Heterogeneous Neuro-Evolutionary Specialization of Collective Rover Behaviors

– Master's Thesis –

Wouter Lieffijn (s1277154)

September 8, 2008

Internal supervisor: Marco Wiering
Artificial Intelligence Department
Faculty of Mathematics and Natural Sciences
University of Groningen
Nijenborgh 9
9400 AK Groningen

External supervisor: Geoff Nitschke
Computational Intelligence Group
Computer Science Department
Vrije Universiteit
De Boelelaan 1081a
1081 HV Amsterdam

Abstract

Within the context of heterogeneous neuro-evolutionary specialization, we compared evolutionary methods operating on the network level with methods operating on the neuron level, as well as evolutionary methods employing multiple genotype populations with methods employing just a single population. As such, four approaches for neuro-evolution were compared: multi-population network-level evolution, single-population network-level evolution, multi-population neuron-level evolution, and single-population neuron-level evolution. These approaches were compared in compliance with two instances of the Multi-Rover Problem, namely the distributed rover task and the collective rover task. According to our results, for either task, network-level evolution appeared more suitable than neuron-level evolution. Besides that, it appeared that single-population evolution is more suitable for the distributed task, whereas multi-population evolution performs better on the collective task. Multi-population evolution consistently leads to a greater amount of diversity than single-population evolution.

Contents

Part I

Introduction

Biological social systems have since long been a source of inspiration to engineers. The collective behavior research on multi-agent systems and artificial life in particular has attempted to replicate the success of social insect societies at decomposing the group's labor into several complementary roles. By decomposing labor, insect societies are able to accomplish goals unattainable by elaborate but solitary individuals. A particular problem domain amenable to biologically inspired engineering is the domain of multi-agent systems, as behavioral specialization has proved to be advantageous for certain types of multi-agent tasks. In this project, our primary interest lies in the emergence of behavioral specialization for the purpose of improving collective performance on two extensions of the Multi-Rover Task. The desired outcome is the derivation of a set of complementary specialized behaviors that together yield a (nearly) optimal level of task performance.

1 Research Questions and Hypotheses

Within the context of heterogeneous neuro-evolutionary specialization, we aim to compare on the one hand evolutionary methods operating on the network level with methods operating on the neuron level, and on the other hand evolutionary methods employing multiple genotype populations with methods employing just a single population. As such, four neuro-evolutionary methods are compared: multi-population network-level evolution, single-population network-level evolution, multi-population neuron-level evolution, and single-population neuron-level evolution. The methods are compared with respect to the degree of behavioral specialization of the evolved rover collectives as well as their performance . We may thus formulate four research questions:

> Which type of evolution leads to greater performance,
> multi-population evolution or single-population evolution?

Given that the rover tasks require specialization, the multi-population approaches should be advantageous, as multi-population approaches have been shown to encourage specialization [28]. May it appear that specialization is not required, then the single-population approaches are expected to be advantageous, as a single genotype population is less likely to converge prematurely.

> Which type of evolution leads to greater performance,
> network-level evolution or neuron-level evolution?

The background section treats several issues on conventional network-level neuro-evolution. Considering the fact that neuron-level evolution was designed to solve these issues, we may expect that neuron-level evolution will prove advantageous. Earlier comparisons between network-level evolution and neuron-level evolution [18, 16] support this hypothesis.

> Which type of evolution leads to stronger behavioral specialization,
> multi-population evolution or single-population evolution?

Multi-population evolution is expected to bring forth greater degrees of specialization than single-population evolution. Multi-population methods evolve a separate collection of genotypes for each rover; single-population methods evolve a single population shared by all group members. During the evolutionary process, the genotypes evolved by single-population methods may be recombined with genotypes from other rovers. The genotypes evolved by multi-population methods on the contrary cannot be recombined with genotypes from other rovers. The segregation in multi-population methods allows the separate genotype populations to follow their own evolutionary path and become more and more deviant from each other, whereas the genotypes from single-population methods are persistently related by recombination.

> Which type of evolution leads to stronger behavioral specialization,
> network-level evolution or neuron-level evolution?

We have no evidence that either level of evolution will produce stronger behavioral specialization.

2 Thesis Overview

Background The background section provides a raw sketch of the two pillars whereupon our research is based, namely the specialization of agent collectives and the evolution of neural networks, or neuro-evolution. The intersection of these two fields is often referred to as neuro-evolutionary specialization. The first chapter elaborates how specialization emerges from interaction between division of labor and dedication of collective members. The second chapter gives a brief summary of network-level evolution and neuron-level evolution.

Method The method section covers the description of our experiments. In the first chapter, two instances of the Continuous Multi-Rover Problem are elaborately described, referred to as the distributed rover task and the collective rover task. The second chapter covers the evaluation of individual rover performances. Thereafter, the compared neuro-evolutionary methods are described. The fourth chapter then discusses the quantification of specialization. Finally, the last chapter summarizes the parameter settings as used in our experiments.

Results and Discussion In this section, the results of our experiments are treated. Before the actual results are presented, some remarkable qualitative features of the evolved behaviors are described, followed by a discussion on the development of several important statistics. So as to find out which method leads to the strongest behavioral specialization, the idea was to apply the specialization measure proposed in the method section. Due to unexpected developments within the experiments, however, this measure appeared inapplicable. The second best option then was to apply the measures for dedication and diversity independently from the proposed specialization measure. These measures were thus used to compare the four evolutionary methods.

Conclusion In the conclusion section, our findings are summarized. The evolutionary processes showed a development quite different from what we presumed. Thereby, we were unable to use the proposed specialization measure. As we still wanted to form an impression of how strong the several evolutionary methods stimulate specialization, the next best option was to compare dedication, diversity, and performance independently.

Part II

Background

The background section provides a raw sketch of the two pillars whereupon our research is based, namely the specialization of agent collectives and the evolution of neural networks, or neuro-evolution. The intersection of these two fields is often referred to as neuro-evolutionary specialization. The first chapter elaborates how specialization emerges from interaction between division of labor and dedication of collective members. The second chapter gives a brief summary of network-level evolution and neuron-level evolution.

3 Specialization

In our world, numerous examples of complex adaptive systems exist, including complex ecological and social communities, such as social insect colonies or macro-level economies, as well as (biological) neural networks and infrastructure networks. Within many of these complex adaptive systems, a mechanism called *specialization* can be observed. This mechanism allows collectives to increase performance or quickly respond to changes in their environment. According to Seligmann [31], specializations in complex ecological communities have evolved over time as a means of diversifying the community in order to adapt to the environment. For instance, morphologically specialized castes have emerged in certain termite colonies [25], while honeybees may adapt their foraging behavior for pollen, nectar, and water to individual preference and colony demand [8]. Ecological communities like these have been a source of inspiration for design principles used in many artificial collective behavior systems. These design principles are especially prevalent in multi-robot systems [27], swarm intelligence [5], and artificial life [24].

Two major components constitute the mechanism for specialization, namely the *division of labor* amongst members of a collective, and the *dedication of members* to a certain share of labor. The interaction between these two components may adopt both morphological [37] and behavioral [6] forms and potentially gives rise to the emergence of specialized castes. Both forms of specialization apply to collectives that are situated and embodied, while operating in physical or virtual environments.

3.1 Division of Labor

Robinson [29] states that division of labor in insect colonies is characterized by two features: (a) different activities are performed simultaneously by (b) groups of specialized individuals, which is assumed more efficient than if tasks are performed sequentially by non-specialized individuals. Although these two features do characterize division of labor within many collectives, they do not hold for division of labor in general. Division of labor may also occur within homogeneous (and therefore non-specialized) groups performing a sequence of tasks, as members are forced to successively take over control from one another, due to for instance limited battery capacity. Labor division thus may also play a role if activities are performed neither simultaneously nor by specialized individuals.

The essence of labor division is actually that the execution of a collective's general task is broken up into smaller parts, which then are distributed amongst multiple members of the collective. Given a collective and its overall task, labor division may lead to either sequential or parallel execution, and may or may not lead to differentiation, dependent on the nature of the collective and the task environment. Typical distributions of labor are distribution over *time* (members successively take over task execution from one another), distribution over *space* (members perform similar tasks at different locations), and distribution over *roles* (members perform tasks fitting their individual capacities and preferences), as well as mixtures of these.

We often speak of labor division as if it would be a freestanding part of the system, a separate module that distributes labor by explicitly guiding individual members to their tasks. In most collective behavior systems, however, such a distribution actually occurs as a result of local decision-making. Division of labor typically is not coordinated on the collective level, but arises from choices autonomously made by individual members. Each member deduces its next action by applying behavioral rules to the information received, in the hope that the action will be in line with those of other members. The collective-level distribution of labor thus lies in the hands of individual-level controllers and emerges from local interactions between members.

3.1.1 Advantages of Labor Division

Labor distribution may enhance the performance of a collective in several ways. In the first place, a labor distribution mechanism is there to prevent members from becoming idle and uncertain regarding their next goal, while there is still work to be done. Individual members must be guided towards a new goal as soon as they finished working on their current share of labor, so as to ensure that they contribute to the collective performance as much as they can.

Secondly, a labor distribution mechanism must keep the collective members from standing in each other's way. For instance, any two members should never try to perform exactly the same task, as either one of them will be watching how the work is done, or even worse, will interfere with the work of the other. Any form of counter-productive interference between members should be eliminated.

The third purpose of a labor distribution mechanism is to choose the right member for each task. Within almost any collective, there exist inequalities between members. Within heterogeneous collective behavior systems, the morphological and behavioral inequalities are obvious. Yet, even within homogeneous collectives, one can find numerous inequalities with regard to the circumstances of individual members. For example, the members of a homogeneous collective may then be morphologically and behaviorally identical, still they may occupy different positions or have different amounts of energy remaining. Although these smaller differences may often seem insignificant, for an accurate view on labor division, they turn out to be just as important as the more outstanding differences. Inequalities within a collective cause that some of its members are more suitable to cope with certain tasks than others, due to advantageous characteristics and circumstances, while other members may be more suitable for other tasks. A mechanism for labor division is to assign each task at disposal to the member that is expected to be most proficient in fulfilling it.

3.1.2 Criteria for Division of Labor

In order to find the member that is most proficient for a particular task, a mechanism for division of labor compares the collective members by a number of criteria. Each of these criteria evaluates a single aspect of the individual characteristics and circumstances. Some criteria evaluate members with respect to a particular aspect of their *competence*, others with respect to their *availability*, while still others evaluate members with respect to their *willingness* to perform a task. The resulting evaluations together form an impression of a member's proficiency. By comparing the evaluation results, the mechanism for labor division infers for each encountered task which member will be most convenient.

Competence Because of the morphological and behavioral differences that exist within heterogeneous collectives, some members will be more capable of performing certain tasks than others, simply due to the fact that these members have morphological and behavioral traits that are advantageous with respect to these tasks, or lack traits that are disadvantageous. The dependency of labor division upon the differing *competences* of collective members is beyond questioning, since it is a member's competence in performing a certain task that will be decisive for successful accomplishment. Criteria based on competence may for instance compare capabilities for crossing rough terrain or look for members with instruments for nocturnal perception.

Availability However, a mechanism for division of labor that is merely dependent on competences will always choose the member most capable of completing it, even if this member is not available. It might therefore come up with impractical solutions. A division of labor mechanism thus also needs to account for criteria selecting for a member's *availability* for a particular task. Such criteria are for instance a maximum distance between a member and the location where the task should be performed, or a minimally required battery level. Incorporating a dependency on availability of members allows for a more practical division of labor.

In addition to that, the dependency on availability of collective members enables labor distribution amongst members with equivalent capabilities for certain tasks, as it selects the member with the greater availability. A labor division mechanism that incorporates a dependency on availability therefore will be capable of solving conflicts between equally capable members in heterogeneous collectives, as well as that it enables division of labor for homogeneous collectives.

Willingness For a given task, an individual's competence and availability together determines its appropriateness. If the individual's capabilities and circumstances allow it to solve a particular task, then the individual must be appropriate. Besides the criteria of competence and availability, however, one might choose to introduce a third criterion: the individual's *willingness* to perform a task. The willingness of an individual can be described as a behavioral predisposition towards performing particular shares of labor, either acquired by learning or genetically inherited, but without respect to the individual's appropriateness. That is, an individual may simply prefer some tasks to others, whether or not that preference is justified by its competence or availability. Labor distribution on the sole basis of willingness does not look for members appropriate for a task, but instead chooses those just willing to perform it.

In their labor division model for social insect workers, Theraulaz et al. [35] nicely illustrate the role that individual preference may play within distribution of labor. They equipped each of their artificial workers with a set of task response thresholds. Such threshold determines the likelihood of reacting to certain task-associated stimuli. The higher a worker's response threshold for stimuli associated with a certain task, the stronger the stimuli must be for the worker to react. In addition, performing a certain task induces a worker to decrease its associated response thresholds, while not performing the task induces the worker to increase them. The artificial workers thus varied their response thresholds during lifetime so as to adjust their responsiveness to stimuli. A worker with a preference for particular tasks will reinforce that preference by reacting to associated stimuli, but will also decrease the likelihood of performing other tasks as it ignores stimuli associated with these depreciated tasks. This combined reinforcement process evokes the emergence of specialized workers from a group of initially identical individuals.

A worker's set of response thresholds refers to neither its competence nor its availability, but instead represents nothing more than the worker's preference for some of the available tasks. The smaller a worker's threshold for a certain task is, the more the worker is willing to perform this task. Theraulaz et al. [35] thus designed a labor division model that does not take into account the appropriateness of workers, but instead distributes labor exclusively based on varying individual preferences.[1]

3.1.3 Arguments for Willingness-Based Criteria

One might wonder what use criteria based on willingness may have: why would we allow a collective member to decide whether it is willing to perform a task, especially if it is already known that this member will be the most convenient? One argument is that such criteria may solve conflicts between furthermore equal members. Criteria based on willingness as such enable distribution of labor for collective behavior systems lacking any foothold for criteria based on competence or availability, such as Theraulaz's artificial insect society mentioned above. A labor division mechanism operating in such systems can be used to optimize the collective performance, even though it cannot distribute labor with respect to competence or availability.

A second argument is more interesting though. Criteria based on willingness can be used to take account for other criteria, on the assumption that the substitute (willingness-based) criteria are somehow representative for the originals. Such replacement might for instance be useful for omitting competence-based criteria, which may be rather hard to construct. One might choose to prescribe the morphological and behavioral characteristics required to perform particular

---

[1] The artificial worker as modeled by Theraulaz et al. [35] actually is nothing but a set of response thresholds, so to say. It does not have any characteristics that would define some sort of competence, nor does it have a position, energy level, or whatsoever, from which its availability for a task could be deduced.

tasks, but this often will be an unattractive solution, as such criteria mostly will not be very flexible and certainly close the door to unexpected but valuable solutions. Besides that, it will not always be possible to prescribe the required characteristics, as it may not always be clear what these characteristics should be like. A more flexible solution then is to establish a proper relationship between a member's competence and willingness, after which the competence-based criteria can be substituted for one based on willingness. The members then learn to prefer tasks they are good at, but dislike those they cannot cope with. A criterion based on willingness as such selects the more proficient members without comparing competences.

Evolutionary methods provide a nice example of how such a relationship between competence and willingness can be established. For any individual counts that the greater the resemblance between its willingness and competence, the smaller the chance it will deny tasks it is convenient with, or accept tasks it cannot handle. Just as well, it counts that individuals exclusively performing tasks they are good at will be more productive than those performing tasks they cannot handle. Considering that evolutionary methods repeatedly select for productivity, the less productive individuals will eventually be eliminated, and only those of which their task preference correctly represent their capabilities will be left, provided that they do have relevant capabilities of course. During an evolutionary process, the relationship between willingness and competences is established as a matter of course.

3.2 Dedication of Members

During the learning stage, individuals have their performance optimized gradually. Each step is made by slightly adjusting morphology and behavior in such a way that performance increases. The difficult thing here is that adjustments made with a view to improved performance on specific shares of labor, might also have unwelcome side effects concerning performance on other shares. Individuals that must be sufficiently equipped for any task they may encounter are thereby forced to spread their capacities over all shares of labor and will often have to settle with just moderate competences. For instance, individuals operating on their own cannot afford to concentrate on just a few shares of labor, while paying no attention to other shares. They must be sufficiently capable of handling any task they may encounter, simply because no help is available.

