In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002),* IEEE, 2002.

A Blended Population Approach to Cooperative Coevolution for Decomposition of Complex Problems

Donald Sofge[1], Kenneth De Jong[2], Alan Schultz[1]
[1]Navy Center for Applied Research in Artificial Intelligence,
Naval Research Laboratory
{Sofge, Schultz}@aic.nrl.navy.mil
[2]Department of Computer Science,
George Mason University
kdejong@cs.gmu.edu

Abstract - Cooperative coevolutionary architectures provide a framework for solving complex problems by decomposing them into constituent subproblems, solving the subproblems, and then reintegrating the solutions. This paper presents a blended cooperative coevolution model which offers advantages over traditional evolutionary algorithms and currently used cooperative coevolutionary architectures.

1 INTRODUCTION

In order to extend the usefulness of evolutionary algorithms to address challenging real world problems, we need to endow these methods with the ability to automatically decompose complex problems. While there are many definitions for complexity, a common aspect of problems that are generally agreed to be “complex” is that brute force solutions are not practical or desirable because the search space is too large for any reasonably powerful computer to examine every possible solution in a lifetime.

Recent cooperative coevolutionary architectures (CCAs) have been proposed as (at least partial) solutions to the decomposition of complex problems for evolutionary algorithms [6],[2],[7]. These CCAs combine the advantages of evolutionary algorithms with the advantages of iterated line search procedures by decomposing the genome representation of the problem into a number of subspecies, which are then coevolved in parallel. More advanced versions of these architectures have been demonstrated which dynamically create new subspecies based upon measured stagnation in the evolutionary process, and destroy subspecies which are no longer making significant contributions. In this way the CCA is able to dynamically evolve the appropriate number of interacting subspecies to solve a problem [8].

The Potter CCA method decomposes the genome into a number of subspecies (initially the same as the number of variables) with the best (using the “greedy” collaboration method) individuals from each subpopulation collaborating together for fitness evaluation. For each subpopulation being evolved, all of the other subpopulations are held fixed. The subpopulations are each evolved in a round-robin fashion, performing mutation, crossover, and/or whatever other genetic operators are designed for the evolutionary process. Credit is assigned to each subspecies by pairing each individual from the subspecies with an individual from each of the remaining subspecies and assigning fitness.

While this architecture offers significant advantages over traditional EAs for many problems, a difficulty encountered is that if there is a high degree of interaction between the subspecies (epistasis), use of the simple collaboration mechanism (best individual from each subspecies) can lead to poor performance [6]. This effect can be somewhat mitigated by using a different collaboration mechanism [6], [7], [10] in which random individuals from each subpopulation are selected for collaboration (the “less greedy” collaboration method).

The alternative approach presented in this paper is to change the nature of the search procedure by combining the advantages of both CCAs with traditional EAs by coevolving separate populations that are blended together into a single common population during the course of the evolutionary cycle. This approach makes use of both the cooperative coevolutionary architecture as well as *island model* approaches to coevolution [3], [1] where individuals are allowed to periodically migrate from one subpopulation to another. The island model methods rely upon the competition of subpopulations, with migration used to maintain subpopulation diversity. However, a key difference between the blended population approach and coevolution using the *island model* methods is that in the blended population approach migration is not used as a tool to maintain subpopulation diversity. Instead, migration encourages greater mixing of subspecies components later in the evolutionary cycle to better facilitate epistatic interactions between the subcomponents (once they are merged in the Common population). This allows the evolutionary process to avoid getting stuck in local optima, an affliction caused by greediness of the CCA approach.

2 COMPLEXITY AND DECOMPOSITION

The notion of solving large complex problems by decomposing them into simpler subproblems that may be solved and then recombined to solve the larger problems probably dates to the origins of human problem solving. Engineers are commonly taught that to solve a difficult problem they need to first break it down into smaller, more manageable subproblems that may be individually solved.

