Assessing Hierarchical Cooperative CoEvolution

Michail Maniadakis and Panos Trahanias
Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
Heraklion, Crete, Greece
e-mail:{mmaniada,trahania}@ics.forth.gr

Abstract

Recently, many research efforts are directed towards coevolutionary algorithms. The present work aims at the assessment of Hierarchical Cooperative CoEvolution (HCCE) being properly formulated to address hierarchical problems where simple components having separate design objectives, are parts of other more complex ones. HCCE is able to highlight the specialties of each component and additionally enforce their successful integration in a composite structure. Here we present HCCE describing also the internal dynamics that provide its effectiveness in addressing difficult distributed design problems. Additionally, the results described in the present work attest to its validity and superior performance against ordinary Unimodal evolution, and Enforced SubPopulation coevolution.

1. Introduction

Evolutionary techniques are often employed as a tool for designing complex computational systems. However, in the majority of real-life applications, partial entities can be identified, which together compose the overall picture of the problem. These entities often follow different design objectives as it is indicated by their specialized role in the composite system. As a result, by utilizing ordinary evolutionary schemes employing a single representation to map complete problem solutions to the genotype, the structural nature of the problem is usually underestimated, or overlooked. In order to be more effective, partial components should be explicitly considered by the evolutionary process, facilitating the exploration of their structure.

This is particularly the approach followed by coevolutionary algorithms which utilize separate populations to evolve partial entities of the problem [14, 17]. In order to formulate a composite problem solution, individuals within different populations have to be selected, put together and operate in parallel [1, 13]. Each population can use its own evolutionary parameters (e.g. encoding, genetic operators). Accordingly, increased search competencies are inherently available in coevolutionary algorithms.

In general, coevolutionary approaches are classified in competitive and cooperative. In the first case, an antagonistic scenario is followed with one population trying to bit the other. In the second case a synergistic scenario is followed with the evolution of partial populations adjusting on one another, formulating a complex system with globally successful functionality [14]. In the current study, we concentrate on cooperative schemes being appropriate to approach problems with explicit notions of modularity.

The majority of existing cooperative coevolutionary schemes have been applied in designing distributed structures consisting of smaller components [1, 4, 6, 8, 15]. However, these schemes are not able to sufficiently investigate hierarchically organized problems where simple components are parts of other more complex ones, formulating gradually more composite structures. Along this line, we have recently introduced a Hierarchical Cooperative Co-Evolutionary scheme that is capable of approaching problems described by multiple levels of modularity [11, 12, 10].

Hierarchical Cooperative CoEvolution (HCCE) aims at both highlighting the independent roles of partial components at various levels, and additionally enforcing their coupled operation as a globally integrated system. This is achieved by evolving separate populations for each component of the system, and additionally evolving populations encoding assemblies of components. These two different kinds of evolutionary processes are hierarchically organized. The evolutionary processes at lower levels investigate the structure of partial system components. They are driven by their own dynamics, trying to meet the special design objectives of each component. At the same time, the evolutionary process at the higher level, explores the integrated performance of substructures trying to identify those component structures that can successfully cooperate. This higher level assembly formulation process tunes lower level component design procedures favoring the structures with the best cooperative performance. This is achieved by means of a newly introduced genetic operator named “Replication”. The architecture of multiple coevolutionary processes tuned by a higher level evolution, can be repeated for as many levels as necessary, formulating distributed systems consisting of many components.

HCCE has been extensively employed to design a series of different distributed cognitive systems (e.g. [11, 12, 10]), but it could be also used in other distributed problems such as designing teams of heterogeneous but cooperating robots, designing distributed mechanical systems consisting of different substructures, etc.

The current study assesses the effectiveness of HCCE scheme on the design of complex distributed systems. Particularly, we perform a series of tests validating the robustness of the hierarchical coevolutionary scheme, and we compare it with (i) Unimodal evolution and (ii) Enforced SubPopulation (ESP) coevolution [7]. Additionally, we test the efficiency of the “Replication” genetic operator, that has been introduced to support the successful convergence of the coevolutionary procedure.

The rest of the paper is organized as follows. In the next section, we present the Hierarchical Cooperative CoEvolutionary (HCCE) scheme. Then, we present the experimental procedure and the obtained results emphasizing on the effectiveness and the internal dynamics of the HCCE scheme. In the last two sections, we discuss the obtained results and we formulate suggestions for future work.