If such an individual had been part of a collective, however, it also would have had fellow members that could account for some of its tasks. Yet, as part of a collective, an individual may only drop its responsibility for certain shares of labor, if it can be confident that it will not have to perform these anymore.

To establish such confidence within a collective, its labor must be distributed amongst the members in a consistent way, so that they will be able to learn which shares of labor they may be assigned to and which not. A consistent division of labor allows members to ignore the larger part of the workload and to dedicate themselves to the smaller part left. These dedicated individuals are then free to further elaborate competences that are advantageous for their part of the workload. Even though the development of extraordinary competences still will be at the cost of other competences, there is no harm in it anymore, as these other competences, employed for the benefit of tasks now ignored, have lost their relevance.

Of course, since all encountered tasks still have to be executed, the members of a collective must not all compete for the same part of the workload, as that would leave other tasks unattended. Instead, they must complement each other, so that despite the fact that they individually cover only a small part of the workload, yet they collectively handle all tasks encountered. This implies that collective members have to assume different roles. Some members take care of one part of the workload, while others will take care of another part. As the members then claim different parts of the workload, however, they will also require different characteristics, since different tasks demand different morphological and behavioral traits. Each collective member therefore has to develop its own morphology and behavior, so it can optimize these for its own range of tasks. Members thus have to differentiate from each other.

Note that the criteria regulating the division of labor also determine the possibilities for dedication. Even though any morphological or behavioral trait is allowed to differentiate, only the differentiation of traits invoked by at least one of the prevailing criteria will be relevant for the distribution of labor. Any trait ignored by these criteria is insignificant as far as concerned by the specialization process. The criteria for labor division thus decide what traits will form a basis for dedication and as such will play a role in the specialization process.

3.2.1 Advantages of Member Dedication

The differentiation of collective members nevertheless is not a strict requirement for distribution of labor. In their work on *Adaptive Teams of Agents*, Bryant and Miikkulainen [7] show that even members of homogeneous collectives may adopt heterogeneous roles appropriate to their environment. Even though homogeneous collectives impose a single morphology and behavior to all of their members, they may thus still negotiate a division of labor merely based on availability of members. However, the strict similarity between the members of a homogeneous collective implies that each of them must be capable of performing any share of labor the collective may encounter. Collectives consisting of such generalist members therefore must have an eminent degree of redundancy. A high degree of redundancy provides for great flexibility and robustness [33, 7], but also goes hand in hand with reduced efficiency. Section 3.2 discussed how generalist individuals are forced to settle with just moderate competences for their broad spectrum of tasks. Collectives merely consisting of generalists therefore cannot appeal to members with special competences. As such, their homogeneity may impede their performance.

On the contrary, heterogeneous collectives let each member develop its own morphology and behavior. Such individual development encourages members to differentiate and allows them to dedicate themselves towards their own range of tasks. The main reason for allowing collective members to differentiate is to gain extra performance. The previous section (3.2) already argued that differentiation is mandatory for dedication of members. Dedication then on its turn encourages members to develop outstanding competences. If the labor division mechanism for each part of the workload will be capable of finding a member that is especially convenient, one may expect that the complete workload is performed in a more efficient way than it would be done by an undifferentiated collective. Provided equal amounts of time and resources, this leads to greater performance.

The second reason for allowing differentiation is to reduce the complexity of morphologies and behaviors by simplifying division of labor. It might be tempting to conclude that labor distribution for homogeneous collectives must be less complex than for heterogeneous collectives. Homogeneous collectives after all mostly have greater flexibility and robustness than their heterogeneous equivalents, which should make it easier to find members for the tasks at disposal. In addition, for homogeneous collectives, the criteria for labor division refer only to the availability of members, whereas for heterogeneous collectives, they refer not only to the availability of members, but as well to their competence and willingness. This conclusion will often be misleading however. The apparent simplicity of labor distribution within homogeneous collectives is often only superficial.

Typical mechanisms for division of labor are – whether applied to homogeneous or heterogeneous collectives – often entirely based on local decision-making; the individual members of a collective choose their tasks autonomously and solely based on local information. In order to ensure that labor is performed efficiently and tasks are selected in accordance with other members, they must somehow individually be capable of applying the prevailing criteria for labor division to the information they collected. The crucial point here is that any collective member must do so according to its own range of tasks. The generalist nature of members belonging to homogeneous collectives requires that they must be able to collect all pieces of local information that might be relevant for any possible task, to which they then must apply all criteria for labor division. The dedicated members from heterogeneous collectives just as well apply the relevant criteria to the information collected, but as their spectrum of tasks is much more restricted, the greater part of criteria will be irrelevant and the information to which these criteria would be applied may just be ignored. Division of labor therefore demands less coordination from members of heterogeneous groups than from members of homogeneous groups. This is likely to have consequences for the complexity of morphologies and behaviors.

The advantages of heterogeneous specialization do come at a certain cost though. Whereas homogeneous specialization processes invariably have to optimize only a single morphology and behavior, heterogeneous processes have to optimize as many morphologies and behaviors as there are collective members, or at least as there are specializations. Even though genotypes of homogeneous specialization processes generally are more elaborate than those of heterogeneous processes, as they encode complexer morphologies and behaviors, the search space of heterogeneous processes will usually still be much greater, as they evolve large numbers of genotypes. In order to exhaustively explore the search space, heterogeneous specialization processes require more computation than homogeneous processes.

3.3 Emergent Specialization

The essence of emergent specialization lies in the interaction between division of labor and dedication of members. At the same time the workload is divided amongst collective members in such a way that each share of labor is assigned to the member most proficient in fulfilling it, each of the members individually dedicates itself to the share of labor it received so as to indeed become more proficient. From this interaction between division of labor and dedication of members rises a diversification of collective that may eventually turn into a set of specialized *castes*. In the course of a collective's specialization process, as the members dedicate themselves to their own range of tasks, one may expect that each member's behavioral repertoire becomes more and more restricted to a limited number of actions and increasingly deviant from those of other members. Therefore, within any group subject to specialization, both the degree of dedication and the degree of diversity are expected to grow.

However, note that specialization involves more than just diversification and dedication. Considering diversification, Li et al. [21] were right in arguing that collectives become specialized only if behavioral diversity is evoked for increased performance. The same holds true for dedication. Collectives may then become greatly diverse – for instance, due to noise in the replication mechanism – and their members strongly dedicated, but if performance does not benefit, one can hardly speak well of specialization. The mere purpose of specialization is to enhance a collective's performance, so diversification and dedication should both serve for that.

4 Neuro-Evolution

Evolutionary algorithms are designed to find suitable parameter values for a system, so that it may operate properly given a certain environment and task. Within the field of neuro-evolution, these algorithms are applied to find suitable parameters for neural networks. Well-known parameters are the strengths of connections between neurons, as well as several activation function parameters, such as the activation threshold and decay.

4.1 Advantages of Neuro-Evolution

Evolution of neural networks has some advantages over more traditional network training methods, like back-propagation [30] and cascade correlation [13]. Firstly, it allows for more flexibility in network design. Most traditional network training methods calculate the gradients required for hill-climbing search from errors in the network output. The derivation of gradients requires continuous activation functions and is costly if applied to complex network architectures, like for instance recurrent networks [39]. Since evolutionary methods do not rely on hill-climbing techniques, the development of networks with complexer architectures and discontinuous activation functions comes within reach.

In the second place, neuro-evolutionary methods do not require examples of correct behavior. These methods may therefore be used for unsupervised learning. There are numerous problem domains to which supervision is not available, because either supervision may be impractical or simply no examples of proper behavior exist. Training methods that rely on supervision cannot cope with such domains. Neuro-evolution does not deduce its reinforcement signals from correct examples and is thus capable of building networks in response to problem domains where supervised learning is not possible.

Thirdly, evolutionary methods are able to train under sparse, infrequent reinforcement. Apart from problem domains to which supervised learning techniques cannot be applied at all, many other domains only allow for sparse, infrequent reinforcement. For instance, certain sequential task domains provide reinforcement signals only after a sequence of actions, or even just one signal at the end of an agent's lifetime. In such domains, training networks will be difficult using back-propagation or other traditional techniques, due to the credit assignment problem. Traditional techniques require separate reinforcement signals for individual network outputs, but receive signals only after series of outputs, which makes it impossible to distinguish how much each individual network output contributed. Since neuro-evolution methods require just one signal at the end of an agent's lifetime, instead of separate reinforcement signals for individual network outputs, they can be applied to sequential tasks more easily. Besides the sequential task domains, many other domains exist as well that return only sparse, infrequent reinforcement. For these domains, neuro-evolutionary methods thus will be more suitable than traditional methods.

4.2 Conventional Neuro-Evolution (CNE)

Conventional methods for neuro-evolution encode a set of network parameters as a string of values, which then serves as a network's genotype representation. A large number of such genotype representations is stored in a population. These genotypes are repeatedly evaluated according to a particular task. In order to evaluate the genotypes, from each is constructed a network, which is then used to solve the task. From the performance of the network is derived a quantitative measure that describes how well this network is suitable for the task, often referred to as the network's fitness value. The fitness value is fed back to the corresponding genotype. After each evaluation, the evolutionary algorithm recombines and mutates the genotypes in accordance with their respective fitness values. The more suitable genotypes are preserved, whereas less suitable ones are mutated or even replaced by recombinations of superior genotypes. By repeatedly selecting for greater performance, an evolutionary method eventually develops a population of appropriate genotypes.

4.2.1 Issues on Conventional Neuro-Evolution

There are some issues on conventional neuro-evolution, however, concerning a bias in recombination of genotypes and premature convergence of populations. In conventional approaches, neural weights correspond directly to a location on the genotype. This strict one-to-one mapping inhibits the flexibility of evolutionary algorithms. The forced correspondence between weights and gene locations introduces a bias with regard to recombination of weights. When using n-point crossover, with n much smaller than the size of the genotypes to recombine, genes located close to each other have greater chance to be recombined than genes located far from each other. This bias may be counter-effective, since weights corresponding to genes located far from each other may still produce highly fit neural structures.

The second issue concerns genetic convergence that is stronger than beneficial. Methods for neuro-evolution repeatedly select and breed the best genotypes of a population. The genetic convergence that inherently follows from this selection and breeding is crucial for the elimination of inferior genotypes. However, without being taken care of, the convergence of a population may also obstruct its development, by prematurely converging the population to a very small set of unique genotypes. Common conventional evolutionary methods are incapable of moderating excessive convergence.

A great loss of genetic diversity is undesirable because it makes exploration of the solution space less efficient and effective. Given a diverse population, genetic algorithms are forced to search different niches of the solution space simultaneously, since different genotypes inherently do not occupy the same niche. However, if population diversity reduces, more and more genotypes will be searching the same niche, making the exploration less efficient. Apart from that, recombination of greatly different genotypes permits greater traversals in the solution space and enables populations to overcome local maxima. If diversity diminishes, greater traversals in the solution space become infeasible and exploration may be trapped into a local maximum, making it less effective.

Excessive convergence also confines genetic adaptability. Given a dynamic task and environment, genetic populations must be capable of quickly adapting to changes. Populations with larger numbers of different genotypes incorporate adaptability to a wider variety of tasks and environments. As its diversity reduces, a population has less unique genotypes available and may thereby not be able to respond adequately.

4.2.2 Maintaining Population Diversity

Several techniques for maintaining genetic diversity have been developed. The most common techniques are to weaken the selection strategy, or to increase the mutation rate. However, delaying convergence by weakening the selection strategy roughly boils down to delaying the complete evolutionary process, ensuring retarded development rather than greater population diversity. The second technique, increasing the mutation rate, attempts to maintain diversity by adding extra noise to a genetic population. Adding extra noise does not result in diversity among proven solutions, however, but in diversity among more or less random values. Besides that, increasing the mutation rate limits the resolution of the genetic search, and as such prevents the population from closely approaching optimal solutions.

Somewhat more sophisticated techniques for maintaining genetic diversity are *fitness sharing* [15], *crowding* [10], and *local mating* [9]. In order to preserve genetic diversity, techniques like these only recombine genotypes that do not differ too much. So as to determine if genotypes do not differ too much, any pair of genotypes that might be recombined must be compared. However, since comparing genotypes is computationally expensive, such comparison-based techniques are impractical for large populations of elaborate genotypes [32].

Another technique that must ensure genetic diversity is *implicit fitness sharing* [20, 32]. In this technique, the evolutionary pressures for diversity do not rely on comparison of genotypes, but emerge from the fact that individuals are being forced to cooperate. Cooperation between individuals is forced, as the individuals do not represent complete solutions to a task, but only partial solutions that need each other in order to complete a common task. This way, the population cannot loose much of its diversity, as many different individuals may be necessary for completing the task.

4.3 Symbiotic Adaptive Neuro-Evolution (SANE)

The work of Horn et al. [20] and Smith et al. [32] on implicit fitness sharing was not designed for neuro-evolution, however. Moriarty and Miikkulainen suited the work to neuro-evolution [22, 23], which resulted in the neuro-evolutionary counterpart of implicit fitness sharing, called *Symbiotic Adaptive Neuro-Evolution*, or SANE. Similar to implicit fitness sharing, which operates on a population of genotypes representing partial solutions instead of full solutions, SANE primarily operates on a population of genotypes representing neurons instead of complete neural networks.

The SANE method thus evolves a population of neuron genotypes. Apart from that, the procedure of iteratively evaluating genotypes and performing genetic operations is quite similar to that of conventional neuro-evolution. From several neuron genotypes together, a network is constructed and then evaluated according to a certain task. The neuron genotypes involved in the construction of the network are rewarded with respect to the network's performance. After all genotypes have been evaluated, the evolutionary algorithm recombines and mutates the genotypes in accordance with their respective fitness values. Also see figure 1.

Evolution of neuron populations on itself, however, cannot generate complex neural networks. On top of the evolution of neuron genotypes, a second population of network genotypes is evolved that must keep track of effective combinations of neurons. This extra evolutionary layer roughly compares to conventional neuro-evolution, including selection and breeding based on fitnesses, but with the difference that these network genotypes do not actually incorporate blueprints of neurons by themselves, but as it were point to neurons from the underlying neuron genotype population.

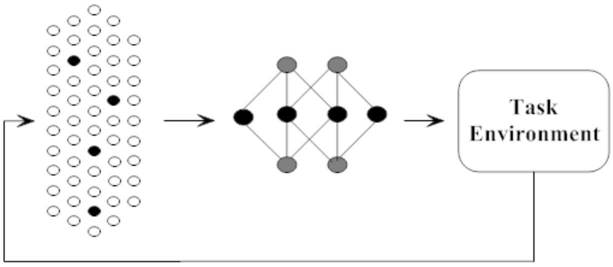

Figure 1: Graphical representation of SANE.

4.3.1 Symbiotic Evolution of Neurons

Obviously, any single neuron cannot perform a complex task by itself; it may only target a small aspect of the complete network response. So, a number of different neurons is forced to cooperate, so as to combine the different aspects of the response. The fact that neurons cannot solve a complex task on their own, but rely on genetically different neurons, ensures that the genotype population cannot loose much of its diversity. In fact, the population will increasingly become divided into various *clusters* of which the genotypes target approximately the same aspect of the network response. Separate clusters emerge from which the genotypes have the same specialization. Neurons from the same cluster, thus having the same specialization, compete with each other, while neurons from different clusters, thus having different specializations, may cooperate with each other. Moriarty [23] uses the term *symbiotic evolution* here, referring to the type of co-evolution where individuals for their survival explicitly rely on the presence of each other.

The advantages of symbiotic evolution are twofold: the maintenance of population diversity increases efficiency and effectiveness, as well as that it allows for greater accuracy in evaluating possible solutions. In contrast to evolutionary methods that evolve complete networks, SANE evolves neurons representing only partials solutions. As these neurons rely on each other, none of their genotypes can take over the complete genotype population, which ensures a certain level of population diversity. As pointed out in section 4.2.1, population diversity is desirable for efficiently and effectively searching for optimal solutions.

SANE also allows for greater accuracy in evaluating solutions. Conventional evolutionary approaches evaluate neurons within the context of invariably the same neural network. In such approaches, highly fit neurons may easily be concealed by poor performance of other neurons in their network or, the other way round, poor neurons may take advantage of favorable neurons. In contrast, SANE evaluates each neuron given multiple, different networks, so as to make evaluation of neurons less dependent on the networks they were part of, which should result in greater accuracy.