Recent work by Kauffman introduced the NK landscape model as a means to build fitness landscapes with tailored properties such as the number of genes or variables and the ruggedness of the landscape [5]. While the NK model allows one to build up landscapes of varying complexity, little is said of how to decompose complex landscapes of already existing problems. In fact, Kauffman's work on Complexity Theory suggests that evolved systems are naturally chaotic and may not be deconstructed, and therefore may not be decomposable [4]. A counter-argument is that while the complex system may not be mathematically decomposable, it may still be possible to evolve decompositions of the complex problem such that it is effectively decomposed into constituent subcomponents.

Perhaps the strongest argument in favor of the decomposability of complex problems is that humans solve complex problems by decomposing them, solving the subproblems and then combining the subproblem solutions together to solve the larger complex problems. If we want to have machines that exhibit human-like intelligence, then we need to endow these machines with the same capabilities. The following sections will detail an approach to improve upon the decomposition capabilities of cooperative coevolutionary algorithms through use of a blended population model.

3 BLENDED POPULATION MODEL FOR COOPERATIVE COEVOLUTION

The goal of this paper is to present a blended population approach to cooperative coevolution by combining both CCA and standard EA methods together in a single evolutionary process, and introducing a migration operator to allow populations to move from CCA subpopulations to a standard EA population. One population is made up of subspecies and is implemented as a Potter CCA, and the other population (Common) is a traditional EA. The two populations may then be blended together over the course of the evolutionary process by allowing individuals to migrate from the CCA subpopulations to the Common population (Figure 1). Migration between the CCA and the EA is stochastic with the rate of migration determined by migration parameter M, where $0 \leq M \leq 1$. For $M=0$ no migration occurs.

For $M=1$ one member of each subspecies migrates to the Common population (and leaves its respective subpopulation) each generation. For values of M between 0 and 1, a random number between 0 and 1 is generated each generation. If the random number is less than M, then migration occurs that generation.

The total combined population size is held constant, and the sizes of the subpopulations (not including Common) are constrained to be the same. Thus, for each migration to take place, each subspecies population is depleted by 1 individual, and the Common population gains a number of new individuals equal to the number of subpopulations present in the CCA model. Migration is only permitted for the best individual from each subspecies (exception noted below). Therefore, the best individuals migrate to the Common population where they are mutated and crossed over with other members of the Common population. As an individual migrates from the subspecies population to the Common population, it may be either deleted from the subspecies population (lower selection pressure) or simply copied from the subspecies population (higher selection pressure). If it is simply copied, then the current least fit member of the subspecies population will be killed off in order to maintain a constant total population size.

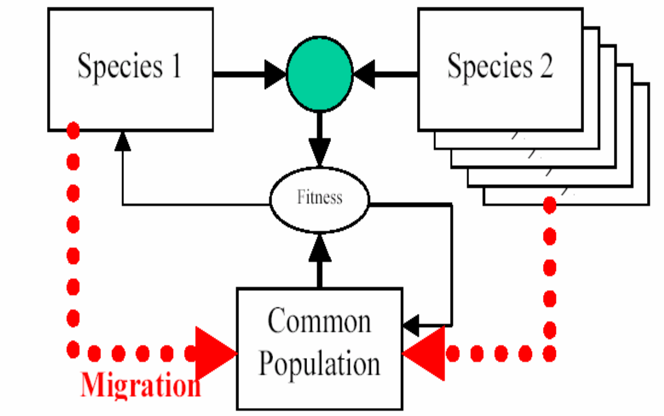

Figure 1: Blended Population Model for Cooperative Coevolution

4 EXAMPLE: FUNCTION OPTIMIZATION

In order to test the proposed technique and provide a basis for comparison with previously published results in cooperative coevolution, a series of function optimization tasks were selected. The functions in the test suite were chosen to be consistent with the 1994 study [6] including the Rosenbrock, Rastrigin, Schwefel, Griewangk, and Ackley functions, as well as an Off-Axis Quadratic studied in [10] and [9]. While previous studies by Potter have included *n*-dimensional versions of some of these functions, this study, like the [10] study, restricted the test suite functions to functions of two independent variables. That is, fitness = $f(x_1,x_2)$. A common characteristic of each of these functions is that the global minimum is zero. Therefore the optimization task may be cast as minimizing fitness. The results shown in this paper are consistent with this approach.