2. Hierarchical Cooperative CoEvolution (HCCE)

Cooperative coevolutionary approaches involve many interactive populations which evolve the components of a distributed structure. Partial *populations* are usually referred as *species* in the coevolutionary literature, and thus these two terms will be employed interchangeably henceforth.

In the majority of existing approaches, individuals of a species cooperate with the best individual from the other species or a randomly selected set of cooperators (e.g. [1, 8, 14]). This is also the case with Pareto coevolutionary approaches [2, 5, 6, 9] which mainly aim at balancing between different criteria in multi-objective optimization problems. The issue of collaborator selection is very important, especially when investigating large systems consisting of many components.

In order to address the collaborator selection issue, a higher level optimization process can be used that searches within species identifying the individuals with the best coupled performance. This higher level search can be implemented by means of one more evolutionary process [2, 15]. The process of simultaneous evolution of partial components and assemblies of components, can be organized hierarchically, formulating a multiple level scheme consisting of gradually more complex assemblies. The hierarchical organization facilitates the solution of problems described by multiple levels of modularity, where simple components are parts of other more complex ones. In our previous works we have presented a Hierarchical Cooperative CoEvolutionary (HCCE) architecture able to design complex systems consisting of a large number of components [11, 12]. Besides the evolution of species corresponding to partial components, the proposed HCCE scheme employs additional higher level evolutionary processes, to select the proper individuals from each species that cooperatively are able to construct effective component assemblies. These configurations are used as a basis to guide the composite coevolutionary process since individuals are more likely to be members of effective assemblies of cooperators.

To the best of our knowledge, there is only one work presenting a hierarchical cooperative coevolutionary method in the literature [4]. In contrast to [4], our approach is capable of coevolving larger assemblies of cooperating species, while at the same time, emphasizes the independence of substructures utilizing multiple and potentially separate criteria to guide partial evolutionary processes. Other approaches on hierarchical problem solving investigate the design of system components sharing the same set of objectives, and thus, they can not address effectively distributed systems consisting of heterogeneous substructures [16, 3, 2]. Additionally, they don’t utilize specialized subpopulations that have been proved to facilitate significantly the evolutionary process [8].

The HCCE scheme employs two different kinds of species encoding the configurations of either a Primitive Structure (PS) or a Coevolved Group (CG). PS species are utilized to explore the structure of system components. A CG consists of a group of PSs, all of them having common design objectives. Thus, CGs specify configurations of partial solutions, encoding assemblies of individuals. The evolution of CG modulates partly the evolutionary process of its lower level PS species to enforce their cooperative performance. A CG can also be a member of another more complex CG. Consequently, several CGs can be organized hierarchically, with the higher levels enforcing the cooperation of the lower ones. This organization is explained by means of an example (Fig 2). We assume the existence of a system consisting of two partial structures and three links facilitating the flow of information (Fig 1(a)). One hypothetical HCCE process employed to specify the structure of the composite system is illustrated in Fig 1(b). A snapshot of the exemplar HCCE process described above is illustrated in Fig 2.

Furthermore, the HCCE scheme supports the consideration of system operation in conditions of partial failure. Specifically, the deactivation of a CG together with its lower

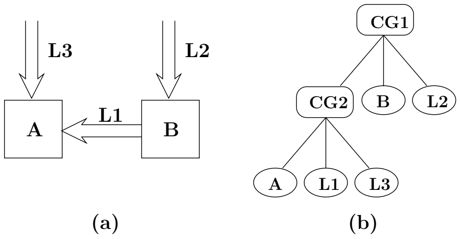

Figure 1. Hierarchical cooperative coevolutionary design. Part (a) represents schematically a hypothetical connectivity of system components. Part (b) represents the hierarchical coevolutionary scheme utilized to evolve partial structures.