4.3.2 Issues on SANE

Regrettably, the SANE method is not free from shortcomings either, since it seems to be designed only for tasks satisfying the Markov assumption. The Markov assumption means that the next state of a system entirely depends on the present state and is thus independent of previous states. Although many tasks are Markovian in nature or can be formulated such that they satisfy the assumption, they may still appear non-Markovian to agents, as these may not have complete knowledge of the environment. Having agents to incorporate some form of memory then may help them to solve their task. Agents may for instance be able to remember locations of objects that went out of sight. If SANE would be able to evolve networks incorporating memory, then the performance on non-Markovian tasks might benefit from that.

A common way to incorporate memory is to evolve networks with recurrent connections between their neurons. Recurrent connections implement a form of memory by maintaining input activation for more than one round. SANE has some difficulties evolving recurrent networks however. A neuron's behavior strongly depends on the behavior of afferent neurons, as its behavior changes as soon as the behaviors changed of neurons it receives input from. In order to function properly, a neuron must therefore be able to have at least some expectations regarding the behavior of afferent neurons. So as to achieve those expectations, over several generations, the behavior of neurons subsequently occupying a certain position in network must be more or less consistent. Small changes are necessary in order to search for optimal solutions, but dramatic changes distort the development.

However, in subsequent generations, SANE allows positions in the hidden layer to be occupied by neurons from different specializations. Whereas genotypes from the same specialization become more or less consistent to each other throughout the evolutionary process, genotypes from different specializations rather should not, as they target different aspects of the network response. As SANE allows neurons from different specializations to subsequently occupy the same position in the hidden layer, it can comprehensibly not achieve the expectations of hidden neurons about their neighbors. SANE thus will only be reliable for evolution of neuro-controllers of which the evolved neurons do not depend upon each other, thereby ruling out evolution of recurrent networks. This is a serious drawback, since reactive single-layer neuro-controllers are only sufficient for less interesting, simpler tasks.

4.4 Enforced Sub-Populations (ESP)

Three years after the introduction of SANE, Gomez and Miikkulainen came up with a second neuro-evolutionary method operating on the neuron level [17]. Gomez's method primarily differs from SANE in that SANE's emerging specializations in the single neuron population are substituted for pre-specified specializations, called *sub-populations*. The forced segregation of specialization clusters solves the expectancy issues on SANE (see section 4.3.2), allowing it to evolve recurrent networks. Gomez exploited this solution to build an evolutionary method called *Enforced Sub-Populations*, abbreviated as ESP.

4.4.1 Enforcing Specializations

The behavior of hidden neurons participating in a recurrent network is strongly dependent on the behavior of other hidden neurons. Throughout the course of evolution, these neurons must be able to expect a certain behavior from other hidden neurons in order to behave properly. Since SANE utilizes loose, implicit segregation of specialization clusters, it cannot satisfy those expectations, disabling it to reliably evolve recurrent networks. The most obvious solution is to replace SANE's loosely, implicitly segregated specialization clusters with strictly and explicitly segregated clusters, or sub-populations. For each position in the hidden layer, such a sub-population is pre-specified. As evolution proceeds, neuron behavior within the several sub-populations becomes more and more consistent, and hidden neurons may increasingly expect a certain behavior from other hidden neurons. This enables ESP to achieve the expectations of neurons about their connected neighbors and so to evolve networks with a more complex structure, such as recurrent networks. Although Gomez extended his method with several features, the basic evolutionary procedure is furthermore similar to SANE, as shown in figure 2.

In [16], Gomez and Miikkulainen argue that segregating neurons into sub-populations speeds up the evolutionary process for two reasons: the specializations do not have to organize themselves out of a single large population, and their progressive specialization is not hindered by recombination across specializations. According to Gomez and Miikkulainen, these cross-specialization recombinations would fulfill relatively orthogonal roles in the evolution of networks. Recombination of genotypes from different specializations may result in offspring from which the genes are completely inconsistent with each other, even if the parent genotypes were highly fit.[2]

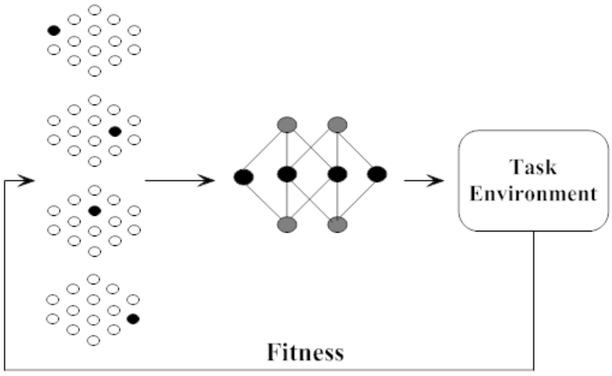

Figure 2: Graphical representation of ESP.

4.4.2 Burst-Mutation

Gomez realized that without recombination across different specializations, ESP is subject to premature convergence of populations, just as well as conventional evolutionary methods are, as treated in section 4.2.1. He therefore equipped ESP with a method called *burst-mutation*, based upon Whitley's *Delta-Coding* method [38]. If performance stagnates for a predetermined number of generations, all genotypes in the least contributing sub-population are substituted for noisy clones of the sub-population's best genotype. To find the poorest sub-population, all sub-populations are compared using *lesioning*, a technique described in section 4.4.4. The newly generated genotypes must then explore the surrounding evolutionary space of the best genotype.

On the one hand, since the best genotype mostly will already have at least some competence in solving the task, dramatic changes will not be necessary for most of the weights. On the other hand, however, a few dramatic changes should not be prohibited, in order to regain genetic diversity. The Cauchy distribution is suitable for generating a genetic noise that fulfills these requirements. During a burst-mutation, the concerned genotypes are mutated according to the Cauchy distribution:

$$f(x) = \frac{\alpha}{\pi\left(\alpha^2 + x^2\right)}$$

---

[2] Note that on the one hand Moriarty states that recombinations across specializations are necessary to maintain population diversity, while on the other hand Gomez and Miikkulainen state that such recombinations will fulfill relatively orthogonal roles. So there must be a trade-off between population diversity and genotype consistency.

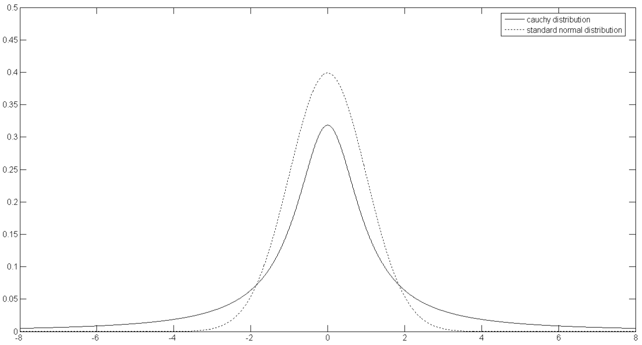

Figure 3: The Cauchy distribution ($\alpha = 1$) and the normal distribution ($\mu = 0$ and $\sigma = 1$). Notice the longer tails of the Cauchy distribution. While most of the returned values will be small, the Cauchy distribution also allows for the more dramatic genetic changes.

4.4.3 Adapting the Network Size

Whether neural networks are trained using gradient-descent techniques or using neuro-evolution techniques, it is important for these networks to have the appropriate number of neurons in order to learn a task sufficiently. Too few or too many neurons may seriously reduce the generalizing capabilities of networks, or at least slow down the learning process. Networks equipped with too few neurons are not powerful enough to be able to generalize the task. On the other hand, oversized networks that learn by means of gradient-descent techniques will more or less store the training data instead of its distinctive features, whereas oversized networks trained by means of neuro-evolution will evolve several ineffective neurons. An evolved network containing too many neurons will thus in general not overfit its training data, yet excess neurons will slow down the evolution process and reduce the accuracy of evaluations.

ESP is able to adapt the size of the evolving networks by adding or removing sub-populations. When performance stagnates for a predetermined number of generations and burst-mutation already has been applied, a neuron may be either added or removed, by respectively adding or removing a sub-population. Using *lesioning*, described in the next section (4.4.4), each of the sub-populations is tested for its contribution to the network response. If every single sub-population proves to be sufficiently contributing, a new sub-population is added, thereby extending the network with an extra neuron position. If one or more sub-populations are found to be insufficiently contributing, the least contributing is removed, and the corresponding neuron position is withdrawn.

4.4.4 Lesioning

If the progression of the evolutionary process stagnates, ESP may apply burst-mutation or adapt the network size. Both features need to know which sub-population is least contributing, as burst-mutation is applied only to that sub-population and adaption of the network size possibly comprehends its withdrawal. A technique called *lesioning* provides this information.

During a lesion study, a network is evaluated with one of its neurons disabled, or lesioned. The resulting loss of performance indicates the contribution of this neuron to the network output. If the performance of the network degraded considerably, the contribution of the neuron apparently was significant. On the other hand, if the performance did not degrade that much, or even increased, the contribution of the neuron must have been insignificant, or even counter-effective. By successively dedicating a lesion study to each neuron, the lesioning technique compares the significance of neurons and their corresponding sub-populations. The lesion with the least degraded performance indicates the least contributing neuron position.

4.4.5 Enforced Sub-Populations for Multi-Agent Systems

In their paper on cooperative co-evolution of multi-agent systems, Yong and Miikkulainen [40] presented a multi-agent extension of the ESP method. The Multi-Agent ESP method comes down to simultaneously running a separate ESP process for each of the collective members, while still evaluating the members together, according to a certain multi-agent task. Each ESP process constructs a network by choosing a neuron from each of its sub-populations. The networks from the several ESP-processes then are brought together to form an agent collective. The collective is evaluated with respect to a certain multi-agent task and the involved genotypes receive a certain fitness value, derived from the collective's performance. Figure 4 shows a graphical representation of the Multi-Agent ESP method.

Bryant and Miikkulainen [7] employed the ESP method to evolve agent collectives as well. However, in contrast to the extension of Yong and Miikkulainen, their version of the ESP method involves just a single ESP process. An agent collective is composed by constructing a single network that then is duplicated for each of the collective members. Whereas the method by Yong and Miikkulainen was designed to evolve heterogeneous collectives, Bryant and Miikkulainen designed theirs to evolve homogeneous collectives.

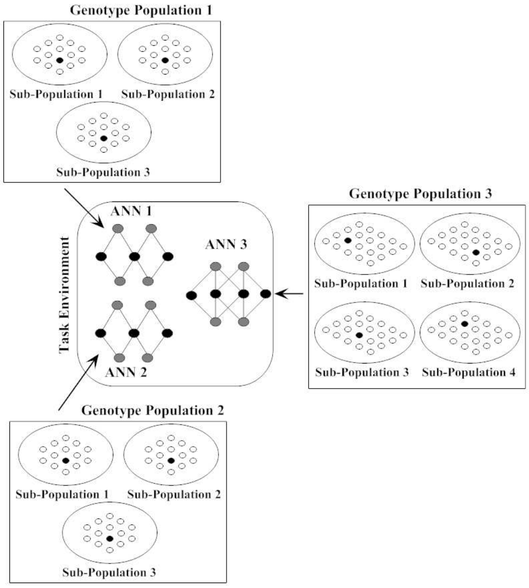

Figure 4: A graphical representation of the Multi-Agent ESP method.

Part III

Method

The method section covers the description of our experiments. In the first chapter, two instances of the Continuous Multi-Rover Problem are elaborately described, referred to as the distributed rover task and the collective rover task. The second chapter covers the evaluation of individual rover performances. Thereafter, the compared neuro-evolutionary methods are described. The fourth chapter then discusses the quantification of specialization. Finally, the last chapter summarizes the parameter settings as used in our experiments.

5 Continuous Multi-Rover Problem

A well-known reinforcement learning problem is the *Grid World Problem* [34]. An agent navigates about a two-dimensional $n \times m$ grid space, of which each square represents a certain value. After each move, the agent is rewarded for the value of the grid square it has moved into. The observable state space of the agent consists only of its current grid coordinate. Given a limited number of moves, the agent then must use that information to find the path that will maximize its sum of rewards.

Agogino devotes a chapter of his dissertation [3] on the description of the *Discrete Rover Problem*, which essentially is a multi-agent version of the traditional Grid World Problem. Agogino's version is slightly different in that it involves a collective of multiple autonomous agents, or rovers, to look for certain tokens that are distributed throughout the discrete environment. When a rover moves into a grid square holding such token, it receives a reward for this token's value, after which the token is removed. The rover collective must then learn to maximize the accumulate reward.[3]

In the next chapter of his dissertation, Agogino describes yet another version of the Rover Problem. In the *Continuous Rover Problem*, a group of autonomous rovers searches for points of interest, just as in the Discrete Rover Problem, but then within a continuous environment. Whereas the number of possible positions is limited for a discrete $n \times m$ grid space, the continuous environment allows for a (theoretically) infinite number of possible positions, even if the size of the environment is restricted.[4]

---

[3] See [36] as well.

[4] See [1, 2] as well.

5.1 Environment

In our case, the task environment is a continuous but confined two-dimensional space containing a given number of rocks. These rocks are the points of interest that rovers will be looking for. Each of them represents a certain value, deviating from one rock to another, but always greater than zero. Three types of rock are present: red rocks, green rocks, and blue rocks. Central point in the design of the task environment (as well as in the design of the rovers themselves) is that no type of rock may a priori be more appealing for rovers than another. Each rock type therefore represents an equal amount of rock value, distributed amongst an equal number of rocks, with an equal deviation in rock value. Figure 5 shows an example of how rocks may be distributed.

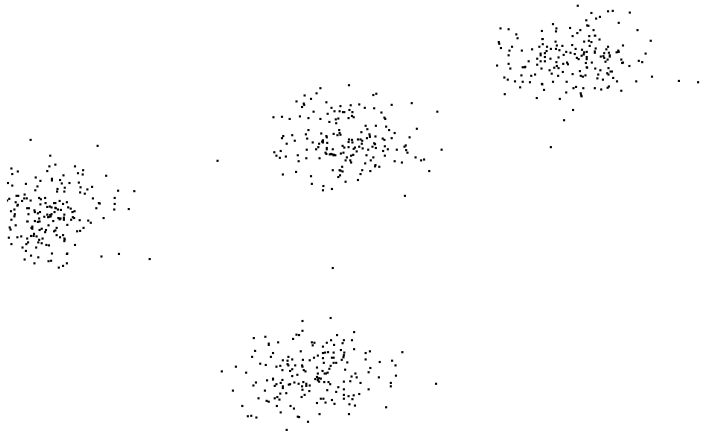

Figure 5: A possible distribution of rocks.

As one can see in the figure, rocks are not uniformly distributed over the environment, but clustered into several piles. These rock piles are randomly located in the environment, but in such a way that the piles do not overlap. Even though all piles contain an equal number of rocks, the amount of rock value is not necessarily equal per pile. The clustering is done with respect to the several rock types, in the sense that one-third of each pile consists of red rocks, one-third of green rocks, and one-third of blue rocks. Besides that, within the piles, rocks of a similar type tend to be located somewhat closer to each other, thereby sometimes forming smaller loose sub-clusters. This bias is very slight though. The sub-clusters of a rock pile should not be situated too far apart, as then it would be likely that one sub-cluster will be easier to find than the others, thereby making one type of rock more appealing.

5.2 Task

Given the described environment, the task for a rover group is to discover as much rock value as possible, provided a limited number of simulation time steps and a limited amount of energy. For our project, we implemented two versions of the rover task, namely a task requiring *distributed behavior* from rover groups and a task requiring *collective behavior.* In the distributed rover task, a rock is "discovered" as soon as a single rover successfully detected it. This in contrast to the collective rover task, which prescribes that a rock may only be discovered if at least two rovers simultaneously detected it – see figure 6. Although the difference between the distributed behavior task and the collective behavior task is very slight, it still makes that rovers adopt fundamentally different approaches.

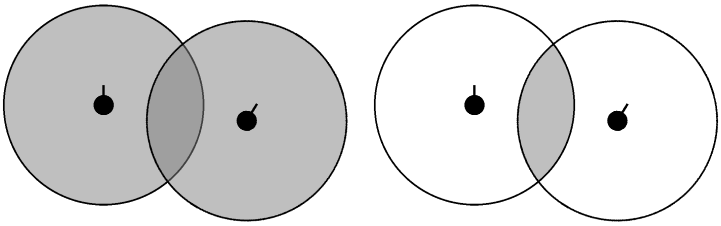

Figure 6: Two pairs of rovers trying to discover rocks. In the distributed task (left pair) rovers may discover rocks individually. The joint area in which rocks are discovered (colored gray) is the union of the individual perceptive fields. In the collective task (right pair), rovers may only discover rocks if at least two of them simultaneously detected these. The joint area in which rocks are discovered thus is the intersection of the individual perceptive fields.