In contrast to the previous studies on these functions, the evolutionary algorithm selected here was an evolutionary strategy utilizing real-valued vectors. The ES was selected since real-valued vectors are the most direct and natural representation for the functions being optimized. The ES utilized both mutation and crossover operators, as well as a migration operator to stochastically allow individuals to migrate from the subspecies populations to the Common population. Further details of the implementation are discussed below in *Section 4.2.*

4.1 Test Suite

The first function in the test suite was the Rosenbrock function, defined as

$$f(x_1,x_2) = 100*(x_1^2-x_2)^2 + (1-x_1)^2 \qquad -2.048 \leq x_i \leq 2.048 \quad (1)$$

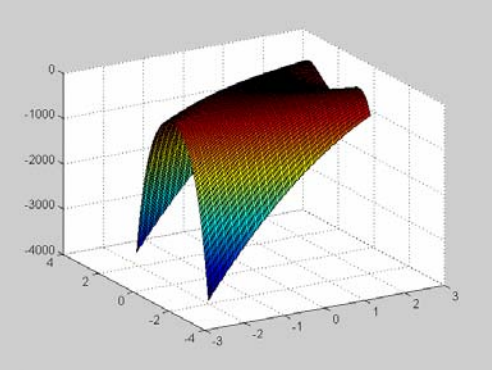

Figure 2: Inverted Rosenbrock Function

This function was included in the 1994 Potter study as well as several cooperative coevolution studies since ([2],[7],[8],[10]). This function was part of the original De Jong test suite (F2), and is known to have strong variable interactions. An inverted form of this function (suitable for *maximization* instead of *minimization*) is shown in Figure 2. The second function in the test suite was the Rastrigin function, defined

$$f(x_1,x_2) = 6+(x_1^2-3*\cos(2\pi x_1)) + (x_2^2-3*\cos 2\pi x_2)) \quad (2)$$
$$\text{where } -5.12 \le x_i \le 5.12$$

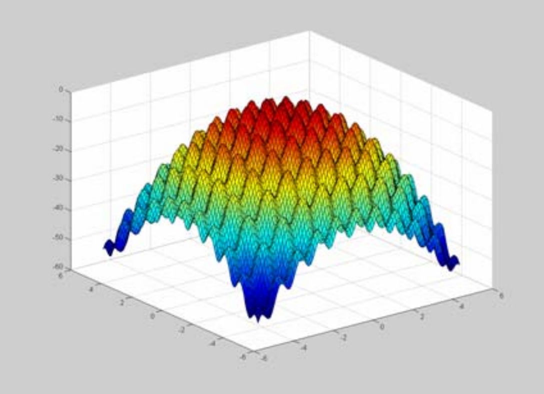

Figure 3: Inverted Rastrigin Function

This function is known to be linearly separable but exhibits many local optima. This function was also used in the 1994 study [6] as well as several studies since ([2],[7]-[10]).

The third function in the test suite was the Schwefel function, which was also part of the 1994 Potter study. This function was designed to trap optimization procedures on a local peak where the second best optima is far away from the global optimum. The two variable form of the function used in this study is defined as

$$f(x_1,x_2) = 837.9658+x_1\sin(\text{sqrt}(|x_1|)) + x_2\sin(\text{sqrt}(|x_2|)) \quad (3)$$
$$\text{where } -500 \le x_i \le 500$$

The global optimum can be seen in the bottom center of the figure, whereas the second best peak is near the top center.