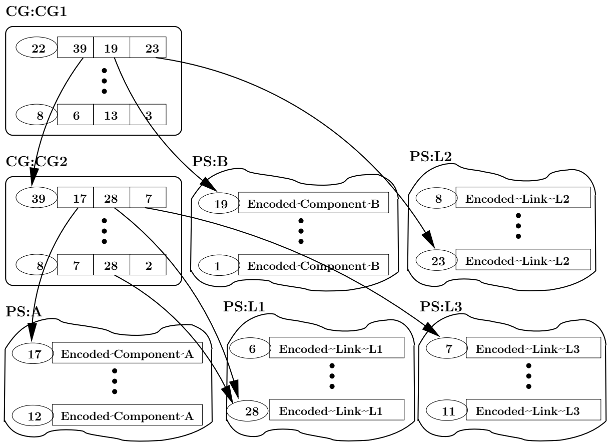

Figure 2. An overview of the hierarchical coevolutionary scheme, with CG species tuning the evolutionary processes of PS species.

level PS species, can easily simulate the elimination of the respective system components. Additionally, appropriate fitness functions can be specified for the evolution of partial structures, in order to indicate the performance of the model when all substructures are present, and also indicate the performance when some partial structures are eliminated. By considering system performance in diverse operating conditions, important insight of the internal mechanisms of the system is provided to the designer. In particular, the exploration of system performance after the removal of components, can be very beneficial for inferring the roles of partial structures in the composite model and the dynamics of their interaction. It is worth emphasizing that the exploration of system performance in conditions of partial failure is very important for distributed systems consisting of autonomous components (e.g. agent-based systems) because it is capable of highlighting those functionalities of the system which are most affected after the collapse of one or more components.

Based on the hierarchical organization of the problem, HCCE utilizes separate fitness measures for different partial evolutionary processes. Specifically, for each species s, a fitness function f_s is defined to drive its evolution. Different CGs can be driven by different fitness functions. However, all PS species under a CG share a common f_s. A partial fitness function $f_{s,t}$ evaluates an individual according to criteria t. The composite fitness value is given by:

$$f_s = \prod_t f_{s,t} \tag{1}$$

Furthermore, the cooperator selection process at the higher levels of hierarchical coevolution will probably select an individual to participate in many assemblies (e.g. the case of individual 28 of PS species L1, of Fig 2). Let us assume that an individual participates in K assemblies. Then, it will get K fitness values $f_{s,t}$, regarding the fulfillment of criteria t. The overall ability of the individual to meet the t-th criteria is estimated by:

$$f_{s,t} = max_k\{f_{s,t}^k\} \tag{2}$$

where $f_{s,t}^k$ is the fitness value of the k-th solution formed with the membership of the individual under discussion.

A common problem for the coevolutionary approaches evolving assemblies of cooperators, is related to the multiple participation of some individuals in many different collaborator assemblies, while at the same time others are offered no cooperation at all [6, 15]. A large number of multiple cooperations is generally a drawback for the coevolutionary process, because different cooperator assemblies could demand evolution of the same individual in different directions. Non-cooperating individuals can be utilized to decrease the multiplicity of cooperations for those individuals which are heavily reused. We have introduced a new genetic operator termed Replication, addressing the issue of multiple cooperations [12]. In short, for each unused individual x of a species, replication identifies the fittest individual y with more than max_c cooperations. The genome of y is then copied to x, and x is assigned $max_c - 1$ cooperations of y, by updating properly the CG population at the higher level. After replication, individuals x and y are allowed to evolve separately following independent evolutionary directions. This is illustrated in Fig 3, for the case of one CG and one PS species. Initially, individual 14 of the PS population participates in five solution assemblies, while individual 29 is offered no collaboration at all. Replication copies the chromosome of 14 to 29, and redirects two of the collaboration indicating pointers to 29. In the following evolutionary generations, individuals 14 and 29 follow separate evolutionary directions, facilitating the exploration of the search space.

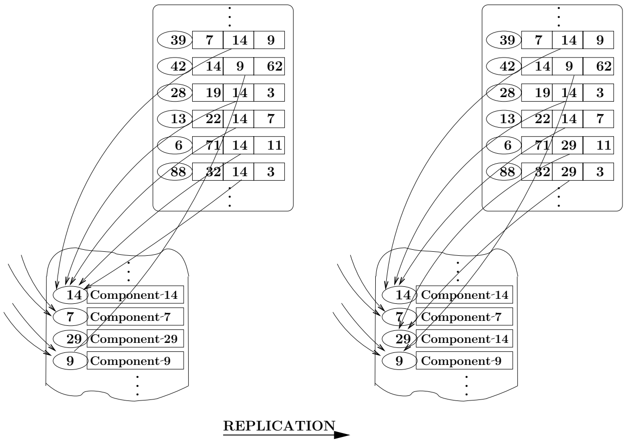

Figure 3. Schematic representation of the replication operator ($max_c = 3$).