Under the regulations of the distributed task, only one rover is required to find all rocks, provided sufficient time and energy. Only as restrictions posed on time and energy become stronger, more rovers will be needful for completion of the task. However, these extra rovers only compensate for lack of time and energy. Although rovers performing the distributed task may certainly combine efforts in order to find as much rock value as possible, there is no need to cooperate. In contrast, the collective task prescribes that at least two rovers are required to find a rock. A single rover will never be able to find any rocks at all, no matter how much time and energy is provided. Notwithstanding the fact that stronger restrictions on time and energy may ask for even more rovers, at least two rovers will unconditionally be an essential ingredient for discovery of any amount of rock value. Rovers performing the collective task therefore will have to bring into line their behaviors, since in order to discover a particular rock, two rovers must both be located close to it, as well as that they must activate their sensors synchronously. Cooperation thus plays an essential role in the collective task.

5.3 Rovers

The members of a rover group may explore their environment by moving around, perceiving each other, and perceiving rocks. There are some constraints though. In the first place, the task must be performed within a limited number of time steps, while rovers may perform only one action per time step. For each time step, a rover thus has to choose between motion, rover perception, and rock perception. In the second place, rovers receive only limited energy resources, while each action consumes a certain amount of energy. Once its energy resources are depleted, a rover will not be able to continue exploring its environment anymore. Rovers therefore will have to make good use of their time and resources.

5.3.1 Perception of the Environment

In order to complete their task successfully, rovers of course must be aware of any rocks and rovers present in their environment. For that, each rover is equipped with two collections of sensors: one collection for detection of rocks, another for detection of fellow rovers. The alignments of these two sensor collections are identical. Either collection consists of four sensors aligned orthogonally to the rover's heading, whereas each sensor covers a different quadrant of the rover's environment (figure 7). For either sensor collection, a detection range r and a detection probability p is defined. The detection range is the distance up to which sensors may perceive any rocks or rovers, the detection probability the chance that an attempt to perceive the environment will "succeed". Rovers attempting to perceive rocks or fellow rovers may not do so by activating individual sensors, but have to activate the respective sensor collection in its entirety, thereby obtaining a panoramic view of their surroundings. The success of an attempt is not determined per sensor, but for the collection at once. If an attempt succeeds, all sensors record the presence of rocks or rovers within their respective scope; does the attempt fail, then no rocks or rovers at all are detected.

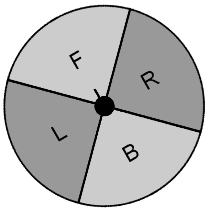

Figure 7: The four fields of view that together give the rover a panoramic view.

Perception occurs at the expense of energy. For both rock detection and rover detection, the detection energy cost Δe_{det} is calculated as:

$$\Delta e_{det} = \rho \pi r^2 p$$

in which ρ is a constant that may differ for rock detection and rover detection. The term πr^2 represents the surface of a rover's panoramic field of view, given detection range r. As such, the energy consumption for detection is proportional to the field of view multiplied by the probability of success p. The detection energy cost is charged regardless of an attempt's success or failure.

Sensor Activation for Rock Detection From a rover's collection of rock sensors S, an individual sensor s receives an amount of activation for each rock within scope, determined by the rock's value and the distance between the rover and rock. Provided that the sensor successfully detected a set of rocks Q_{det}, the amount of activation A_s obtained from these rocks is defined as:

$$A_s = \sum_{q \in Q_{det}} \frac{v_q}{1 + \delta \left\| \mathbf{\Delta x}_q \right\|^2}$$

in which v_q is the value of a detected rock q, $\mathbf{\Delta x}_q$ the disposition between the rock and the sensor, and δ a global constant specifying the slope of the activation function. Note that, according to this function, the more valuable a rock, the more salient it will be to rovers. In addition, the sensor activation decreases as the distance between rock and rover grows. A successful rock detection results in an activation vector $\mathbf{A_S}$ containing $|S|$ sensor activations. As our rovers carry four rock sensors, the activation vector for rock detection contains four activation elements. If an attempt for rock detection fails, the activation vector will be filled with zeros. A rover that received such array filled with zeros cannot tell whether that was because no rocks were found or because its attempt failed.

Whereas in the distributed task rovers may discover rocks independently of other rovers, in the collective task, rocks will only be marked as discovered if detected by at least two rovers concurrently. However, this additional requirement does not affect the perception of rovers itself. Rovers that independently detected a rock still receive an amount of activation for that rock, despite the fact that this rock will not be counted as discovered.

Sensor Activation for Rover Detection The sensors for rover detection receive their activation for perceived rovers in a similar way. The single difference is that rovers do not have some kind of value, so the activation function is merely dependent on the distance between rovers. If U is a rover's collection of rover sensors, and u an individual sensor that successfully detected a set of rovers R_{det}, then the amount of activation A_u for this sensor equals:

$$A_u = \sum_{r \in R_{det}} \frac{1}{1 + \delta \left\| \mathbf{\Delta x}_r \right\|^2}$$

in which $\mathbf{\Delta x}_r$ is the disposition between a detected rover r and the sensor. Parameter δ is the same global constant as used in the rock detection activation function and still specifies the slope of the activation function. Similar to rock detection, successful rover detection results in an activation vector $\mathbf{A_U}$ containing $|U|$ sensor activations. In the case that an attempt for rover detection fails, then the activation vector is filled with zeros.

Detection Modes Corresponding to the three types of rock, rovers can assume three detection modes: a red mode, a green mode, and a blue mode. The relation between these modes and detection of rocks and rovers is rather straightforward. Rovers that assumed a certain mode – red rovers, green rovers, blue rovers – are allowed to detect rocks and rovers of the same color. Red rovers may as such detect red rocks and other red rovers; green rovers may detect green rocks and other green rovers; blue rovers may detect blue rocks and other blue rovers. Rovers can switch between modes at any time.

However, there are two penalties for switching between detection modes. In the first place, after a rover has switched between modes, it will be unable to detect for τ iterations. In addition, switching from one mode to another happens at the expense of the switching energy cost Δe_{mod} according to:

$$\Delta e_{mod} = \mu$$

in which μ is a global constant. A rover remains in its current mode until it tries to detect a different color of rock or rover. Rovers may thus change from detecting rocks to detecting rovers and vice versa, choose to accelerate, or to perform no actions at all, without actually switching detection mode.

5.3.2 Motion

In order to explore their environment, rovers must of course be able to move around. However, rovers cannot control their velocity directly, but will have to adjust it through acceleration. For any acceleration a rover intends, it must select both a magnitude and direction. The magnitude for the intended acceleration is unrestricted, but the direction can only be chosen from the four orthogonals to the rover's current heading – forward, back, left, or rightward.

A rover's motion is defined by the following set of equations:

$$\begin{aligned}\mathbf{\Delta x} &= \mathbf{v} \\ \mathbf{\Delta v} &= \mathbf{a} - \frac{k}{2m} \|\mathbf{v}\| \mathbf{v}\end{aligned}$$

$\mathbf{x}$, $\mathbf{v}$, and $\mathbf{a}$ being vectors representing respectively the rover's position, velocity, and acceleration. In the velocity function, the term $-\frac{k}{2m}\|\mathbf{v}\| \mathbf{v}$ represents friction, with k and m respectively being the friction constant and the rover's mass. In our simulation, both are equal for all rovers. A rover derives its acceleration from the output of its neural network. Section 5.4 elaborately describes the topology of such a network, for now it is sufficient to mention that the network output consists of eleven activation values, of which four values concern the rover's motion. If one of these four activation values exceeds all other values of the network output, the rover will accelerate in the corresponding direction. The magnitude of the acceleration is deduced from the activation value:

$$\|\mathbf{a}\| = \alpha A_a$$

where α is a global constant and A_a the activation value in question. Rovers are charged a certain amount of energy for each of their accelerations, respective to the acceleration's magnitude. The acceleration energy cost Δe_{acc} equals:

$$\Delta e_{acc} = \frac{1}{2} m \|\mathbf{a}\|^2$$

5.3.3 Orientation

The simulation environment is just a small part of the practically unlimited space that the rovers are situated in. No restrictions are posed on the movement of rovers, so even though rovers initially are positioned within the boundaries of the simulation environment, they are not restricted to stay within these. As all rocks are located within the boundaries of the environment, however, rovers are preferred to spend their time within the boundaries as well. The rovers thus will need some kind of orientation, so that they might have a sense of their disposition towards the environment and the rocks, and may choose to return when they are too far off.

Rovers therefore are equipped with two orientation neurons. One of these translates the horizontal component of the disposition towards the center of the task environment, the other the vertical component, simply by keeping their activation level equal to the respective component of the disposition. Rovers then must learn to interpret these activation values correctly and utilize them in order to keep close to the rocks they must discover.

5.4 Control Architecture

Each rover's behavior is determined by its private control architecture. Rovers are equipped with an Elman neural network [11, 12], a triple-layered network having each of its hidden neurons recurrently connected to each other, and fully connected to the neurons of the input and output layers. Figure 8 shows the topology of the network employed by the rovers. In the figure, the neurons from the input and output layers are grouped by functionality. On the left side, one can find the input neurons: those reacting to the rock sensor in front (F), at the back (B), to the left (L), and right (R); the neurons reacting to the rover sensors; and the two neurons translating the horizontal (H) and vertical (V) component of the rover's position. On the right side, the output neurons are shown: those corresponding to the activation of the rock sensors for the three different colors red (R), green (G), and blue (B); the neurons corresponding to the activation of the rovers sensors; the neurons specifying the rover's motion; and the neuron acting as a threshold neuron. In between the input and output neurons, the hidden layer is shown with eight neurons, although its size is to be pre-specified by the user.

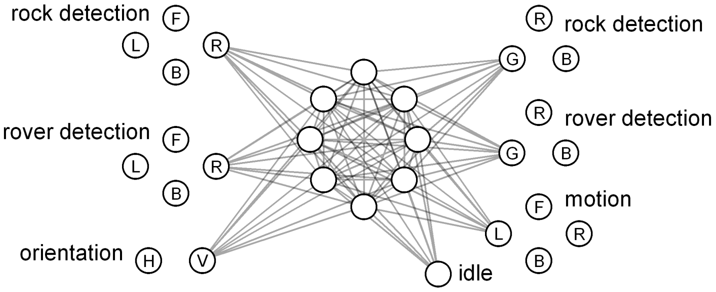

Figure 8: The topology of the Elman network [11, 12] rovers are equipped with. For the sake of clarity, from each group of input or output neurons, one neuron is shown to be connected to the hidden layer, though in fact, all input and output neurons are connected.

This network layout does not provide room for discerning rocks or fellow rovers by their color, as the input layer does not contain separate neurons for the different colors. The reason for this is that our rovers are expected to dedicate themselves to just a single color and the ability to discriminate colors thus would be somewhat superfluous. A certain awareness of colors will doubtlessly be helpful in accomplishment of the rover task. The description of the environment (section 5.1) mentioned that rocks of the same color tend to be located somewhat closer to each other. As a rover just encountered rocks of a particular color, it might want to look out for more, since it will be likely that more rocks of this color are around. In order to decide whether it should keep looking for this color, however, the rover then of course must be aware of the color it just encountered. Just as well, it often will be advantageous for a rover to be aware of the colors of recently detected fellow rovers. The collective rover task prescribes that rocks may only be discovered if two rovers simultaneously detect these. As such, either two rovers must be very lucky looking at the same time for the same color at the same location, or they must somehow coordinate their behaviors.

Generalist rovers require a control architecture that is capable of recognizing colors. Since they may choose to look for any color, one can imagine that they would become confused if their control architecture is not capable of distinguishing colors. Specialist rovers in contrast may still be aware of the color they encountered without truly being able to discern between colors, because they never look out for any but the color they are dedicated to. A rover that will only try to detect red rocks or other red rovers may clearly be certain about the color of the rocks and rovers it observed, even though it might be ignorant of the existence of different colors. As our rovers are to become such specialists, their control architecture has no need to provide for discrimination of color.

Having said that, one might wonder why rovers then still receive three separate pairs of detection output neurons, one pair for each color. It is true that during the specialization process, control architectures are likely to develop one predominant pair of detection output neurons, while the two other pairs become more and more redundant. However, as specialization emerges from interaction between division of labor and dedication of members, one cannot predict what role individual members are to assume. The architecture must therefore avoid any predetermination towards a particular role. This can only be done by supplying neurons for all possible actions.

6 Evaluating Individual Rover Performances

After a collective has tried to complete the task, its members must somehow be evaluated with respect to the discovered amount of rock value. Our starting point for the design of a suitable evaluation function will be that the members of a collective must collectively strive to optimize a global evaluation function rating the collective's performance. One solution would be to couple each member's fitness directly to the collective performance. The collective members will then certainly strive to optimize the collective's performance, since that will in turn straightaway enhance their own fitness. An important drawback of this solution is that it completely ignores contributions of individual members. The more contributing members may easily be concealed by poor performance of other members, while poorly performing members may be given credit for the contribution of fellow members. A better solution is to equip the members with an evaluation function that takes into account their individual contribution.

Agogino and Tumer [1, 2] describe two requirements to which such evaluation functions should comply. The first requirement is that they should be in line with the global evaluation function, so that the maximization of individual fitness does not hinder the collective performance. The second requirement states that they still should be sensitive to individual performances, so that it may provide for the right selective pressure. In the evaluation of a certain member, the impact of other members should be minimized. Based on these two requirements, Agogino and Tumer constructed an evaluation function they refer to as the *difference evaluation function*. Its abstract form is defined as:

$$D_\eta = G(z) - G(z_{-\eta} + c_\eta)$$

The function obtains the fitness of member η by taking the difference between the global evaluation function $G(z)$ and the evaluation function with respect to all variables not affected by member η, denoted as $G(z_{-\eta} + c_\eta)$. In the latter term, $z_{-\eta}$ stands for all components of z on which member η had no effect and c_η is a constant replacing those components of z that were actually affected by η. The most intuitive application of the difference evaluation function is to measure the collective's performance first with all members and thereafter with all members but η. The difference in performance then must have been the contribution of member η.

6.1 Distance-Based Difference Evaluation Function

Furthermore, Agogino and Tumer [1, 2] describe how their difference evaluation function may be applied to the Continuous Rover Problem. In their version of the Continuous Rover Problem, a rover's sensors have unlimited range, so rovers may instantly see all rocks. The evaluation function compensates for that by reducing the reward for observed points of interest as the distance increases. In addition, points of interest are not removed after being observed, otherwise the unlimited sensor range would have caused the points of interest to be removed instantly. Provided that a point of interest i represents a value V_i, then an observation yields an observation value inversely proportional to the distance between the rover and the point of interest:

$$O^\delta_{\eta,i,t} = \frac{V_i}{\delta(L_i, L_{\eta,t})}$$

where $\delta\left(L_j, L_{\eta,t}\right)$ is a measure that increases with the distance between position L_j of rock j and position $L_{\eta,t}$ of rover η at time step t. So, the closer a rover approaches a point of interest, the greater the reward. The collective's task then is to maximize the total amount of value observed. The global evaluation function G and the private evaluation function P_η are respectively defined as:

$$G^\delta = \sum_t \sum_i \frac{V_i}{\min_\eta \delta\left(L_i, L_{\eta,t}\right)}$$

$$P_\eta^\delta = \sum_t \sum_i \frac{V_i}{\delta\left(L_i, L_{\eta,t}\right)}$$

For each time step, the global evaluation function G accumulates the observation values of rovers that were closest to some point of interest. A rover's private evaluation function P_η accumulates for each time step its observation value with respect to all points of interest. By inserting these two evaluation functions into the abstract difference evaluation function, Agogino and Tumer created a distance-based difference evaluation function specifically designed for the Continuous Rover Problem:

$$\begin{aligned} D_\eta^\delta &= \sum_t \left[\sum_i \frac{V_i}{\min_{\eta'} \delta\left(L_i, L_{\eta',t}\right)} - \sum_i \frac{V_i}{\min_{\eta' \neq \eta} \delta\left(L_i, L_{\eta,t}\right)} \right] \\ &= \sum_t \sum_i I_{i,\eta,t}\left(z\right) \frac{V_i}{\delta\left(L_i, L_{\eta,t}\right)} \end{aligned}$$

where $I_{i,\eta,t}\left(z\right)$ is an indicator function, returning one if and only if rover η was closest to point of interest i at time t. Within the square brackets of the first equation, the left term denotes the global evaluation function for time step t, whereas the right term represents the total observation value collected at time step t if rover η were not in the system. The second equation states that η only is rewarded for a point of interest at time steps no other rover was closer.

