A Cooperative Coevolutionary Multiobjective Algorithm Using Non-dominated Sorting

Antony W. Iorio and Xiaodong Li

School of Computer Science and Information Technology,
Royal Melbourne Institute of Technology University,
Melbourne, Vic. 3001, Australia
{iantony, xiaodong}@cs.rmit.edu.au
http://goanna.cs.rmit.edu.au/~xiaodong/ecml/

Abstract. The following paper describes a cooperative coevolutionary algorithm which incorporates a novel collaboration formation mechanism. It encourages rewarding of components participating in successful collaborations from each sub-population. The successfulness of the collaboration is measured by a non-dominated sorting procedure. The algorithm has demonstrated it can perform comparably with the NSGA-II on some multiobjective function optimization problems.

1 Introduction

In nature, coevolution is the process of reciprocal genetic change in one species, or group, in response to another. This process can also be utilised within evolutionary algorithms, and recently there has been a growing interest in the application of coevolution within multiobjective evolutionary algorithms. The reciprocal change observed in coevolution can be considered either as a competitive arms race, such as the coevolution of test cases for a problem (predators) with the solutions (prey) [1], or more recently, cooperative approaches where separate sub-populations evolve components of the solution [2, 3].

The cooperative coevolutionary algorithm (CCA) [3], which was utilised in this work, separates the components of a problem solution into sub-populations, where each sub-population is subject to an evolutionary process. Solution components are rewarded in terms of their participation within good candidate solutions. In principle, the process of rewarding components of candidate solutions should also improve convergence towards the global Pareto front on particular multiobjective problems.

The CCA has already demonstrated benefits in a number of single objective optimization problem domains such as inventory control [4] and learning sequential decision rules [5]. Typically in this kind of algorithm, one picks the best collaborators from each sub-population to form a candidate solution, however this greedy approach can sometimes lead to premature convergence. We have proposed a new collaborator selection mechanism for the multiobjective domain; components of a solution can have the same rank if they belong to the same non-domination level. For example, a solution component of non-domination level 1

has participated in complete solutions which are non-dominated. Similarly, solution components of non-domination level 2 have participated in collaborations which are dominated by rank 1 collaborations, but are non-dominated with respect to all other levels. Because there can be more than one solution component in a non-domination level, the selection process is no longer a single 'best' component but a randomly selected component from the best non-domination levels in each sub-population. This collaborator selection process is quite different from other MOEAs implementing the CCA [6, 7]. We will discuss the implications of this in more detail in section 4.

The second aspect of this work is the rewarding of collaborators from their contribution to non-dominated solutions. Some of the more robust and successful multiobjective evolutionary algorithms (MOEAs) use a dominance ranking mechanism [8], however there is a potential problem with this approach; there is no mechanism which awards a ranking based on the contribution of the solution components. Without such a mechanism, potentially good components are lost because they may participate with poor components in a candidate solution. This observation provides our motivation to build upon existing approaches to multiobjective evolutionary optimization, and to provide a rewarding mechanism for the components of solutions.

We will begin with an introduction to multiobjective optimization in section 2. Section 3 provides a review on coevolution and some work which has been done applying coevolution to multiobjective problems, section 4 will describe the proposed algorithm, followed by a description of the performance metrics, parameter settings, and experiments in section 5 and 6. In section 7 and 8 we discuss the results of these preliminary experiments and some future directions for this work.

2 Multiobjective Optimization

We are interested in solving problems which are formulated with some or possibly all of the objectives in conflict with each other. The problem can be described as a vector of objectives $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), .., f_n(\mathbf{x}))$ subject to a vector of parameters $\mathbf{x} = (x_1, x_2, ..., x_m) \in \mathbf{X}$, where $\mathbf{x}$ is an input parameter vector from the parameter vector space $\mathbf{X}$, n is the number of objectives, and m is the number of parameters. A solution $\mathbf{x} = (x_1, x_2, ..., x_n)$ dominates a solution $\mathbf{y} = (y_1, y_2, ..., y_n)$ if objective function $f_i(\mathbf{x})$ is no worse than objective function $f_i(\mathbf{y})$ for all n objectives and there exists some objective j where $f_j(\mathbf{x})$ is better than $f_j(\mathbf{y})$. The non-dominated solutions in a population are those solutions which are not dominated by any other individual in the population. Multiobjective evolutionary optimization is typically concerned with finding a diverse range of solutions close to the Pareto-optimal front, which is the globally non-dominated region of the objective space.

A number of evolutionary multiobjective algorithms have been developed since the late 80s, and NSGA-II [9], amongst others, is typically regarded as the current state of the art.

3 Previous Work

A number of approaches have been proposed which incorporate both competitive or cooperative coevolutionary methods to improve the performance of multiobjective EAs. We will present a brief overview of these approaches in this section.

3.1 Cooperative Coevolution

MOCCGA [7] integrates cooperative coevolution [3] with MOGA [10]. It uses a dominance rank for individuals where a count of the number of individuals dominating an individual is the fitness criterion. In the MOCCGA objectives are evaluated twice for each individual; with the best ranked individuals from each sub-population, as well as randomly selected individuals, which is the approach described by Potter and De Jong to minimise premature convergence on some test problems [11]. The better of the two collaborations can then be determined. In the MOCCGA sub-components were apparently ranked only within the same 'species' or sub-population. Ranking components in this way could potentially limit the assessment of an individual. The number of evaluations used was also not reported.