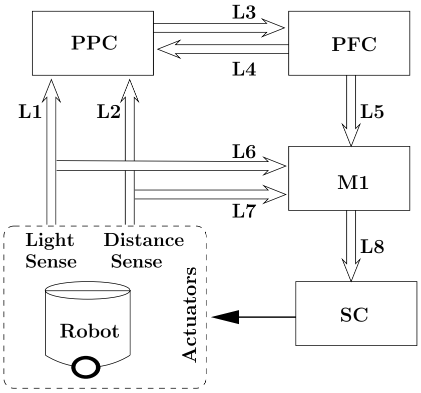

Figure 4. A schematic overview of the computational model. Cortical agents are illustrated with blocks, while link agents are illustrated with a double arrow.

Evolutionary steps are performed separately but synchronously for each species of the HCCE scheme. This is shortly described below. First, individuals are sorted according to their fitness values. Then, Replication is applied to reduce multiple cooperations. Next, a predefined percentage of individuals are probabilistically crossed over. An individual selects its mate from the whole population, based on their accumulative probabilities. Finally, mutation is performed in a small percentage of the resulted population. This process is repeated for a predefined number of evolutionary epochs, driving the species exploring the structure of system components to the accomplishment of their own design objectives, and additionally driving the higher level cooperator selection processes to the formulation of successful composite structures.

3. Results

The effectiveness of the proposed coevolutionary scheme is assessed on the design of a complex distributed system. This approach is in contrast to other works employing mathematical functions as a test-bed for the study of evolutionary approaches. Particularly for the case of coevolutionary algorithms, mathematical functions based on few independent variables are usually employed [17, 13], decomposing the overall problem in few and very simple entities. However, this approach can not reveal the power of each algorithm and its capability to address difficult problems. Additionally, previous works on hierarchically formulated problems investigated system components sharing common design objectives [16, 3]. Thus they can not tackle systems consisting of heterogeneous substructures, as this is the case with HCCE.

In order to take a better insight of the coevolutionary procedure, the HCCE scheme is tested on the design of a complex brain-inspired cognitive system. The problem of developing brain-like computational structures fits adequately to coevolutionary approaches, because different coevolved species can be used to perform separate design decisions for the substructures representing brain areas, in order to highlight their particular characteristics. An agent-based approach is followed to represent partial brain modules, addressing explicitly their autonomous role in the composite system [12]. Furthermore, following recent trends that study lesions of brain areas, the HCCE scheme supports systematic modelling of biological experiments, evaluating the performance of the model in pre- and post- lesion conditions.

During the modelling process, environmental interaction is of utmost importance, since it is difficult to investigate cognitive system functionality without embedding the model into a body to interact with its environment. Thus, a two wheeled simulated mobile robot is utilized to support environmental interaction, while at the same time the model enriches the behavioral repertory of the robot.

We note that the present series of experiments is an extension of our previous work which demonstrated the development of the cognitive working memory model [11]. Due to space limitations the problem will be described here in short, since the emphasis of the current study is to explore the dynamics of the HCCE design mechanism, rather than the biological reliability of the model.

3.1. Problem Statement

We study a partial brain model, which simulates posterior parietal cortex (PPC) - prefrontal cortex (PFC) - pri-

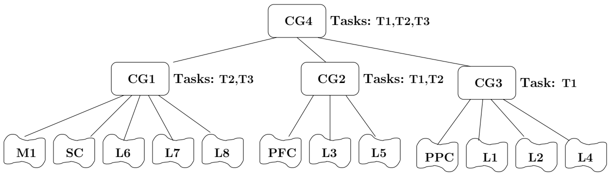

Figure 5. A graphical illustration of the coevolutionary process.

mary motor cortex (M1) - spinal cord (SC) interactions, emphasizing on working memory usage. The architecture of the model is demonstrated in Fig 4. The distributed model consists of 12 neural agents having separate self-organization dynamics. In particular 4 cortical components (represented by blocks) and 8 link components (represented by double arrows) are employed. The components have to develop different roles formulating a single, globally functional artificial system. Overall, the problem investigated here involves the specification of totaly 192 variables interacting in a highly non-linear way. Thus, the current problem can be used as an advanced test case for investigating the dynamics of HCCE, and revealing its beneficial characteristics.