6.2 Time-Based Difference Evaluation Function

Even though our version of the Continuous Rover Problem is somewhat different from the one Agogino and Tumer [1, 2] designed their difference evaluation function for, we might still exploit some of the function's benefits. In the rover task described by Agogino and Tumer, rovers may perceive any point of interest at any time step, due to an unlimited sensor range. To compensate for that, the observation value had to be inversely related to the distance between a rover and a detected point of interest, to rule out that rovers always collect the maximum amount of observation. With concern to the evaluation functions, a certain notion of time would have had only little meaning, since rovers may always detect each individual point of interest.

In our rover task, however, rovers may detect points of interest only up to a certain range. The observation value thereby does not have that much need anymore for being inversely related to the distance. On the other hand, the notion of time becomes quite important, since rovers may not anymore detect any point of interest at any time step. As such, in our simulation, we are not so much interested in how closely rovers approach the points of interest, or rocks, rather that they should discover as many rocks as possible within a limited time frame, given restricted perception. The evaluation function therefore will not be dependent on the distance between a rover and a detected rock, but on the time passed before a rock was found. Consequently, the distance measure is substituted for a time measure $\tau\left(T_{\eta,i}\right)$ that increases with the time passed. The time-based observation value is defined as follows:

$$O^{\tau}_{\eta,i} = \frac{V_i}{\tau\left(T_{\eta,i}\right)}$$

in which $T_{\eta,i}$ is the time step rover η discovered rock i. So, the earlier a rover detected a point of interest, the greater the reward. The collective's task still is to maximize the total amount of observation value. The global evaluation function G and private evaluation function P_η are thus respectively defined as:

$$G^{\tau} = \sum_i \frac{V_i}{\min_\eta \tau\left(T_{\eta,i}\right)}$$

$$P^{\tau}_{\eta} = \sum_i \frac{V_i}{\tau\left(T_{\eta,i}\right)}$$

The global evaluation function G accumulates the observation values of rovers that were first to observe a point of interest. A rover's private evaluation function P_η accumulates its observation value with respect to all points of interest. By coupling these two evaluation functions to the abstract difference evaluation function, we obtain the time-based difference evaluation function:

$$\begin{aligned} D^{\tau}_{\eta} &= \sum_i \frac{V_i}{\min_{\eta'} \tau\left(T_{\eta',i}\right)} - \sum_i \frac{V_i}{\min_{\eta' \neq \eta} \tau\left(T_{\eta,i}\right)} \\ &= \sum_i I_{i,\eta}\left(z\right) \frac{V_i}{\tau\left(T_{\eta,i}\right)} \end{aligned}$$

where $I_{i,\eta}\left(z\right)$ is an indicator function, returning one if and only if rover η was first to detect rock i. Note that the difference evaluation function rewards faster rovers in two ways. In the first place, the indicator function selects for each rock the rover that was first to detect it. In the second place, the sooner a rover found a rock, the greater the accompanying reward. In a certain sense, the evaluation function thus delivers the same message twice. One might therefore choose to eliminate the time measure:

$$D_\eta^\tau \quad = \quad \sum_i I_{i,\eta}(z)\, V_i$$

The simplified function rewards rovers for the values of the rocks they were first to detect. Provided that rocks are removed after being detected, the rewarding mechanism applies the (simplified) time-based difference evaluation function simply by rewarding rovers for the rocks they discovered. The evaluation function can then also be readily adapted to the collective rover task, by instead rewarding the two rovers first to detect a rock jointly.

7 Methods for Neuro-Evolution

Rover collectives are to become familiar with their task by means of neuro-evolution. Neuro-evolutionary methods employ genetic algorithms so as to find suitable weight values for neural networks. Since the behavior of our rovers is controlled by such neural networks, we can use neuro-evolution to optimize their neuro-controllers. Before the actual process starts, a large number of genotypes is created and then stored in a genotype population. These genotypes will then be repeatedly evaluated and adapted in compliance to the concerned rover task, so that they become more and more appropriate.

Evolutionary processes last for a certain number of generations. With each generation, all genotypes are evaluated according to the rover task several times, before they are put through the process of recombination and mutation. Since our rovers operate as part of a collective, genotypes are always evaluated in association with others. From the genotypes not yet evaluated during the current evaluation cycle, several genotypes are randomly brought together so as to form the blueprint of a rover group. The rover group created from these genotypes then performs its task within each of the specified task environments for a fixed number of iterations, while for each group member is recorded the total amount of rock value it has discovered. The discovered amount of rock value is fed back as a fitness value to the genotypes involved in the construction of the rover. After one rover group has completed its trials, another one is put to the test, until all genes have participated in one of the collectives.

At the end of an evaluation cycle, all genotypes have received a certain amount of fitness value. Such an evaluation cycle is repeated several times, so as to evaluate the genotypes in association with several different genotypes. After the evaluation cycles are completed, the genotypes are recombined and mutated according to their fitness values. Copies are made of the highly fit genotypes, which then are recombined and replace the genotypes with the lowest fitness values. Finally, the genotypes that earned an intermediate fitness value – those neither copied nor replaced – are mutated. After the process of recombination and mutation, the next generation starts.

The background section already hinted that there exist several levels of neuro-evolution. Although more levels may exist, only two were mentioned, namely the network level and the neuron level. Section 7.1 briefly summarizes the distinction between these levels and provides a short description of how we built respective implementations.

In addition to the different levels of neuro-evolution, we also have multiple options regarding the organization of the genotypes. As rover collectives accommodate multiple rovers, the evolution of collectives involves multiple agents. Concerning the organization of genotypes, one option is to provide each rover for its own collection of genotypes, which thus involves multiple genotype populations. Another option is to have the rovers share a common collection of genotypes. Section 7.2 further elaborates the distinction between multi-population evolution and single-population evolution.

7.1 Network-Level versus Neuron-Level

In the first place, this project aims to compare methods for neuro-evolution operating on the network level with methods operating on the neuron level. Conventional neuro-evolution methods typically operate on the network level, in the sense that they evolve genotypes acting as blueprints of complete neural networks. Somewhat more recent methods for neuro-evolution, like SANE [23] and ESP [19], operate on the neuron level, as they evolve genotypes representing single neurons. In sections 4.3 respectively 4.4, it is described how Moriarty [23] came up with the idea to evolve separate neurons instead of complete networks and how that idea was further elaborated by Gomez [19]. Both Moriarty and Gomez argue that neuron-level evolution has certain advantages over network-level evolution, although Moriarty provides for different arguments than Gomez. These arguments are presented in the background section as well.

In order to compare neuro-evolution on the network level with that on the neuron level, we first rebuild Gomez's ESP-method, without the mechanisms for burst-mutation and lesioning. The reason for disposing these stagnation counter-mechanisms is that these techniques are not necessarily unique to evolution on the neuron level, but may as well be incorporated into network-level evolution. In order to make an accurate comparison between both levels of evolution, one must therefore either implement these techniques for both levels of evolution, or leave them out at all. As the counter-measures are likely to make the contest needlessly complex, we chose to leave them out. From our simplified version of the ESP-method, we then constructed its network-level counterpart.

7.2 Single-Population versus Multi-Population

So as to add another dimension to our experiments, we also compared evolving a single shared genotype population[5] for the complete rover collective, as opposed to evolving a separate genotype population for each rover individually. Accordingly, four methods were constructed: a multi-population network-level method, a single-population network-level method, a multi-population neuron-level method, and a single-population neuron-level method.

On the network level, single-population network-level methods work with just a single population of network genotypes shared by all rovers. In contrast, multi-population network-level methods have for each rover a separate population of network genotypes. Nevertheless, for a fair comparison, the total number of genotypes must be equal for both methods, so as to ensure an equally sized search space. Assume n rovers are evolved, where k network genotypes are supplied per rover. A single-population method then makes use of just one population, containing $n \times k$ network genotypes. A multi-population method utilizes n genotype populations, containing k network genotypes each.

Concerning neuron-level evolution, single-population neuron-level methods provide for just one collection of neuron genotype sub-populations, which is shared by all rovers. Neurons that are to occupy the same positions in the several rover networks must be extracted from a common sub-population. In contrast, multi-population neuron-level methods provide each rover with a separate collection of sub-populations. For the network-level methods counts as well that they should involve just as many genotypes. Assume n rovers are evolved, each employing a neuro-controller with m neurons, with k neuron genotypes per neuron position. A single-population method would then count m sub-populations, where each sub-population contains $n \times k$ neuron genotypes. A multi-population network-level would count $n \times m$ sub-populations, with each k neuron genotypes.

The terms "single-population" and "multi-population" may be somewhat confusing concerning the neuron-level methods. However, note that the single-

---

[5] Not to be confused with the sub-populations from the neuron-level methods.

population methods have in common that rovers share their genotype resources, whether these consist of network genotypes or neuron genotypes, whereas in the multi-population methods, rovers have their private genotype resources. Also see figures 9 and 10.

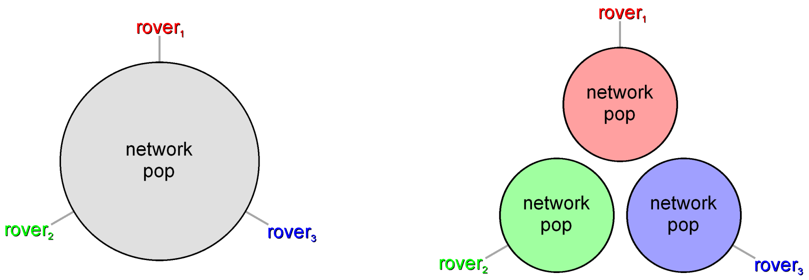

Figure 9: Graphical representations of the single-population network-level method (left-hand figure) and the multi-population network-level method (right-hand figure). In the left-hand figure, three rovers share one network genotype population, while in the right-hand figure, each rover has a private population.

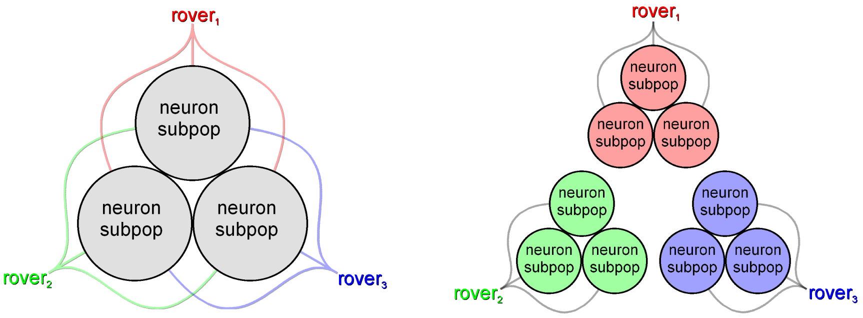

Figure 10: Graphical representations of the single-population neuron-level method (left-hand figure) and the multi-population neuron-level method (right-hand figure). In the left-hand figure, three rovers share a single set of three neuron genotype sub-populations; in the right-hand figure, each of the rovers received its own set.

Regarding the ESP methods for evolution of multi-agent collectives mentioned in section 4.4.5, our multi-population method does indeed closely resemble Yong and Miikkulainen's Multi-Agent ESP method [40], but on the other hand our single-population method is fundamentally different from Bryant and Miikkulainen's version [7]. The latter methods may both then maintain just a single set of sub-populations for the complete collective, but whereas Bryant and Miikkulainen's method extracts only one network and duplicates that for each of the collective members, our method extracts a different network for each of the members. Bryant and Miikkulainen's method as such is designed to compose homogeneous collectives, whereas our single-population method is designed to compose heterogeneous collectives.

7.3 Initial Heterogeneity of Collectives

For the analysis of a specialization process, it would be most intuitive to have a collective of initially identical agents that slowly drift apart into several castes as the agents learn to assume different roles and dedicate themselves to their own share of labor. The specialization process, as it were, turns a non-specialized group of agents into a specialized group. Although such a point of view may be helpful for some mechanisms for specialization, within the context of evolutionary specialization, however, it will undoubtedly lead to a conflict of interests. As the agents of the collective should be identical at the start of the evolutionary specialization process, so must be the genotypes from which these agents are built. However, this will always be in conflict with the interest of evolutionary methods, as these require a certain amount of genetic diversity in order to work properly. Without any diversity, a genotype population will hardly be able to evolve.

Consequently, on the one hand an intuitive analysis requires the genotype population to be initialized homogeneously, while on the other hand a great deal of heterogeneity is imperative for a proper development of agents. As poor population diversity will intolerably frustrate the specialization process and distort the analysis, we chose to provide for a sufficient amount of diversity for a proper development and accept that the analysis of the specialization process may not always lead to intuitive results.

8 Quantifying Specialization

The essence of emergent specialization is that the workload is divided amongst collective members, in such a way that each share of labor is assigned to the member most proficient in fulfilling it, while at the same time members individually dedicate themselves to the share of labor they received, so as to indeed become more proficient. Provided that members have a finite set of different actions to choose from, one may keep a histogram of performed actions for each individually, and use these to measure the degree of dedication as well as the degree of behavioral diversity.

Concerning the measuring of dedication, it counts that the more dedicated a member is to a limited range of tasks, the more the distribution of its action histogram is likely to be concentrated on relatively few actions. The collective measure for dedication thus can be defined in terms of the average degree of convergence in the individual action histograms. Likewise, concerning the measuring of diversity, the more the behaviors of members differ in their selection of actions, then for each different action, the stronger the distribution of its cross-member histogram will be concentrated on just a few members of the collective. The measure for behavioral diversity therefore can be defined as the average degree of convergence in all cross-member histograms.

In the coming sections, the measures for dedication and diversity are further elaborated. First, a method for quantification of a histogram's degree of convergence is introduced. Then is described how this method can be applied to action histograms so as to measure dedication and diversity. After that, these measures are extended for the use of action transition tables, allowing for a somewhat more detailed analysis. Finally, the quantification of dedication and diversity is coupled to the performance of collectives, which results in a measure for specialization. Note that for both the dedication measures and the diversity measures a collective consisting of n rovers is assumed, and that each of these rovers may choose its actions from a common set of m alternatives.

8.1 Measuring the Degree of Convergence in a Histogram

The first step in finding a suitable measure for the degrees of diversity and dedication based on action histograms is to define a function determining the degree of convergence within a histogram. Given a histogram H enclosing n categories, we define the degree of convergence $\operatorname{conv}(H)$ of the histogram as:

$$\operatorname{conv}(H) = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(H_i - H_j\right)^2}}{\sum_{k=1}^{n} H_k}$$

The function accumulates the squared difference between each pair of histogram elements and normalizes the summation so as to guarantee a value between zero and one. If a histogram's degree of convergence is equal to zero, then it is uniformly distributed. On the other extreme, if its degree of convergence equals one, then the distribution is completely oriented towards a single element.