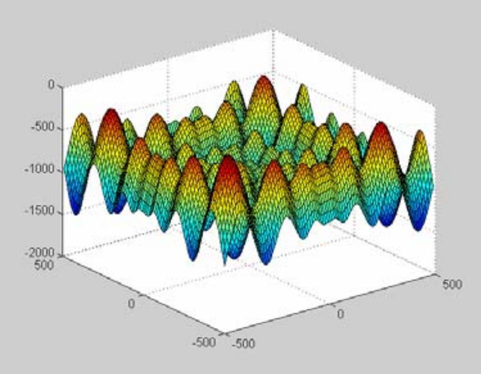

Figure 4: Inverted Schwefel Function

The fourth function in the test suite was the Griewangk function in two variables, defined

$$f(x_1,x_2) = 1 + (x_1^{2+} x_2^2)/4000 - \cos(x_1)*\cos(x_2/\text{sqrt}(2)) \quad (4)$$
$$\text{where } -600 \le x_i \le 600$$

This function, part of the 1994 Potter study, was designed to exhibit interactions between the product terms. This effect may be substantially lessened in the simplified form of the function (two-dimensional rather than thirty-dimensional) used in this study and shown in Figure 5 below. While slight interaction effects are noticeable upon close inspection, the function mostly appears unimodal, nearly taking the form of a parabolic surface.

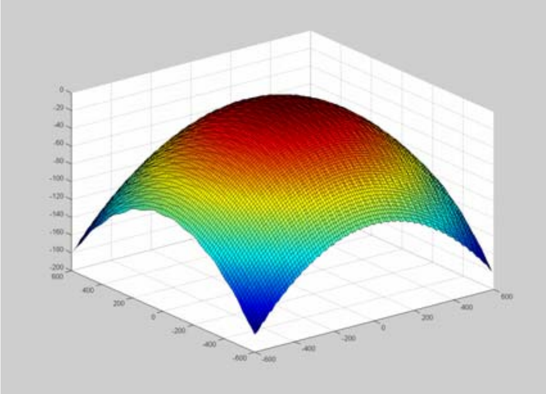

Figure 5: Inverted Griewangk Function

The fifth function in the test suite was the Ackley function, also used in the 1994 Potter study [6]. It has the characteristic that while it appears unimodal at low resolutions, at higher resolutions it appears more complex. The Ackley function was defined as

$$f(x_1,x_2) = 20+e-20*\exp(-0.2*\text{sqrt}((x_1^{2+} x_2^2)/2))$$
$$- \exp((\cos(2\pi x_1) + \cos(2\pi x_2))/2) \quad -30 \le x_i \le 30 \quad (5)$$

The final function included in the test suite was an off-axis quadratic shown to introduce problems for some evolutionary

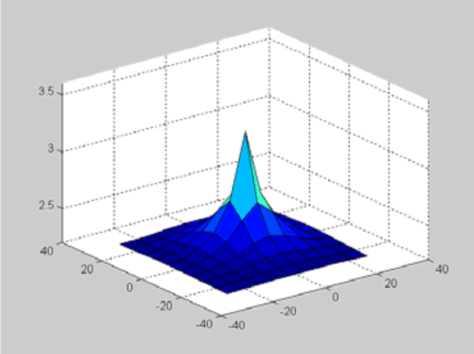

Figure 7: Inverted Ackley Function

algorithms [9] due to the lack of axis alignment with the principal axes of the subspecies (assuming decomposition of the landscape into x-axis and y-axis domains). This function was also used in the 2001 study of collaboration methods for cooperative coevolution [10]. It is defined as

$$f(x_1,x_2) = x_1^2 + (x_1 + x_2)^2 \qquad -65.536 \leq x_i \leq 65.536 \quad (6)$$

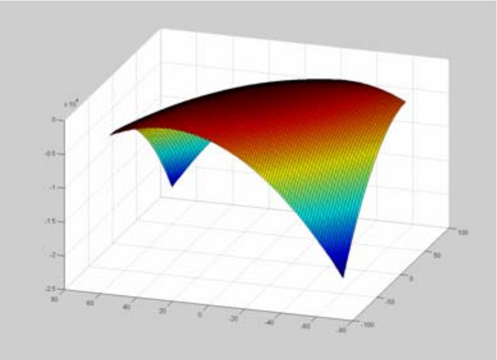

Figure 8: Inverted Off-Axis Quadratic Function

4.2 Implementation Details

Evolutionary Strategy (generational)
Representation: real-valued vectors
No fitness scaling
Selection: fitness proportionate
Elitist Strategy: single most fit parent survives
Operators: mutation (Gaussian random)
cross-over (avg. of parents)
Migration: (move from subpops to common pop)
Mutation Probability: 0.9
Mutation Step-size (var): 0.1
Crossover Probability: 0.2
Population Size (total): 100
Maximum Generations: 100 (10,000 Function Evals.)