Three tasks are designed to highlight the role of each agent in the model. The first task $T1$ accounts for developing working memory -like neural activity on PFC. The success on $T1$ task is evaluated by the fitness measure E_1. The second task T_2 aims at exploiting working memory in order to accomplish a delayed response task. This is evaluated by fitness measure E_2. The third task T_3, addresses the operation of the model in lesion conditions. Specifically, when simulated lesion is performed on PFC structure, the behavioral repertoire of the artificial organism is reduced, avoiding however collapse of the composite system. In that case, the robot is still able to drive but in a purposeless mode. This is evaluated by fitness measure E_3. Overall, the experimental process discussed above aims at reproducing computationally a biological scenario addressing pre- and post-lesion rat behavior in a T-maze [1]. The measures E_1, E_2, E_3 are described in detail in [11], and thus, they are omitted here due to space limitations.

3.2. Computational Modelling

We turn now to the design of the model by means of the HCCE. The coevolutionary scheme is demonstrated in Fig 5. According to the lesion scenario described above, each agent serves more than one tasks. This is illustrated in Fig 5, at the right side of each CG. Thus, different fitness functions guide the evolution of different species, enforcing the accomplishment of the respective tasks. In particular, individuals are assigned a combination of evaluation indexes, for the accomplishment of tasks where the composite model is performing, and the accomplishment of tasks with performance of the lesioned model. Following the formulation introduced by eqs. (1) and (2), the composite and partial fitness functions employed for the evolution of populations, are summarized in Table 1. It is reminded that all PSs share the same fitness functions with their higher level CG.

Table 1. Different design objectives guide evolution of different partial populations.

Composite Fitness:	$f_{CG1} = f_{CG1,T2} \cdot f_{CG1,T3}$
Partial Fitness:	$f^k_{CG1,T2} = \sqrt{E}_2, \;\; f^k_{CG1,T3} = E_3$
Composite Fitness:	$f_{CG2} = f_{CG2,T1} \cdot f_{CG2,T2}$
Partial Fitness:	$f^k_{CG2,T1} = {E_1}^2, \;\; f^k_{CG2,T2} = \sqrt{E}_2$
Composite Fitness:	$f_{CG3} = f_{CG3,T1}$
Partial Fitness:	$f^k_{CG3,T1} = E_1$
Composite Fitness:	$f_{CG4} = f_{CG4,T1} \cdot f_{CG4,T2} \cdot f_{CG4,T3}$
Partial Fitness:	$f^k_{CG4,T1} = {E_1}^2, \;\; f^k_{CG4,T2} = E_2,$ $f^k_{CG4,T3} = \sqrt{E}_3$

The coevolutionary process described above employed populations of 200 individuals for all PS species, 300 individuals for $CG1$, $CG2$, $CG3$, and 400 individuals for $CG4$. During the evolutionary steps, Replication threshold max_c = 3 is employed. Additionally, an elitist evolutionary strategy was followed in each evolutionary step with the 7 best individuals of each species, copied unchanged in the respective new generation, supporting the robustness of the evolutionary process. After 170 evolutionary epochs the process converged successfully and the cooperation of agent structures with completely different objectives, e.g. those under $CG1$ and those under $CG3$ (see Table 1), is achieved.

3.3. HCCE Assessment

The problem of cognitive system design is used as a test case, investigating HCCE internal dynamics. Because of the embodiment of the cognitive system in the robotic platform and the observation of robot performance for a large number of simulation steps, the coevolutionary process demands on average ten hours to run for 170 evolutionary epochs, and it is impractical to test a large number of different runs. Therefore, we have performed six different runs of the hierarchical coevolutionary scheme. The obtained results are

[1] The biological relevance of the model has been discussed in [11].

illustrated in Fig 6, where each column corresponds to a different run.