8.2 Measuring Dedication and Diversity from Action Histograms

In order to measure dedication within rover collectives, we denote the k^{th} rover's individual action histogram $\mathbf{A^{rov_k}}$ as the vector

$$\mathbf{A^{rov_k}} \quad = \quad \left[\begin{array}{ccc} rov_k act_1 & \cdots & rov_k act_m \end{array} \right]$$

in which each element $rov_k act_i$ refers to the number of times the k^{th} rover performed the i^{th} action. The vector corresponds to the k^{th} row in table 1. The dedication measure for this rover equals the degree of convergence within its action histogram:

$$ded_k \quad = \quad \mathrm{conv}\left(\mathbf{A^{rov_k}}\right)$$

The degree of dedication $\overline{ded}$ for the complete rover collective then is obtained by taking the average of the degrees of dedication for the individual rovers:

$$\overline{ded} \quad = \quad \frac{1}{n} \sum_{k=1}^{n} ded_k$$

In a similar way, one may also build a measure for diversity within rover collectives. We define the cross-rover action histogram $\mathbf{R^{act_i}}$ as the vector

$$\mathbf{R^{act_i}} \quad = \quad \left[\begin{array}{ccc} rov_1 act_i & \cdots & rov_n act_i \end{array} \right]$$

This vector indicates how often each rover in the collective performed action act_i and corresponds to the i^{th} column in table 1. The diversity measure concerning action act_i is defined as the degree of convergence within the linearly normalized cross-rover action histogram:

$$div_i \quad = \quad \mathrm{conv}\left(\mathbf{\tilde{R}^{act_i}}\right)$$

The elements of the normalized histogram represent the proportions of time spent on respective actions. Given that the members of a collective performed Q actions altogether, the diversity measure $\overline{div}$ for the rover collective is defined as the average of the diversity measures for the several actions, weighted by the number of times q_i each action act_i was performed:

$$\overline{div} \quad = \quad \frac{1}{Q} \sum_{i=1}^{m} q_i div_i$$

$$\begin{array}{ccccccccc}
 & act_1 & act_2 & act_3 & \cdots & act_m & & \\
rov_1 & . & . & . & . & . & \rightarrow & ded_1 \\
rov_2 & . & . & . & . & . & \rightarrow & ded_2 \\
rov_3 & . & . & . & . & . & \rightarrow & ded_3 \\
\vdots & . & . & . & . & . & \rightarrow & \vdots \\
rov_n & . & . & . & . & . & \rightarrow & ded_n \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\
 & div_1 & div_2 & div_3 & \cdots & div_m & &
\end{array}$$

Table 1: The relation between individual action histograms and dedication, as well as cross-rover action histograms and diversity. Each row represents the action histogram $\mathbf{A^{rov_k}}$ for rover k, used for deriving the rover's degree of dedication ded_k. Each column can be interpreted as the cross-rover action histogram $\mathbf{R^{act_i}}$, which yields the degree of diversity div_i regarding action act_i.

8.3 Measuring Dedication and Diversity from Transition Tables

In addition to the action histogram, one may also keep a table of transitions between actions. Such an action transition table thus keeps track of how often one action was succeeded by another. The individual action transition table $\mathbf{U^{rov_k}}$ for the k^{th} rover may be denoted as:

$$\mathbf{U^{rov_k}} = \begin{bmatrix} rov_k trn_{1,1} & \cdots & rov_k trn_{1,m} \\ \vdots & \ddots & \vdots \\ rov_k trn_{m,1} & \cdots & rov_k trn_{m,m} \end{bmatrix}$$

In the transition table, each element $rov_k trn_{i,j}$ refers to the number of occurrences the k^{th} rover has performed action act_j consecutive to action act_i. Putting the rows of the table after another results in the individual action transition histogram $\mathbf{T^{rov_k}}$:

$$\mathbf{T^{rov_k}} = \begin{bmatrix} rov_k trn_{1,1} & \cdots & rov_k trn_{1,m} & \cdots & rov_k trn_{m,1} & \cdots & rov_k trn_{m,m} \end{bmatrix}$$

The histogram corresponds to the k^{th} row in table 2. Provided that the k^{th} rover's degree of dedication obtained from its transition histogram is defined as:

$$ded_k = \operatorname{conv}\left(\mathbf{T^{rov_k}}\right)$$

then the degree of dedication $\overline{ded}$ for the complete rover collective is obtained by averaging the n individual dedication measures:

$$\overline{ded} = \frac{1}{n}\sum_{k=1}^{n} ded_k$$

We may use the action transition histograms to obtain a more detailed measure for diversity as well. The cross-rover transition histogram $\mathbf{R}^{\mathbf{trn_{i,j}}}$ for the transition $trn_{i,j}$ from action act_i to action act_j is defined as:

$$\mathbf{R}^{\mathbf{trn_{i,j}}} \quad = \quad \begin{bmatrix} rov_1 trn_{i,j} & \cdots & rov_n trn_{i,j} \end{bmatrix}$$

The vector indicates how often each rover in the collective performed action act_j right after action act_i, and corresponds to the columns in table 2. The diversity measure concerning action transition $trn_{i,j}$ is defined as the degree of convergence within the linear normalization of $\mathbf{R}^{\mathbf{trn_{i,j}}}$:

$$div_{i,j} \quad = \quad \operatorname{conv}\left(\tilde{\mathbf{R}}^{\mathbf{trn_{i,j}}}\right)$$

Finally, our measure for diversity is obtained by taking the weighted average of the degrees of diversity for each possible action transition. Given that the members of performed Q actions altogether, and that each action transition $trn_{i,j}$ occurred $q_{i,j}$ times, then the degree of diversity $\overline{div}$ is defined as:

$$\overline{div} \quad = \quad \frac{1}{Q} \sum_{i=1}^{m} \sum_{j=1}^{m} q_{i,j} div_{i,j}$$

$$\begin{array}{cccccccccc}
 & trn_{1,1} & \cdots & trn_{1,m} & \cdots & trn_{m,1} & \cdots & trn_{m,m} & & \\
rov_1 & . & . & . & . & . & . & . & \longrightarrow & ded_1 \\
rov_2 & . & . & . & . & . & . & . & \longrightarrow & ded_2 \\
rov_3 & . & . & . & . & . & . & . & \longrightarrow & ded_3 \\
\vdots & . & . & . & . & . & . & . & \longrightarrow & \vdots \\
rov_n & . & . & . & . & . & . & . & \longrightarrow & ded_n \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\
 & div_{1,1} & \cdots & div_{1,m} & \cdots & div_{m,1} & \cdots & div_{m,m} & &
\end{array}$$

Table 2: The relation between individual transition histograms and dedication, as well as between cross-rover transition histograms and diversity. Each row is to be interpreted as the k^{th} rover's individual transition histogram $\mathbf{T}^{\mathbf{rov_k}}$, used for deriving its degree of dedication ded_k. Each column stands for a cross-rover transition histogram $\mathbf{R}^{\mathbf{trn_{i,j}}}$, used for deriving the degree of diversity $div_{i,j}$ with respect to the transition $trn_{i,j}$ from action act_i to action act_j.

8.4 Relating Dedication and Diversity to Performance

In the past, several specialization metrics have been designed. O'Donnell and Jeanne [26] define the degree to which individual forager wasps have been specialized as the entropy of the proportions of activity. Low entropy indicates that a forager has focused on fewer activities. Any metric based upon proportions of activity, however, cannot distinguish between for instance an individual spending half of its time on a certain task A and then switches to task B for the remainder of the time ($AAAABBBB$) and an individual repeatedly alternating between two tasks ($ABABABAB$), since for both individuals the proportions are equal. However, according to Gautrais, Theraulaz, Deneubourg, and Anderson [14], the first individual intuitively still seems more of a specialist than the latter. As such, they designed a specialization metric based upon the frequency of switching between tasks. The lower the frequency of changing between tasks, the more an individual is considered to be specialized.

However, note that in terms of our project, both of these specialization measures actually measure the degree of dedication and thus put individual dedication on a par with specialization. Notwithstanding the ingenuity behind these measures, Li et al. [21] yet have two points of criticism. In the first place, they argue that these definitions are too much focused on individuals, while specialization may also be considered as a quality of the complete collective. Viewed in that light, they propose Balch's measure for collective diversity [4]. In the second place, they argue that specialization measures should somehow be linked to the productiveness of the collective. According to Li et al., the degrees of dedication and behavioral diversity themselves do not say much about the degree of specialization: a collective may then be very diverse and the individual members strongly dedicated, but if the performance of the collective does not benefit, one can hardly speak well of specialization. The mere purpose of specialization is to increase the efficiency of a collective, so dedication and diversity should simply lead to greater performance. A well-designed specialization measure makes clear whether the performance of a particular task actually is enhanced by dedication and diversification. Without relating them to performance, the measures mentioned above thus cannot be used to quantify specialization.

Li et al. [21] designed a specialization measure based on the correlation between a collective's behavioral diversity and its performance. Assuming that the system starts from a homogeneous setting with no diversity or specialization, and the collective's diversity D and performance P change with time as correlated random variables, then the correlation coefficient between D and P naturally acts as the proportion of specialization in diversity. If the developments of diversity and performance are respectively denoted as the vectors $\mathbf{D}$ and $\mathbf{P}$, then the specialization measure $\mathbf{S}$ is defined as:

$$\mathbf{S_D} \quad = \quad R\left(\mathbf{D},\mathbf{P}\right) \times \mathbf{D}$$

where $R\left(\mathbf{D},\mathbf{P}\right)$ is the correlation coefficient between the collective's behavioral diversity and its performance. The specialization measure returned by this function actually is a vector representing the development of the collective's degree of specialization. When the collective's diversity enhances its performance, **D** and **R** will be positively correlated, and $\mathbf{S_D}$ will be positive. If diversity holds back the collective's performance, then **D** and **R** will be negatively correlated, and $\mathbf{S_D}$ will be negative. The specialization measure thus tells whether diversification may be helpful to complete a task. In a similar way, we may also construct a second specialization measure, based on the correlation between the development of dedication **E** and the development of the performance:

$$\mathbf{S_E} \quad = \quad R(\mathbf{E}, \mathbf{P}) \times \mathbf{E}$$

As we have seen in section 7.3, however, for heterogeneous neuro-evolutionary specialization processes, it will often be unrealistic to assume that the evolving collective is initialized homogeneously, since that would intolerably impede the collective's development. The measuring of specialization in heterogeneous neuro-evolutionary systems therefore might require extra care.

9 Experimental Setup

Table 3 shows the parameter settings used in the experiments. Note that the number of epochs per evaluation cycle also determines number of genotypes per rover as well as the number of rover collectives per evaluation. These parameter settings were derived from several preliminary experiments. In these test experiments, minor changes led to similar results. Before the actual experiments were started, the implementation of each evolutionary method was tested in compliance with the Iterated Prisoners Dilemma.

	distributed task	collective task
number of generations	501	1001
number of evaluation cycles per generation	4	
number of epochs per evaluation	32	
number of trials per epoch	4	
number of iterations per trial	250	1000
number of rovers ($\|R\|$)	24	
number of rock piles ($\|P\|$)	4	
number of rocks ($\|Q\|$)	720	
total rock value (v_Q)	36000.0	
rock value deviation	25.0	
rover group starting position deviation	0.25	
rover starting position deviation from group position	0.035	
rock pile position deviation	0.45	
rock position deviation from pile position	0.065	
number of rock sensors ($\|S\|$)	4	
rock sensor power/scope ratio (ρ_S)	0.05	
rock sensor range (r_S)	0.08	
rock sensor resolution (p_S)	1.0	
number of rover sensors ($\|U\|$)	4	
rover sensor power/scope ratio (ρ_U)	0.05	
rover sensor range (r_U)	0.08	
rover sensor resolution (p_U)	1.0	
energy cost for switching colors (μ)	0.0	
delay for switching colors (τ)	1	
number of simultaneous detects to discover rocks	1	2
distance factor for detection of rocks and rovers (δ)	1000.0	
conversion factor for activation to acceleration (α)	0.00025	
friction constant (k)	-100000.0	
rover mass (m)	2000.0	
energy capacity (e)	0.25	1.0
number of inputs	10	
number of hidden neurons	8	
number of outputs	11	
fraction of genotypes as example for recombination	0.25	
mutation probability for initialization	1.0	
mutation probability for common mutations	0.2	
mutation rate	0.55	

Table 3: An overview of the parameter settings used.

Part IV

Results and Discussion

In this section, the results of our experiments are treated. Before the actual results are presented, some remarkable qualitative features of the evolved behaviors are described, followed by a discussion on the development of several important statistics. So as to find out which method leads to the strongest behavioral specialization, the idea was to apply the specialization measure proposed in the method section. Due to unexpected developments within the experiments, however, this measure appeared inapplicable. The second best option then was to apply the measures for dedication and diversity independently from the proposed specialization measure. These measures were thus used to compare the four evolutionary methods.

10 Qualitative Analysis of Evolved Rover Behaviors

In the course of the specialization processes, rovers evolved certain behaviors. Some features of these behaviors are particularly worth mentioning. The fundamental difference in the nature of the distributed task and the collective task unsurprisingly led to different strategies for discovery of rocks. One may expect that rovers performing the collective task would learn to interact with each other, more than rovers performing the distributed task would. In the distributed task, rovers may individually detect rocks, without help of others. Rovers performing the distributed task thus are not likely to become interactive. In contrast, the collective task requires rovers to coordinate their behaviors, since they may discover rocks only if they activate their detection sensors concurrently with at least one other rover. Although the coordination of behaviors itself will be difficult to quantify, the number of rover detections may yet provide for an indication of how strong the interaction between rovers is. The more rover detections were performed, the stronger the interaction between rovers.

10.1 Distributed Rover Task

In the distributed rover task, the members of a rover group may individually look for rocks. In contrast to the collective task, rovers do not have to coordinate their behaviors, since rovers may discover rocks without the help of other rovers. As one might expect, in compliance to the distributed task thus very primitive behaviors evolved. Rovers are all initialized in a random direction. Provided enough rovers, the simplest solution then is have each rover move straight ahead, while looking out for any rocks. Part of the rovers also learned that as they crossed the borders of the simulation environment, they had to turn around and roam the environment in another direction.

While some rovers indeed dedicated themselves to a single color, others seem to have made do with the penalty for switching color modes. In our experiments, sensors were disabled for just a single iteration, so apparently these rovers learned to cope with the brief loss of perception.

10.2 Collective Rover Task

With regard to the collective rover task, the rover behaviors evolved in a rather different way than expected. Whereas the distributed task does not provoke rovers to become aware of each other, we actually did expect some form of interaction with respect to the collective rover task. However, it appeared that even under the regulations of the collective task, rovers just scarcely learned to interact. We assumed that the rovers would learn to stick together pairwisely, so as to be able to detect rocks uninterruptedly, but instead they came up with an even more straightforward and elegant solution. When a rover came across a pile of rocks it would simply stay within that pile, waiting for another rover to come by. As then after a while a second rover with the same color turned up, the rocks could be detected. At last, when the rovers could not discover any rocks there anymore, they would go their own way. Instead of explicitly interacting with each other, rovers made use of changes in the environment so as to coordinate their actions; a mechanism often referred to as *stigmergy*. As such, the rovers were able to cooperate without ever taking notice of each other. Such a strategy is not only easier to learn, but might also be more effective. Whereas interactive rovers would need to spend part of their actions on mutual awareness, the evolved rovers were free to devote all time and energy resources to detection of rocks.

Not all rovers took advantage of the information provided about their disposition towards to the center of the environment. While some rovers consequently returned to the task area as soon as they crossed the boundaries, other rovers simply ignored the signs and did not come back.

11 Developments of Dedication, Diversity, Performance

During the evolutionary processes, for each generation several statistics were collected. From these statistics, the development of the dedication, diversity, and performance were analyzed. The degree of dedication and the degree of diversity were both expected to grow during the evolutionary process. However, it turned out that the development of these statistics would be rather different. Sections 11.1 and 11.2 show how the dedication, diversity, and performance evolved in the course of the processes, in accordance with the distributed task and the collective task, respectively. Section 11.3 then describes how this affects the quantification of specialization.

11.1 Distributed Rover Task

Figures 11, 12, and 13 visualize the developments of respectively dedication, diversity, and performance during the evolutionary specialization processes in the distributed rover task. The graphs contain four series each, corresponding to the four neuro-evolutionary methods being evaluated. Each series is constructed by taking the average of ten runs. Some points in the development of the series are particularly worth mentioning. These are described below.

Dedication The several degrees of dedication rise sharply for just a few generations, but only until the high degrees of dedication quite suddenly collapse, especially for the single-population methods. In the end, the degrees of dedication of the multi-population methods turn out to be slightly greater than for the single-population methods, whereas the neuron-level methods preserved a somewhat greater degree of dedication than their respective network-level counterparts did.

Diversity For each evolutionary method, the development of diversity starts with a quick drop. After that, the degrees of diversity recover for a while, but then again start to decline, although just very slightly for the multi-population methods. The multi-population methods end up with evidently higher degrees of behavioral diversity.

Performance Just as we may expect, all curves presenting the performance begin at a value barely greater than zero. After a brief period of growth, the development of the performance soon encounters a short stagnation. The single-population neuron-level method may then be slower to overcome the stagnation, but compensates for that by suddenly gaining performance at a remarkably high rate. At the end of the evolutionary process, the single-population methods show significantly better performance than the multi-population methods.

11.2 Collective Rover Task

Figures 14, 15, and 16 show the developments of respectively dedication, diversity, and performance during the evolutionary specialization processes for the collective rover task. The graphs each contain four series, one for each of the evaluated neuro-evolutionary methods. Each series is constructed by taking the average of ten runs.