These parameters were used uniformly for the cooperative coevolution evolutionary strategy (CCES) model (for evolving each of the subspecies) as well as the ES used on the Common population. The same parameters were used for the ES used in isolation (no CCES) for comparison purposes. No attempt was made to tune the parameters to any particular optimization task or for any of the algorithms.

Each function was optimized using a range of blended population models, both with and without migration, as well as with CCES and standard ES models. In the cases where migration was turned off, a fixed Common population size (denoted by C on the graphs) was used, and the sizes of the subspecies populations were also held constant. This approach is equivalent to having the CCES model and ES model compete for best-so-far, but have no actual exchange of genetic material take place. This approach, while more simplistic than the migration model, has advantages over both standard ES and CCES approaches in that it combines the best attributes of each in finding the global optima. By setting the Common population size C, one can smoothly vary the search from a mostly CCES approach (small C) to a mostly ES approach (large C).

The second method used with the blended population models made use of a nonzero migration rate setting. The Common population was seeded with 10 individuals (out of a total population of 100), with the remaining 90 individuals divided evenly between the two subspecies populations (one for each variable). Thus, there are 3 subpopulations altogether. The migration parameter *M*, where *0≤M≤1*, governs the rate of mutation.

Migration between the CCA and the EA is stochastic. A random number between 0 and 1 is generated each generation. If the random number is less than *M*, then migration occurs that generation.

Each time migration is activated one member of each subspecies migrates to the Common population. The first individual selected for migration is the fittest individual from the first species and its most recent mate from the second species. The second individual would normally be the fittest individual from the second species and its most recent mate from the first species. However, what often happens in the CCES process is that the fittest individual of the first species mates with the fittest from the second species. Migrating both of them with their mates would be the equivalent of adding two identical individuals to the Common population. When the fittest individuals point to one another, the second individual selected for migration is a randomly selected pairing from each of the two subpopulations.

Two alternatives exist with respect to deletion of the migrating individuals from the subspecies populations. The greedy approach (higher selection pressure) is to simply copy the best individual from each subpopulation to the Common population, and then rebalance the total population by killing off the least fit member of each of the subspecies populations. The less greedy approach is to kill off the migrating individuals from the subspecies populations. The greedy method was used for the results shown below.

4.3 Simulation Results

Each of the simulation runs was repeated 50 times with the random number generator reseeded each time based upon the system clock. The results shown are the means of those 50 runs. For each mean 95% confidence intervals were calculated. On the results graphs certain means are shown with error bars. These error bars are the 95% confidence intervals. Where no error bars are shown, the lines still indicate means over 50 trials, but for clarity of presentation their confidence intervals were not included on the graphs.

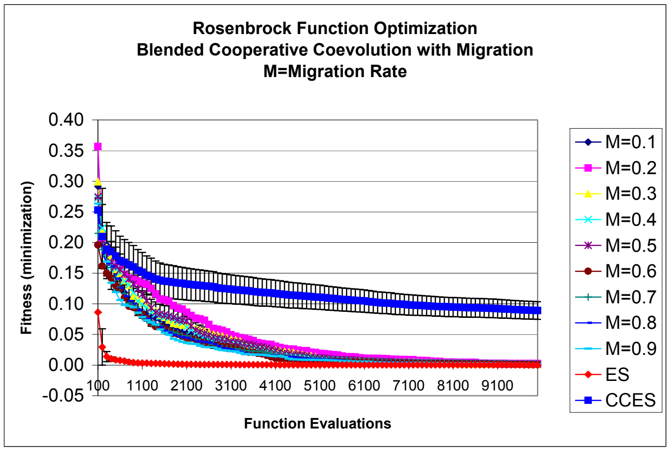

Figure 9: Rosenbrock Optimization Results

4.3.1 Rosenbrock Function Optimization Results

As shown in Figure 9, the Blended Cooperative Coevolution Evolutionary Strategy (BCCES), 0.1≤M≤0.9, outperformed the Cooperative Coevolution Evolutionary Strategy (CCES). In fact the CCES usually failed to converge while the other techniques all converged within the 10,000 function evaluations (100 generations). This is consistent with the result found by Potter [6] using a CCGA with the simple (greedy) collaboration method. The greedy collaboration method was also used for these trials.