In the first run, the progress of the HCCE is initially slow, but after about 100 evolutionary epochs, the probabilistic search identifies a promising evolutionary direction which is efficiently exploited to identify a successful set of solutions. In the following two runs, we see that the coevolutionary process is rather unstable. Specifically, the evolution of species $CG4$ is not able to formulate successful assemblies of cooperators that will be preserved in the consecutive epochs. This fact additionally affects the progress of evolution in species $CG2$, $CG3$, which are trapped in suboptimal solutions. In the fourth run, the progress of the composite coevolutionary scheme develops slowly, and simultaneously for all species. The coevolutionary procedure is terminated without reaching the success rate of the first run. Nevertheless, we easily observe that the progress of evolution is not stabilized, which implies that if the coevolutionary procedure could continue for more epochs, it should be able to estimate a sufficiently good result. The progress of the fifth run is similar to the first. The progress of the HCCE procedure is initially slow, when a promising assembly of cooperators is identified. After a small unstable period in the advancement of the coevolutionary procedure, an effective assembly is preserved, driving also the other individuals in an area of successful solutions. Finally, the progress of the last run is similar to the fourth. The evolution of each CG develops without rapid changes. However, in the current case, the advancement is a bit faster than the fourth run, and thus the composite procedure is able to converge in a set of solutions with a nearly optimum fitness value.

3.4. Unimodal Evolutionary Design

In order to get a better appreciation of HCCE effectiveness, we have also approached the problem at hand by using an ordinary evolutionary algorithm. Specifically, the structure of all cortical and link agents is encoded in a single chromosome. In the current set of experiments a population of 400 individuals is evolved for 170 epochs. The probability of applying crossover and mutation operators over the structure of a cortical or a link agent is the same with the respective probabilities of the coevolutionary scheme.

The fitness function employed to guide the evolutionary process is defined according to the last line of Table 1, similar to the function f_{CG4} that evolves the top-level CG, facilitating also direct comparison of Unimodal and HCCE processes. Specifically, the evolutionary process is driven by:

$$f = \sqrt{E_1} \cdot E_2^2 \cdot E_3 \tag{3}$$

The results of 6 independent runs of the unimodal evolutionary process are illustrated in Fig 7 (a) (compare with the last line of Fig 6). Evidently, none of the ordinary evolutionary processes was successful. Additionally, even the best of them, was not as good as the worst case of the coevolutionary scheme. These results highlight the unsuitability of unimodal evolution to design distributed structures with distinct roles of partial components, and the need for a specialized scheme able to consider explicitly the individual characteristics of substructures. All these issues are sufficiently addressed by the HCCE scheme.

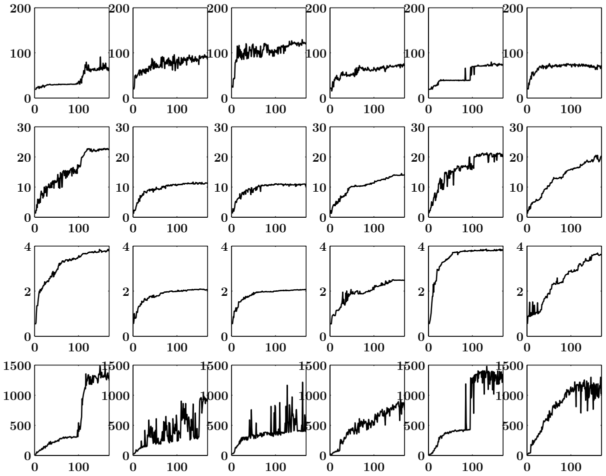

Figure 6. Graphical illustration of the progress of six different HCCE coevolutionary runs. Each column is related to the results observed on the respective run. The lines 1-4 demonstrate the progress observed on the evolution of $CG1$, $CG2$, $CG3$, $CG4$, respectively. All plots demonstrate the fitness value of the best candidate solution in a population, against evolutionary epochs.

3.5. Enforced SubPopulation (ESP) Design

Additionally, we have investigated the possibility of solving the same problem by utilizing the Enforced SubPopulation (ESP) coevolutionary scheme. In the current work, we have implemented the ESP algorithm described in [7], without however activating the stagnation check that practically re-initializes populations when the process gets stalled. Specifically, twelve different species are employed to specify the structure of the twelve components of the model. All partial populations of the ESP scheme are evolved according to the same fitness function describing the objectives of the overall system. Similar to the Unimodal evolutionary scheme, the fitness of ESP solutions is

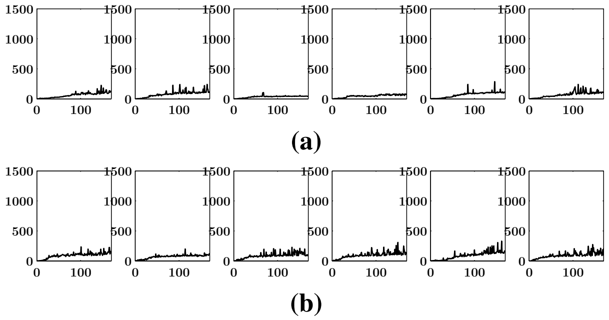

Figure 7. Graphical illustration of the progress of Unimodal and ESP schemes. Each plot demonstrates the fitness value of the best candidate solution in a generation, against evolutionary epochs. (a) The results of six different runs of the ordinary evolutionary process. (b) The results of six different runs of the ESP scheme.