Dedication Generally speaking, the degrees of dedication develop in a way quite similar to those of the distributed task, although the developments seem to be temporized. For just a few generations the degrees rise sharply. The high degrees of dedication then remain for a while, but sooner or later start to decay.

Diversity The diversity series is quite similar to that in the distributed task, in the sense that they show a rapid decline as well, followed by an increase again. After this revival, the degrees of diversity decay during the remainder of the process, although the multi-population methods loose very little. The multi-population methods eventually enclose a significantly larger amount of behavioral diversity.

Performance For each method, the development of its performance starts with a short period of growing performance, followed by a shorter or longer period of stagnation. The multi-population methods then are the first to overcome the stagnation. After its period of stagnation, the single-population network-level method brings forth a promising development, yet seems to drop out after about 300 generations. Although the single-population neuron-level method lingers for more than 250 generations, it still does catch up with the other methods by suddenly showing an impressive growth. The multi-population network-level method performs significantly better than any of the other methods, while its single-population counterpart performs significantly worse. Both neuron-level methods show intermediate performance.

11.3 The Course of Specialization

We presumed that the degrees of dedication and diversity would grow during an evolutionary specialization process, as rovers increasingly dedicate themselves

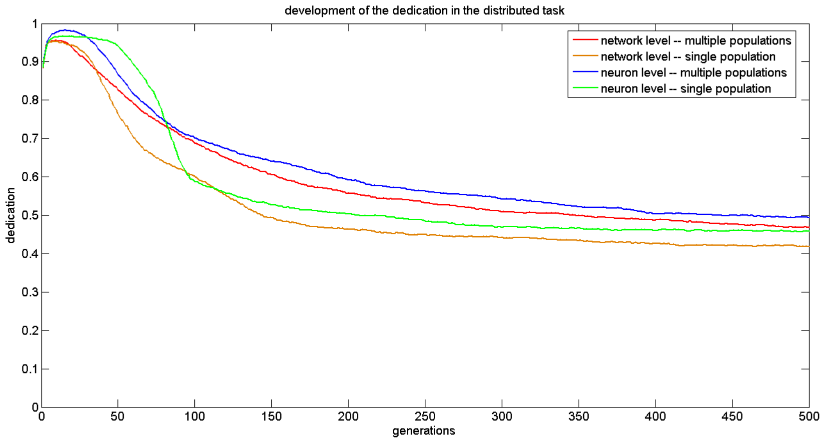

Figure 11: The development of dedication in the distributed task.

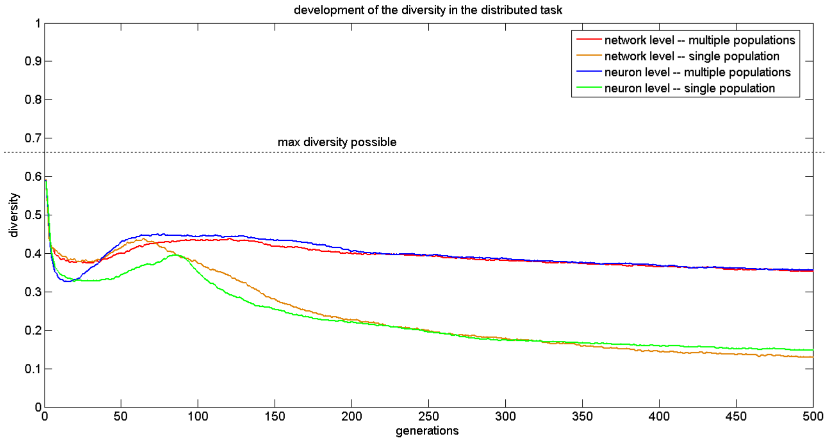

Figure 12: The development of diversity in the distributed task.

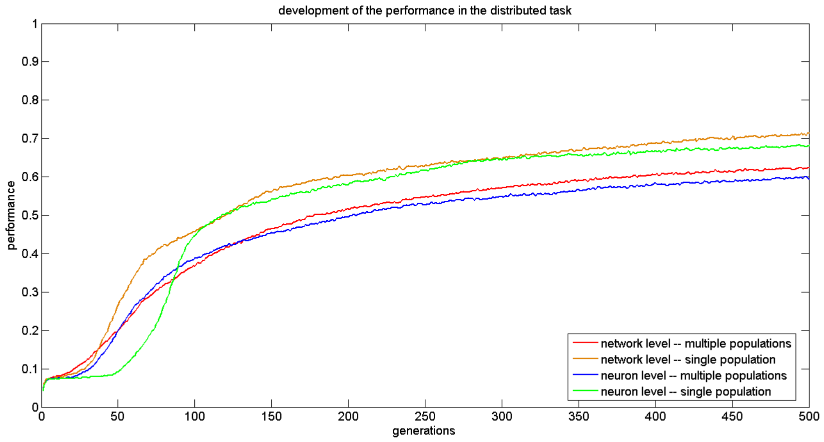

Figure 13: The development of performance in the distributed task.

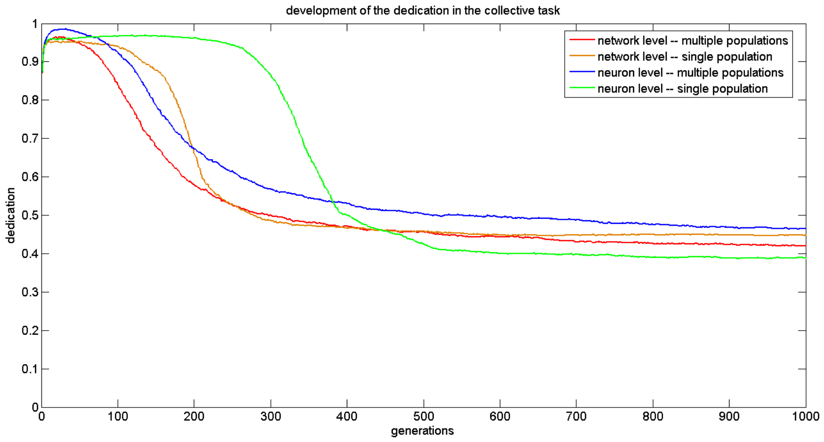

Figure 14: The development of dedication in the collective task.

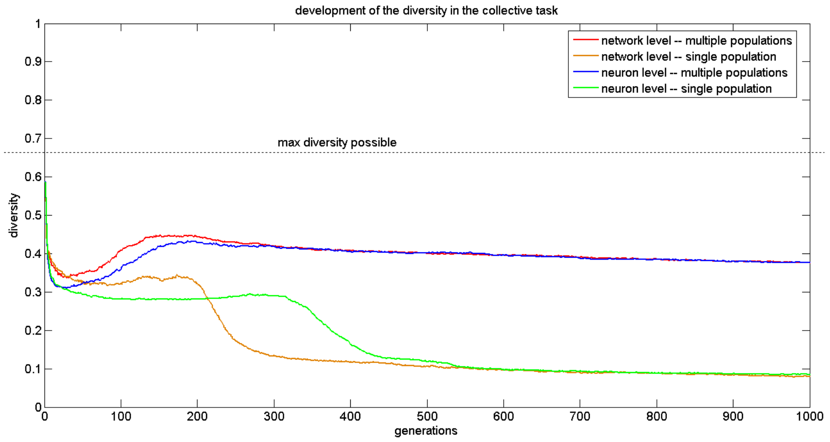

Figure 15: The development of diversity in the collective task.

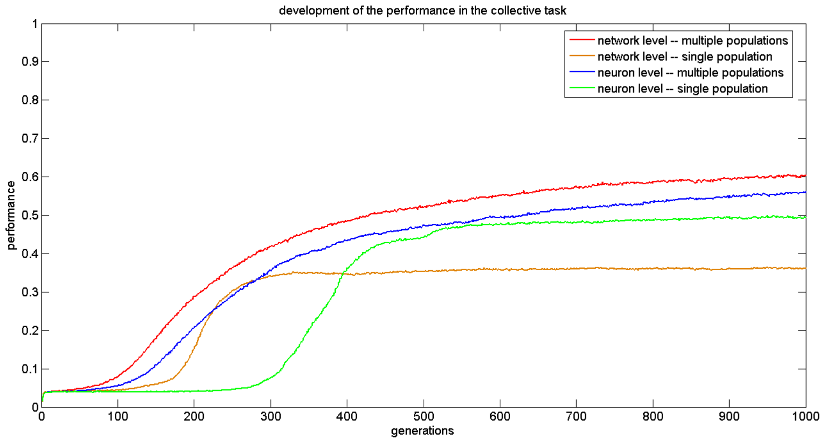

Figure 16: The development of performance in the collective task.

to a limited share of labor and the group becomes more and more diverse. It is clear that for both tasks the progress of dedication and diversity does not comply with these presumptions. We expected dedication to grow, but it actually starts at almost the highest degree possible and then descends until it reaches a certain minimum. Concerning diversity, from the fact that collectives are initialized heterogeneously, we did expect some degree of behavioral diversity to be present at the very beginning. However, as one can see in the graph, the degree of diversity then already is close to its maximum as well.

A possible explanation for the impressive degrees of dedication and diversity during the first few generations are the primitive behaviors of collective members. It appeared that during the earliest stages, most members repeat at each time step almost exclusively the same action, although this action differs per member. Such a strong domination of the behavioral repertoires by only one action leads to an exceptionally high degree of dedication. After all, provided such simplistic behaviors, a rover's action transition table will be concentrated entirely on a single action transition, namely the transition from the repeated action to itself. This results in a high degree of convergence in its transition table, which then on its turn leads to an eminent degree of dedication. Having said that, then any variation between members in these dominant actions inevitably leads to a high degree of diversity. As each member strictly focuses itself on just a single action transition, while the transition differs per member, each action's cross-rover transition histogram will be strongly concentrated on just a few rovers. The resulting high degrees of convergence aggravate the degree of diversity. Therefore, right at the beginning of the specialization process, the collective's degrees of dedication and diversity already will be greater than we may ever expect for the evolutionary end products.

In terms of quantification, the development of a heterogeneous evolutionary specialization process cannot be characterized by growing degrees of dedication and diversity, but instead by ensuring certain minimum levels. One might say that strong dedication and diversity are signs of primitive behaviors, whereas weak dedication and diversity indicate complex behaviors. So, in the course of the process, the collective members learn to extend their behavioral repertoires and temper excessive dedication and diversity, while yet they are prevented from becoming overly complex. Instead of collectives simplifying complex adequate behaviors, which indeed would lead to growing dedication and diversity, we have collectives extending primitive inadequate behaviors. Although upcoming dedication and diversification may play an important role in homogeneously initialized specialization processes, it seems that the essence of evolutionary specialization is more like channeling already existent dedication and diversity, so as to enable fruitful cooperation, and to keep collective members from becoming as complex as their non-specialized counterparts.

The "inverted" development of dedication and diversity does introduce some complications for the design of a specialization measure however. As an evolutionary specialization process advances, one does expect performance to rise, but dedication and diversity to decline. The correlations between dedication and performance on the one hand, and diversity and performance on the other, will thereby always be negative if measured over evolutionary time, no matter if dedication and diversity are known to enhance the performance – as if any order of dedication or diversity would impede the evolving collective's performance. Clearly, the latter will not generally be true. However, for any heterogeneous evolutionary specialization process, it is simply impossible to distinguish the tempering of dedication and diversity due to growing complexity of behaviors from the strengthening caused by emergence of different roles and castes. Consequently, the correlation between the developments of dedication and diversity on the one hand and the performance on the other will be unsuitable for heterogeneous evolutionary specialization. As such, both specialization measures proposed earlier in section 8.4 are unusable.

11.4 Unsteady Development of Single-Population Methods

One might have noticed that for either rover task the single-population methods show a fairly unsteady development. Whereas the multi-population methods show a fluent development for each of the statistics, especially the development of the single-population neuron-level method seems unstable. A possible explanation may be that multi-population methods include a genotype population for each rover separately, in contrast to single-populations methods evolving a single population, from which all rovers are to be constructed.

The cumbersome genotype population of the single-population methods then has great difficulty achieving a sufficient degree of consistency. On the one hand, the nature of our rover tasks causes the genotypes to differentiate and thus become incompatible with each other, but on the other hand, these genotypes are nevertheless all contained by a single population, so may still be recombined. The recombination of highly fit but incompatible genotypes then still produces improper descendants. Note that this issue closely resembles the expectancy issue concerning Moriarty's SANE method [23], mentioned in section 4.3.2.

As long as there is a lack of consistency in the genotype population, there also will be no consistency in the composition of rover collectives. Whereas in the multi-population methods each position in the rover group is occupied by a rover constructed from invariably the same subset of genotypes, this will never be the case in single-population methods, since these contain just a single set. It may then happen that rover collectives are composed of highly competent but incompatible rovers. Even if a rover has made a great contribution to one collective's performance, it may yet not stand out well in the context of another collective. This clearly will corrupt the accuracy of the genotype evaluation. Only if the genotype population in its entirety shows a certain degree of consistency, the consistency of the derived rover collectives may grow.[6]

The multi-population methods omit the issue troubling the single-population methods in the same way Gomez's ESP method [19] omits SANE's expectancy issue. The segregation of genotypes in the multi-population methods allows the populations to become increasingly consistent. As each position in the rover groups is occupied by a rover constructed from invariably the same subset of genotypes, the collective may slowly develop certain expectancies regarding each position, while at the same time the separate populations by themselves grow in consistency as they become increasingly compliant to these expectancies. The separate populations differentiate from each other, but the genotypes within each population converge and become more and more compatible. The recombination of highly fit genotypes then is likely to produce descendants that are highly fit as well.

Consequently, as rover collectives are composed by constructing one rover from each genotype population, a certain consistency will grow in the composition of the rover groups. Rovers that made a great contribution to the performance of one rover group then probably will do so within other collectives as well. However, note that population convergence may yet also be dangerous, in the sense that premature convergence eventually impedes a population's further evolution. As in the course of the evolutionary process the separate populations converge and grow increasingly consistent, genetic diversity will inevitably decay. If a (nearly) optimal solution already has been found, strong convergence prevents an evolutionary process from substantially drifting away from its solution. However, as long as no optimal solution has been found yet, strong convergence is undesirable, as it inhibits the effectiveness and efficiency of the evolutionary process by hindering the exploration of its solution space and restraining its genetic adaptability.

---

[6] Although we cannot provide for a sound verification, it might be the case that a certain degree of consistency in the genotype population evokes the sudden turning points in the development of the several statistics.

12 Quantitative Comparison of Methods

Section 11.3 already pointed out that the proposed specialization measure based on the correlation between diversity and performance appeared unsuitable for quantifying specialization during a heterogeneous neuro-evolutionary specialization process. The second best option then is to apply the measures for dedication and diversity independently from the specialization measure.

In order to make a statistically sound comparison between the four neuro-evolutionary methods, for either rover task we performed ten runs per method ($n = 10$). From each run's final generation several statistics were collected, so as to obtain per statistic a mean and deviation for each of the methods. By applying a t-test with $\alpha = 0.05$ to these means and deviations, for all statistics we determined whether the differences between the evolutionary methods are statistically significant. After some tuning, we created such circumstances that the rover collectives were able to achieve moderate performances, so that any comparison involving performance would not be affected by either floor or ceiling effects.

12.1 Distributed Rover Task

Multi-population methods Between the two multi-population methods, there existed no significant differences for any of the statistics. (See table 4)

Single-population methods The single-population neuron-level method preserved just a slightly higher degree of dedication (0.4589 ± 0.0430) than the single-population network-level did (0.4197 ± 0.0072). On the other hand, their degrees of diversity as well as their performances were statistically equal.

Network-level methods Even though the difference between the degrees of dedication was statistically significant, this difference yet was not very impressive. Concerning the degrees of diversity, however, the multi-population network-level method yielded a much higher degree of the diversity (0.3838 ± 0.0199) than the single-population network-level method (0.1448 ± 0.0120). Despite its reduced dedication and diversity, the single-population neuron-level method still achieved a significantly better performance (0.7137 ± 0.0337) than the multi-population network-level method did (0.6220 ± 0.0384).

Neuron-level methods The same is true for the neuron-level methods. The difference between the degrees of dedication was not very impressive again, but the multi-population neuron-level method yielded a much higher degree of diversity (0.4077 ± 0.0375), compared to the single-population neuron-level method (0.1543 ± 0.0289). The performance of the single-population neuron-level method (0.6807 ± 0.0381) was significantly better than that of the multi-population network-level method (0.6025 ± 0.0274).

