−1.2652 10.3969 −7.4688	5.3211 6.4543 −4.3220
I4	−12.5645 −3.7842 −0.9512	1.6202 −4.0423 −5.9577	−23.0578 −2.9160 4.5904	−17.1913 10.7070 0.9817	−9.9539 −3.3007 −6.4934
I5	−4.1354 −4.4189 −13.2337	−18.5600 −5.7110 −14.1586	−29.7086 19.1680 9.1074	10.2509 17.1326 −9.0013	8.7577 16.9818 3.0825
I6	−6.5137 0.4599 3.6858	−6.1416 12.4709 7.5152	20.4864 12.2297 −12.0451	−9.7952 12.1516 9.4021	−9.7862 11.1230 3.5342
I7	−14.7397 −6.5361 −3.3573	−0.3089 −1.9316 −28.9402	1.8071 −4.0502 −11.6406	−7.7787 11.3361 −5.6689	16.4815 −11.0411 −3.3256
I8	−17.8851 −1.9204 −0.0601	−5.7448 −14.0966 −7.2883	21.9404 5.3706 6.0055	−0.2007 4.5334 8.2013	5.6772 6.4374 21.5769
A	22.9544 7.8931 −1.4041	9.4024 0.4131 2.8481	12.6595 10.5549 −7.4620	6.2381 −15.9924 −9.6113	−7.9057 11.0764 18.9744

Weights between the input nodes and the hidden nodes of the neural network controller for the three-object annular sorting solution. The first weight of each input group is from a node that fires if a central object is present in the corresponding perceptual field, the second weight corresponds to an intermediate object, and the third weight corresponds to an outermost object.

	P	D	F	B	L	R
H1	11.0764	−5.4093	8.4537	−29.6185	−16.4671	21.6694
H2	2.6739	−20.1966	−6.4421	−4.9010	−25.4109	−15.2076
H3	3.8054	−10.8612	8.5909	14.3441	28.5957	−12.1485
H4	−1.1835	2.4962	1.8131	−5.2162	−5.8019	−4.1271
H5	−5.3439	1.4049	16.0352	−1.1580	−3.0617	−6.5533

Weights between the hidden nodes and the output nodes of the neural network controller for the three-object annular sorting solution.

Bibliography

[1] G. C. Barker and M. J. Grimson. The physics of muesli. *New Scientist*, 126:37–40, 1990.

[2] R. Beckers, Owen Holland, and Jean-Louis Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Proceedings of the Fourth International Conference on Artificial Life*, pages 181–189, 1994.

[3] Eric Bonabeau. Social insect colonies as complex adaptive systems. *Ecosystems*, 1:437–443, 1998.

[4] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[5] Eric Bonabeau and Guy Theraulaz. Swarm smarts. *Scientific American*, March 2000.

[6] Eric Bonabeau, Guy Theraulaz, Jean-Louis Deneubourg, Serge Aron, and Scott Camazine. Self-organization in social insects. *Trends in Ecology & Evolution*, 12(5):188–193, 1997.

[7] Rodney Brooks. A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 2(1):14–23, 1986.

[8] Scott Camazine. Self-organizing pattern formation on the combs of honey bee colonies. *Behavioral Ecology and Sociobiology*, 28:61–76, 1991.

[9] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, New Jersey, 2001.

[10] Scott Camazine, James Sneyd, Michael J. Jenkins, and J. D. Murray. A mathematical model of self-organized pattern formation on the combs of honeybee colonies. *Journal of Theoretical Biology*, 147:553–571, 1990.

[11] L. Chrétien. *Organisation Spatiale du Matériel Provenant de l'excavation du nid chez Messor Barbarus et des Cadavres d'ouvrières chez Lasius niger (Hymenopterae: Formicidae)*. PhD thesis, Université Libre de Bruxelles, 1996.

[12] Jean-Louis Deneubourg, S. Gross, Nigel R. Franks, Ana Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–363, 1991.

[13] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[14] Nigel R. Franks and Jean-Louis Deneubourg. Self-organizing nest construction in ants: Individual worker behavior and the nest's dynamics. *Animal Behaviour*, 54:779–796, 1997.

[15] Nigel R. Franks and Ana Sendova-Franks. Brood sorting by ants: Distributing the workload over the work-surface. *Behavioral Ecology and Sociobiology*, 30:109–123, 1992.

[16] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[17] Pierre-Paul Grassé. La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes sp.* la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.

[18] Julia Handl, Joshua Knowles, and Marco Dorigo. Ant-based clustering and topographic mapping. *Artificial Life*, 11, In Press, 2005.

[19] Vegard Hartmann. Evolving agents swarms for clustering and sorting. In *Proceedings of GECCO 2005*, In Press, 2005.

[20] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, 1975.

[21] Owen Holland and Chris Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 1999.

[22] Nicolas Labroche, Nicolas Monmarché, and Gilles Venturini. On improving clustering in numerical databases with artificial ants. In *Lecture Notes in Artificial Intelligence*, pages 626–635, 1999.

[23] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson Education Limited, Essex, England, 2002.

[24] Erik D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 501–508, 1994.

[25] M. Martin, Bastien Chopard, and P. Albuquerque. A minimal model for the formation of an ant cemetery. In *Proceedings of ACRI 2000: Theoretical and Practical Issues on Cellular Automata*, pages 90–98, 2000.

[26] Kishan Mehrotra, Chilikuri K. Mohan, and Sanjay Ranka. *Elements of Artificial Neural Networks*. MIT Press, Cambridge, Massachusetts, 2000.

[27] Chris Melhuish, Owen Holland, and Steve Hoddell. Collective sorting and segregation in robots with minimal sensing. In *Simulation of Adaptive Behavior: From Animals to Animats 5*, pages 465–470, 1998.

[28] Chris Melhuish, Matt Wilson, and Ana Sendova-Franks. Patch sorting: Multi-object clustering using minimalist robots. In *Lecture Notes in Artificial Intelligence*, volume 2159, pages 543–552, 2001.

[29] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, Massachusetts, 1998.

[30] Vitorino Ramos and Juan J. Merelo. Self-organized stigmergic document maps: Environment as mechanism for context learning. In *Proceedings of AEB02 - Spanish Conference on Evolutionary and BioInspired Algorithms*, pages 284–293, 2002.

[31] Samuel R. Scholes, Matt Wilson, Chris Melhuish, and Ana Sendova-Franks. Comparisons in evolution and engineering: The collective intelligence of sorting. *Adaptive Behavior*, 12(3-4):147–159, 2004.

[32] Ana Sendova-Franks, Samuel R. Scholes, Nigel R. Franks, and Chris Melhuish. Brood sorting by ants: Two phases and differential diffusion. *Animal Behaviour*, 68(5):1095–1106, 2004.

[33] Matt Wilson, Chris Melhuish, Ana Sendova-Franks, and Samuel R. Scholes. Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17:115–136, 2004.

Index