estimated by:

$$f = \sqrt{E_1} \cdot E_2^2 \cdot E_3 \tag{4}$$

Thus the progress of ESP is directly comparable with the progress of the HCCE scheme.

Similar to HCCE, each population evolving the structure of a component of the model consists of 200 individuals, while 2000 assemblies of components are randomly created, trying to identify a successful solution to the overall problem. Thus, each individual representing a candidate structure of a system component, participates in about ten complete solution assemblies. All solutions are evaluated according to eq. (4). The average fitness of individuals drives the evolution of each species.

We have performed 6 independent runs of the ESP scheme which are evolved for 170 epochs. The probability of applying crossover and mutation operators over the structure of a cortical or a link agent is the same with the respective probabilities of the HCCE scheme. The results of these processes are illustrated in Fig 7 (b). These results are directly comparable with the last line of Fig 6. Evidently, none of the ESP processes was successful. This is mainly due to the fact that ESP process is not directed towards explicitly creating successful complex assemblies. We note that in contrast to HCCE, the population of 2000 complete solution assemblies of ESP is not evolved but it is randomly generated in each evolutionary epoch [7]. In other words, it is expected that due to the large number of complex assemblies being evaluated, satisfactory distributed configurations will be randomly formulated. Unfortunately, as it is indicated by the present results, this is not the case for large problems where many components need to be coevolved.

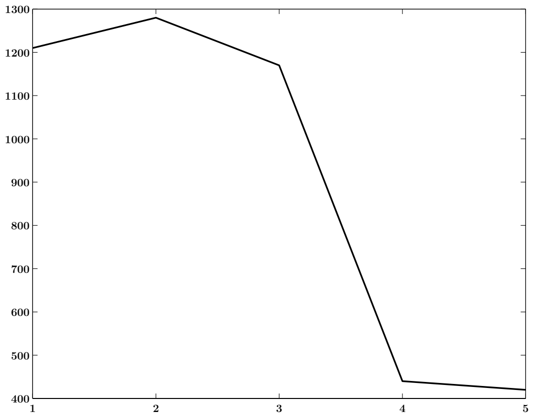

Figure 8. The average fitness of six runs of the HCCE procedure, utilizing different values of replication threshold max_c.

3.6. Replication Operator

Finally, we performed a set of experiments in order to investigate the effect of the "Replication" operator, on the progress of coevolution. The effect of this operator is maximal when max_c threshold values are low, and reduces gradually by increasing max_c. Intuitively, high values of the replication threshold indicate that the assemblies of individuals of partial species remain largely un-affected, in order to be employed as test cases for the individuals of the rest species. In contrast, low values of replication threshold prevent individuals of partial species to participate in many cooperator assemblies, enforcing their independent evolution towards many different directions, according to the percentage of mutation.

We investigate the performance of the HCCE scheme for five values of max_c threshold ($\{1, .., 5\}$), performing six independent runs. The average of maximum fitness values of $CG4$ over the six runs is illustrated in Fig 8. According to the results, the successful convergence of the coevolutionary process for the problem addressed in the current study, is facilitated more by the efficient exploration of the search space. This can be explained by the increased complexity of the problem and the high non-linear interaction among the partial elements of the solution.

4. Discussion

The current work aims at the assessment of the Hierarchical Cooperative CoEvolutionary scheme that is capable of (i) coevolving a large number of complex partial components, (ii) enforcing seamless integration of components in a unified structure, (iii) considering the special role of each component in the unified system, (iv) investigating the performance of the overall system in conditions of partial failure. The results obtained attest to its validity and effectiveness.

Specifically, it is shown that HCCE is capable to design successfully a system consisting of totaly twelve complex partial components. Similar to all evolutionary processes, HCCE does not guarantee the identification of the optimum solution. This is due to the probabilistic nature of the procedure. However, it is shown that HCCE significantly outperforms ordinary Unimodal evolution and the Enforced Sub-Population Coevolution. We believe this is because from the one side HCCE is capable of explicitly addressing the autonomous characteristics of each system component, and from the other side it is equipped with a systematic search mechanism identifying the component structures that can more effectively cooperate.