From the data appears that single-population methods show greater convenience with the distributed task than multi-population methods. Even though dedication experiences a different development using single-population methods than using multi-population methods, in the end it turns out to be fairly indifferent towards the method for evolution. Regarding behavioral diversity, however, collectives evolved by either of the two multi-population methods tend to become much more diverse than those evolved by single-population methods do. For the development of each of these three statistics counts that multi-population methods show a smoother development than single-population methods.

level	populations	dedication	diversity	performance
network	multiple	0.4671 ± 0.0302	0.3519 ± 0.0242	0.6220 ± 0.0384
network	single	0.4197 ± 0.0072	0.1301 ± 0.0101	0.7137 ± 0.0337
neuron	multiple	0.4938 ± 0.0376	0.3562 ± 0.0326	0.6025 ± 0.0274
neuron	single	0.4589 ± 0.0430	0.1491 ± 0.0306	0.6807 ± 0.0381

Table 4: A summary of the three most important statistics for the several neuro-evolutionary methods in compliance with the distributed rover task.

12.2 Collective Rover Task

Multi-population methods The multi-population methods showed a small but statistically significant difference between the degrees of dedication. The degrees of diversity were similar. Nevertheless, the multi-population network-level showed significantly better performance (0.6059 ± 0.0456) than the multi-population neuron-level method (0.5590 ± 0.0482). (See table 5)

Single-population methods For the single-population methods, the difference between the degrees of diversity was insignificant. The single-population network-level method preserved a significantly higher degree of dedication, but did not perform as well as its neuron-level counterpart. The single-population neuron-level method achieved a performance of 0.4970 ± 0.1231; the single-population network-level method stuck to a performance of 0.3638 ± 0.1349. Both methods seem to struggle with considerable deviation in performance.

Network-level methods The multi-population network-level method yielded a much higher degree of diversity (0.3754±0.0201), compared to the degree of diversity yielded by the single-population network-level method (0.0798 ± 0.0114). As well, the multi-population variant did perform considerably better than the single-population variant.

Neuron-level methods Regarding the neuron-level methods, both the degrees of dedication and the degrees of diversity were significantly different, with degrees of 0.4654 ± 0.0264 respectively 0.3768 ± 0.0118 for the multi-population network-level method and 0.3866 ± 0.0326 respectively 0.0835 ± 0.0155 for the single-population network-level method. These differences did not lead to a statistically significant difference in performance. However, the fact that the multi-population neuron-level method does not significantly outperform the single-population network-level method might be due to the considerable amount of deviation for the latter method.

Regarding the collective task, the multi-population network-level method outperformed all other methods. Its single-population counterpart on the other hand showed least impressive performance. The performances of the two neuron-level methods lay in between those of the network-level methods and were statistically equal. Despite the disparity in development, the degrees of dedication eventually turned out to be more or less comparable. Similar to the distributed task, collectives evolved by multi-population methods tend to become significantly much more diverse than those evolved by single-population methods do. For the collective rover task counts as well that multi-population methods show a much smoother development than single-population methods.

level	populations	dedication	diversity	performance
network	multiple	0.4203 ± 0.0197	0.3754 ± 0.0201	0.6059 ± 0.0456
network	single	0.4475 ± 0.0764	0.0798 ± 0.0114	0.3638 ± 0.1349
neuron	multiple	0.4654 ± 0.0264	0.3768 ± 0.0118	0.5590 ± 0.0482
neuron	single	0.3866 ± 0.0326	0.0835 ± 0.0155	0.4970 ± 0.1231

Table 5: A summary of the three most important statistics for the several neuro-evolutionary methods in compliance with the collective rover task.

12.3 Multi-Population vs Single-Population

In accordance with the distributed rover task, the single-population methods evolved better collectives than the multi-population methods. Concerning the collective rover task, however, the roles were reversed, as for this task the multi-population methods showed to be more suitable than the single-population methods. The multi-population methods did not only outperform the single-population methods by achieving better performances, but also by showing greater reliability, since both single-population methods seem to suffer from a considerable amount of deviation. The segregation of genotypes as such holds back the performance with respect to the distributed task, but nonetheless enhances the performance in compliance with the collective task.

For either task, the segregation of genotypes evidently leads to a higher degree of behavioral diversity, as well as that it might have a slightly positive effect on the degree of dedication. The question is however to what extent evolutionary search may influence the development of dedication and diversity. Since evolutionary specialization methods are to search for optimal collective behaviors, one may expect that the degrees of dedication and diversity are optimized as well. Apparently, this is not necessarily the case, as the multi-population methods consistently preserve an amount of diversity substantially greater than the single-population methods, even though that leads to inferior performance with respect to the distributed task. The other way round, the single-population methods seem incapable of maintaining an appropriate degree of diversity in compliance with the collective task, even when the multi-population methods show that higher degrees of diversity may lead to better performance.

Both the multi-population methods and the single-population methods thus are not always capable of finding the optimal degree of behavioral diversity. The same might be true for the degree of dedication, but since the variances in dedication are not that pronounced for either rover task, this cannot be verified. Yet, it appears that the organization of genotypes has such a strong impact on the development of collectives, and thereby also on the development of the dedication and diversity, that evolutionary algorithms may only manipulate the collectives and their dedication and diversity within a very limited range. A more flexible organization of genotypes may therefore turn out to be helpful.

12.4 Network-Level vs Neuron-Level

One of the key features of the neuron-level methods is that neuron-level evolution allows for a more fine-grained view on the contribution of individual neurons. Network-level methods evaluate neurons within the context of invariably the same neural network. Highly fit neurons may then easily be concealed by poor performance of other neurons in its network, as well as that poor neurons may take advantage of the greater performance of other neurons. Neuron-level methods evaluate each neuron within the context of multiple networks, so as to make the evaluation of each neuron less dependent of the networks it was part of. This must then result in greater accuracy, speeding up the evolutionary process.

Be sure to note however, that this improved accuracy does require multiple evaluations per generation. Considering tasks such as the rover problem, for which evaluations are computationally very expensive, these multiple evaluations turn out to be a considerable weakness. Whereas for network-level evolution a single evaluation cycle per generation will be sufficient, since these neurons cannot be evaluated within the context of multiple networks anyway, neuron-level methods need multiple evaluation cycles. Assuming that the performance of the rover tasks accounts for by far the larger part of computation, under furthermore equal circumstances, a neuron-level process performing four evaluations per generation will take about as much as four times the computational time as a network-level process.[7]

The comparison between network-level evolution and neuron-level evolution shows that the advantage from greater accuracy is quite relative, as apparently it led to higher performance for neither the distributed rover task nor the collective rover task. Although the single-population neuron-level method outperformed the single-population network-level method on the collective task, yet the multi-population network-level method again showed a performance significantly superior to both neuron-level methods. Considering that the neuron-level methods could not turn the additional computational expenses into an increased performance compared to the network-level methods, we may doubtlessly conclude that network-level evolution is more effective with respect our rover tasks.

Furthermore, based on our results, there is no reason to believe that the level of evolution largely affects either the amount of dedication or the amount of diversity. Although for both statistics it is true that, dependent on the level of evolution, their development may substantially differ during the evolutionary process, especially for the single-population methods, yet at the end of the process either statistic seems to be indifferent towards the level of evolution.

---

[7] Nonetheless, in our experiments, both the network-level methods and the neuron-level methods performed four evaluations per generation, so as to preserve possible side-effects.

Part V

Conclusion

In the conclusion section, our findings are summarized. The evolutionary processes showed a development quite different from what we presumed. Thereby, we were unable to use the proposed specialization measure. As we still wanted to form an impression of how strong the several evolutionary methods stimulate specialization, the next best option was to compare dedication, diversity, and performance independently.

13 The Course of Specialization

At the beginning of each evolutionary specialization process, the genotypes were initialized randomly, so as to facilitate for proper evolution. However, such an approach appeared to have considerable consequences for the behavioral development of collective members. During the earlier stages of the evolutionary process, the members of evolving collectives often conducted an improper, primitive behavior. Although a few members might have been little more sophisticated, the greater part did invariably repeat a particular action for all of the time steps, where this action differed per individual. Such a strong imbalance in the behavioral repertoires led to an overly strong degree of dedication. Variation across members in these dominant actions resulted in an overly strong degree of behavioral diversity.

In the further course of the process, the collective members learn to extend their behavioral repertoires and temper the excessive degrees of dedication and diversity, while specialization keeps them from becoming as complex as their non-specialized counterparts. Although upcoming dedication and diversification may play an important role in (homogeneously initialized) non-evolutionary specialization, it seems that the essence of evolutionary specialization is more like learning how to make optimal use of already existent dedication and diversity in individual preferences and capabilities, so as to enable fruitful cooperation.

14 Consequences for Quantifying Specialization

The inverted development of dedication and diversity seriously affects the quantification of specialization. Although members undoubtedly will learn to extend their primitive behavioral repertoire in the course of the specialization process, initial behaviors still introduce considerable complications for the design of an adequate specialization measure. Strongly unbalanced behavioral repertoires will lead to high degrees of dedication, while variation in dominant actions across members leads to high degrees of behavioral diversity. Therefore, the degrees of dedication and diversity are at the beginning of the process already greater than ever to be expected for the evolutionary end products. As the process carries on and behaviors become more and more complex, the degrees of dedication and diversity decline, down to a certain minimum.

The point then is that during the evolutionary process, one does expect the performance to rise, but the dedication and diversity to decline. Both the correlations between dedication and performance on the one hand and diversity and performance on the other, will thus invariably be negative if measured over evolutionary time, even in task domains for which dedication and diversity are known to enhance performance. As such, one cannot simply correlate the developments of dedication and diversity to performance. Neuro-evolutionary specialization is a promising method for evolving heterogeneous collectives, but certainly needs some metric for the quantification of specialization. Further investigation of such a metric might thus be valuable.

15 Comparison of Methods

In the distributed rover task, the single-population methods evolved better collectives than the multi-population methods. In the collective task, however, the multi-population methods showed to be more suitable than the single-population methods. The segregation of genotypes thus holds back the performance on the distributed task, but nonetheless enhances rover performance in the collective task. Both the multi-population methods and the single-population methods thus were incapable of consistently finding the optimal degree of dedication and diversity. An explanation might be that the organization of genotypes has such a strong impact on the development of collectives, that an evolutionary algorithm may only manipulate the collectives – and their dedication and diversity – within a very limited range. A more flexible segregation of genotypes might perhaps overcome this problem.

For the comparison between the network-level methods and the neuron-level methods, one must take into account that neuron-level methods normally consume considerably more computation than network-level methods. Neuron-level evolution requires multiple evaluations per generations, whereas for network-level evolution just a single evaluation will suffice. Considering that the neuron-level methods could not turn these additional computational expenses into an increased performance compared to the network-level methods, we may doubtlessly conclude that the network-level methods were more effective with respect to our rover tasks. Furthermore, based on our results, there is no reason to believe that the level of evolution does largely affect either the amount of dedication or the amount diversity.

References

[1] Adrian Agogino and Kagan Tumer. Efficient evaluation functions for multi-rover systems. In *GECCO '04: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 1–12, 2004.

[2] Adrian Agogino, Kagan Tumer, and Risto Miikkulainen. Efficient credit assignment through evaluation function decomposition. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1309–1316, New York, NY, USA, 2005. ACM Press.

[3] Adrian Kujaneck Agogino. *Design and Control of Large Collections of Learning Agents.* PhD thesis, The University of Texas at Austin, 2003.

[4] Tucker Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Auton. Robots*, 8(3):209–238, 2000.

[5] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems.* Oxford University Press, Inc., New York, NY, USA, 1999.

[6] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. In *Biological Sciences*, volume 263, pages 1565–1569, November 1996.

[7] Bobby D. Bryant and Risto Miikkulainen. Neuroevolution for adaptive teams. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, 2003.

[8] Nicholas W. Calderone and Robert E. Page. Genotypic variability in age polyethism and task specialization in the honey bee, apis mellifera (hymenoptera: Apidae). *Behavioral Ecology and Sociobiology*, 22(1):17–25, January 1988.

[9] Robert J. Collins and David R. Jefferson. Selection in massively parallel genetic algorithms. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[10] Kenneth Alan De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[11] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225, 1991.

[12] Jeffrey L. Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–9, 1993.

[13] Scott E. Fahlman and Christian Lebiere. *The cascade-correlation learning architecture*, pages 524–532. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[14] Jacques Gautrais, Guy Theraulaz, Jean-Louis Deneubourg, and Carl Anderson. Emergent polyethism as a consequence of increased colony size in insect societies. *Journal of Theoretical Biology*, 215(3):363–373, April 2002.

[15] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.

[16] F. Gomez and R. Miikkulainen. Robust non-linear control through neuroevolution, 2002.

[17] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adapt. Behav.*, 5(3-4):317–342, 1997.

[18] Faustino J. Gomez and Risto Miikkulainen. Solving non-markovian control tasks with neuro-evolution. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1356–1361, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[19] Faustino John Gomez. *Robust non-linear control through neuroevolution.* PhD thesis, The University of Texas at Austin, 2003. Supervisor-Risto Miikkulainen.

[20] Jeffrey Horn, David E. Goldberg, and Kalyanmoy Deb. Implicit niching in a learning classifier system: Nature's way. *Evolutionary Computation*, 2(1):37–66, 1994.

[21] Ling Li, Alcherio Martinoli, and Yaser S. Abu-Mostafa. Learning and measuring specialization in collaborative swarm systems. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 12(3-4):199–212, 2004.

[22] David E. Moriarty and Risto Mikkulainen. Efficient reinforcement learning through symbiotic evolution. *Mach. Learn.*, 22(1-3):11–32, 1996.

[23] David Eric Moriarty. *Symbiotic evolution of neural networks in sequential decision tasks*. PhD thesis, The University of Texas at Austin, Austin, TX, USA, 1998.

[24] Shin I. Nishimura and Takashi Ikegami. Emergence of collective strategies in a prey-predator game model. *Artif. Life*, 3(4):243–260, 1997.

[25] C. Noirot and J. M. Pasteels. Ontogenetic development and evolution of the worker caste in termites. *Experientia*, 43(8):851–860, 1987.

[26] S O'Donnell and RL Jeanne. Forager specialization and the control of nest repair in polybia occidentalis olivier (hymenoptera : Vespidae). *Behavioral Ecology and Sociobiology*, 27(5):359–364, September 1990.

[27] Mitchell A. Potter, Lisa Meeden, and Alan C. Schultz. Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists. In *IJCAI*, pages 1337–1343, 2001.

[28] Matt Quinn. A comparison of approaches to the evolution of homogeneous multi-robot teams. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 128–135, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 2001. IEEE Press.

[29] G E Robinson. Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37(1):637–665, 1992.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[31] H Seligmann. Resource partition history and evolutionary specialization of subunits in complex systems. *Biosystems*, 51(1):31–39, July 1999.

[32] Robert E. Smith, Stephanie Forrest, and A. S. Perelson. Searching for diverse, cooperative subpopulations with genetic algorithms. *Evolutionary Computation*, 1(2):127–149, 1993.

[33] Peter Stone and Manuela Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artif. Intell.*, 110(2):241–273, 1999.

[34] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.

[35] G. Theraulaz, E. Bonabeau, and J. Deneubourg. Response threshold reinforcement and division of labour in insect societies, 1998.

[36] K. Tumer, A. Agogino, and D. Wolpert. Learning sequences of actions in collectives of autonomous agents, 2002.

[37] T. Wenseleers, F.L.W. Ratnieks, and J. Billen. Caste fate conflict in swarm-founding social hymenoptera: an inclusive fitness analysis. *Journal of Evolutionary Biology*, 16(4):647–658, July 2003.

[38] D. Whitley, K. Mathias, and P. Fitzhorn. Delta coding: An iterative search strategy for genetic algorithms,. In Rick Belew and Lashon Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 77–84, San Mateo, CA, 1991. Morgan Kaufman.

[39] Ronald J. Williams and David Zipser. Experimental analysis of the real-time recurrent learning algorithm. *Connection Science*, 1(1):87 – 111, 1989.

[40] Chern H Yong and Risto Miikkulainen. Cooperative coevolution of multi-agent systems. Technical report, The University of Texas at Austin, Austin, TX, USA, 2001.