It is interesting to note that the best technique on this task was the standard Evolutionary Strategy (ES). This is probably due to the fact that the ES can easily find the optimum of such a small surface, with a population size of 100, after only a few generations.

4.3.2 Schwefel Function Optimization Results

From Figures 10 we can see that CCES and ES both out-performed the BCCES strategies with no migration. However, once migration was turned on (Figure 11), the BCCES strategies did as well or better than the CCES strategy. Once again, they were all outperformed by the simple ES on this task. It appears at first glance that none of the techniques actually reached the zero minimum over the 10,000 allocated function evaluations. In fact most of them did, but not consistently so. Often they would get stuck on a local minima (fitness range was 0-1700). The results show means across 50 trials.

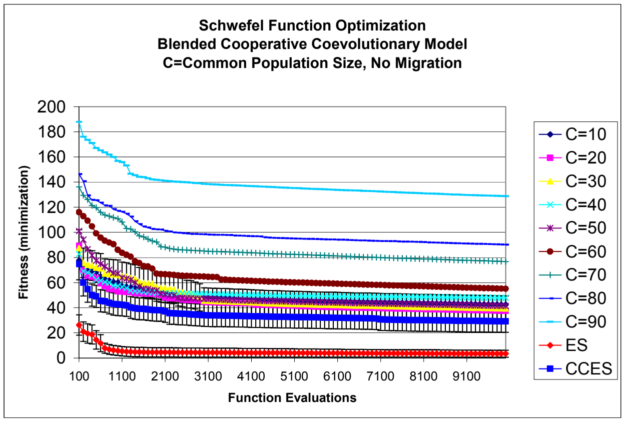

Figure 10: Schwefel Optimization Results, No Migration

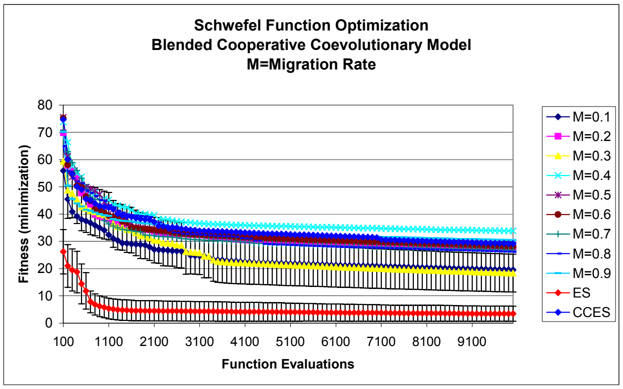

Figure 11: Schwefel Optimization Results with Migration

4.3.3 Rastrigin, Griewangk, OAQ, and AckleyResults

On each of these remaining surfaces all of the techniques were able to find the optimum quickly such that there was no statistically significant difference between the various techniques, as shown in Figure 12 for the Rastrigin function.

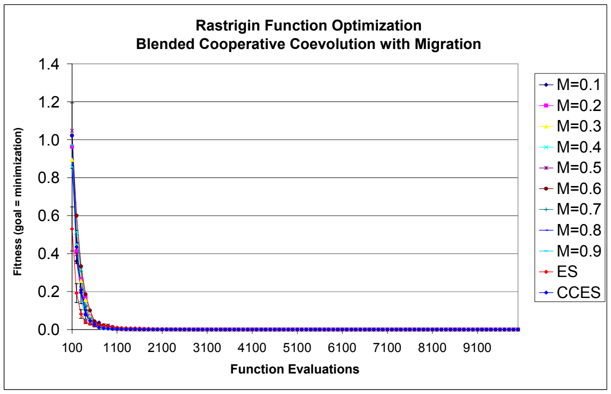

Figure 12: Rastrigin Optimization Results with Migration

No conclusions can be drawn from the results on these functions with regard to the relative strengths and weakness of each approach.