Additionally, the obtained results demonstrated that the Replication operator facilitates significantly the successful convergence of the coevolutionary process. This is because it provides a means to the assembly encoding populations to modulate the evolution of system components. In other words, Replication conveys information from the higher to the lower levels of the hierarchy, coordinating partial evolutionary processes. Thus, this particular operator can be utilized in other coevolutionary schemes evolving assemblies of individuals [6, 15], in order to support effective integration of partial components.

5. Future Work

The current work studies the effectiveness of Hierarchical Cooperative CoEvolution to solve distributed optimization problems. It has been demonstrated that the proposed coevolutionary scheme is particularly appropriate to design complex systems consisting of many different components. Thus, it can be potentially employed as a general-purpose mechanism that facilitates the design of complex distributed systems.

Additionally, further work is currently underway, aiming at the advancement of HCCE. Research directions that seem to invite productive work concern the enrichment of the hierarchical scheme with Pareto optimality characteristics, and the design of new genetic operators specialized to support the coevolutionary process.

Acknowledgments

The work presented in this paper has been partly supported by the European Commission funded project MATHESIS, under contract IST-027574.

References

[1] J. Casillas, O. Cordón, F. Herrera, and J. Merelo. Cooperative coevolution for learning fuzzy rule-based systems. In *Proc. AE*, pages 311–322, 2001.

[2] E. De Jong. Representation development from pareto-coevolution. In *Proc. Genetic and Evolutionary Computation Confernce, (GECCO)*, 2003.

[3] A. Defaweux, T. Lenaerts, and J. vanHermet. Transition models as an incremental approach for problem solving in evolutionary algorithms. In *Proc. GECCO*, 2005.

[4] M. Delgado, V. F. Zuben, and F. Gomide. Coevolutionary genetic fuzzy systems: a hierarchical collaborative approach. *Fuzzy Sets and Systems*, 141(1):89–106, 2004.

[5] S. Ficici and J. Pollack. Pareto optimality in coevolutionary learning. In *Proc. of 6th European Conference on Artificial Life (ECAL)*, pages 316–325, 2001.

[6] N. Garcia-Pedrajas, D. Ortiz-Boyer, and C. Hervas-Martinez. Cooperative coevolution of generalized multilayer perceptrons. *Neurocomputing*, 56:257–283, 2004.

[7] F. Gomez. Robust non-linear control through neuroevolution. *PhD Thesis, AI-TR-03-303, Department of Computer Sciences, University of Texas at Austin.*, 2003.

[8] F. Gomez and R. Miikkulainen. Solving non-markovian control tasks with neuro-evolution. In *Proc. IJCAI*, pages 1356–1361, 1999.

[9] A. Iorio and X. Li. A cooperative coevolutionary multiobjective algorithm using non-dominated sorting. In *Proc. Genetic and Evolutionary Computation Confernce, (GECCO)*, pages 537–548, 2004.

[10] M. Maniadakis, E. Hourdakis, and P. Trahanias. Modeling overlapping execution/observation brain pathways. In *Proc. Int. Joint Conference on Neural Networks, (IJCNN-07), accepted for presentation*, 2007.

[11] M. Maniadakis and P. Trahanias. Distributed brain modelling by means of hierarchical collaborative coevolution. In *Proc. IEEE CEC*, pages 2699–2706, 2005.

[12] M. Maniadakis and P. Trahanias. Modelling brain emergent behaviors through coevolution of neural agents. *Neural Networks Journal*, 19(5):705–720, 2006.

[13] E. Popovici and K. De Jong. Understanding cooperative coevolutionary dynamics via simple fitness landscapes. In *Proc. GECCO*, 2005.

[14] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Computation*, 8:1–29, 2000.

[15] J. Reisinger, K. Stanley, and R. Miikkulainen. Evolving reusable neural modules. In *Proc. GECCO*, 2004.

[16] R. Watson and J. Pollack. A computational model of symbiotic composition in evolutionary transitions. *Biosystems*, 69(2-3):187–209, 2002.

[17] R. Wiegand, C. Liles, and A. De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Proc. GECCO*, pages 1235–1242, 2001.