5 RESULTS AND CONCLUSIONS

5.1 Summary of Experimental Findings

Of the six test functions included in the test suite, only two (Rosenbrock and Schwefel) produced "interesting" results which might shed light on the advantages and/or disadvantages of the BCCES model when compared with a standard ES and with a CCES model. The remaining four test functions (Rastrigin, Griewangk, Ackley, and Off-Axis Quadratic) were trivially solved in less than 30 generations each time by each technique, with the exception that in certain cases the BCCES model with migration turned on performed better than with it turned off. It should be noted that only the simplest (two variable) versions of these functions were used.

With regard to performance on the Rosenbrock and Schwefel functions, on the Rosenbrock function the BCCES model significantly outperformed the CCES model both with migration turned off and with it turned on. On the Schwefel function the BCCES model performed the same as the CCES model (within the bounds of statistical significance) with migration turned off. With migration turned on BCCES performed as well or better than CCES (depending upon the migration rate parameter setting).

5.2 Conclusions

In this paper a blended population cooperative coevolution model was presented which combines the advantages of the Potter cooperative coevolutionary architecture with those of a standard evolutionary architecture. In addition, the use of a migration operator (similar to that used for *island model* type competitive coevolutionary architectures) allows the population mass to shift from the CCA subspecies populations to a Common population during the course of evolution. The effect of this shift is that as evolution progresses, the model is able to better handle epistatic interactions between the subcomponents.

Preliminary experimental results on a somewhat simplified test suite that includes functions used for evaluating previous cooperative coevolutionary architectures indicates that the blended population model offers clear advantages over the standard CCA approach for at least some problems which exhibit a high degree of coupling or epistatic interaction between components.

The motivation behind this effort is to enhance the capabilities of coevolutionary architectures to better evolve solutions to complex problems through emergent decomposition of the problems into constituent subcomponents, evolution of these subcomponents, and recombination of the results. The use of a migration operator for allowing movement of individuals between the subspecies populations and the Common population mirrors this approach to decomposition (breaking the domain into subspecies), evolution of the subcomponents (the subspecies are coevolved in parallel) and recombination (subspecies are rejoined in the Common population).

REFERENCES

[1] Cohoon, J., S. Hegde, W. Martin, and D. Richards (1987). Punctuated equilibria: A parallel genetic algorithm. In J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 148-154. Lawrence Erlbaum Associates.

[2] De Jong, K. and M. Potter (1995). Evolving Complex Structures via Cooperative Coevolution, *Fourth Annual Conference on Evolutionary Programming*, March 1-3, San Diego, CA.

[3] Grosso, P. (1985). *Computer Simulations of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model.* Ph. D. thesis, University of Michigan, Ann Arbor, MI.

[4] Kauffman, S. A. and S. Johnsen (1991). Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Artificial Life II, SFI Studies in the Sciences of Complexity*, Volume 10, pp. 325-369. Addison-Wesley.

[5] Kauffman, S.A. (1993). *The Origins of Order*. Oxford University Press.

[6] Potter, M. and K. De Jong (1994). A cooperative coevolutionary approach to function optimization. *In Proceedings from the Third Parallel Problem Solving from Nature*, pages 249-257. Springer-Verlag.

[7] Potter, M. (1997). *The Design and Analysis of a Computational Model of Cooperative CoEvolution.* PhD thesis, George Mason University, Fairfax, Virginia.

[8] Potter, M. and K. De Jong (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1-29.

[9] Salomon, R. (1996). Performance degradation of genetic algorithms under coordinate rotation. In L. Fogel, P. Angeline, and T. Back, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming V*, pages 153-161. MIT Press.

[10] Wiegand, P., W. Liles, and K. De Jong (2001). An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms. *Proceedings Genetic and Evolutionary Computation Conference*, July 7-11, San Francisco, CA.