The Distributed Cooperative Coevolutionary Multiobjective Evolutionary Algorithm (DCCEA) [6] used a similar ranking technique however the domination count was against individuals maintained in an archive. A relatively large number of evaluations were used on the test functions in order to demonstrate improvements over NSGA-II and other algorithms.

Coello has also demonstrated a cooperative coevolutionary algorithm for multiobjective optimization which subdivides the decision variable space [2]. This approach determines which portions of the decision variables intervals are being used and discards portions of the intervals that it deems are not being used by the search process. It also subdivides intervals so separate sub-populations can operate on the portions of these intervals which are deemed to contribute to the search. Sub-populations which are not making contributions are eliminated from the search.

Another cooperative genetic algorithm which has been developed, utilises symbiosis parameters which can affect the fitness of individuals before they are ranked [12]. These parameters enable the modification of an individual's fitness based on its interaction between objective functions and between individuals.

3.2 Competitive Coevolution

Barbosa and Barreto have demonstrated a competitive algorithm which coevolves a weight vector which weights the objective functions [13]. Both populations are coupled with a fitness evaluation. Assigning weights to favour particular solutions is difficult, and coevolution helps in this regard.

Another competitive approach is applied to the design of airframes. In this approach two and three objectives are coevolved in separate sub-populations [14].

Lohn *et. al.* have also devised a competitive algorithm with solutions in one population and another population containing a population of target objective vectors (TOVs). These vectors contain targets for the trial population to overcome [15, 16].

4 The Non-dominated Sorting Cooperative Coevolutionary Genetic Algorithm (NSCCGA)

In the following section we will describe the NSCCGA. The algorithm begins with a random initialization of all individuals across all sub-populations. Each sub-population is responsible for a particular parameter, x_i, from the decision space. If there are n decision variables there will be n sub-populations according to a natural decomposition of the problem.

4.1 Method

In the first generation random collaborations are formed and evaluated. This step is similar to the procedure outlined in Figure 1 step 1, except that we select random individuals from the complete set of components in each of the other sub-populations. Once these collaborations are formed, they are evaluated on the objectives, and the results from the evaluation are assigned back to the individual undergoing evaluation. In the first generation there will be only the current sub-populations, and the non-dominated sorting will only be over entirely random collaborations formed by this first generation of sub-populations. This is the only difference between collaboration formation in the first generation and following generations.

After the first generation, the resulting child sub-populations Q_1 to Q_n are evaluated (where Q_i is the child sub-population dealing with variable x_i) by forming collaborations with randomly selected components from the 'best' no-domination levels in the previous generation's sub-populations, P_1 to P_n (Figure 1 step 1). This collaboration formation is explained in more detail in section 4.2.

Following collaborator formation and evaluation, we perform a fast non-dominated sorting procedure [9] in step 2, over all collaborations from Q_1 to Q_n and the parent sub-populations P_1 to P_n (The parents have been assigned evaluations from their participation in collaborations in the previous generation). The sorting occurs on the values resulting from the evaluation on the objective functions. This will assign a front membership (non-domination level) F, to each of the individuals from the child sub-populations and parent sub-populations. F_1 contains the best candidates, F_2 the next best, and so on.

In step 2 the crowding distance [9] is calculated for each of the collaborations as well, just as it is for the NSGA-II. If two solutions are of the same non-domination level, the crowding distance sort determines which is better. Solutions with a higher crowding distance are preferred because they contribute

to a more uniform non-dominated front. Both the non-dominational level and crowding distance are assigned back to the individual undergoing evaluation.

Step 3 applies the elitism operator which removes a number of the worst individuals from each sub-population which is in proportion to the number of children that were added. This allows the sub-population sizes to remain constant, while preserving good components. Good candidate components from the previous generation will have an opportunity to continue participating as well. The resulting sub-populations are the new parent sub-population P_1^{t+1} to P_n^{t+1}. The elitism operator is the same as NSGA-II except it operates on sub-populations.

Using the tournament selection operator, individuals are selected for mating from each sub-population and inserted into their respective mating pools. Crossover and mutation are then performed for each mating pool producing children for each of the sub-populations. The tournament operator rule we use for selection [9] states that a solution i wins a tournament against solution j if any of the following conditions are true:

- If solution i has a better non-domination level.
- If they have the same non-domination level but solution i has a better crowding distance than solution j.

After generating the new child sub-populations, the algorithm iterates until some termination condition is met. For further details regarding the elitism and non-dominated sorting procedures, including mechanisms for maintaining a diverse set of solutions, the reader is referred to the following paper on the NSGA-II [9]. The NSCCGA is implemented as a real coded GA, therefore the simulated binary crossover [17, 18] and mutation operators [19] were used for recombination. These operators were also used in the NSGA-II.

4.2 A Novel Collaboration Formation Mechanism

Previous work with the CCA on single objective problems has primarily selected the current 'best' components from each sub-population to merge into a collaboration, or performed a tournament with candidate solutions formed with the current 'best' and randomly selected components. In a multiobjective scenario there may be a number of individuals in each sub-population which are parts of overall solutions that are non-dominated in relation to each other, so we cannot favour one over the other. In this case, we have proposed a novel collaboration formation mechanism where we select collaborators randomly from the 'best' non-domination level in each sub-population (Step 1 in Figure 1). The non-dominated sorting procedure sorts the collaborations into a number of separate non-dominated fronts. Individuals from the best non-dominated front, F_1, are given a non-domination level of 1, followed by 2 for the second front, F_2, and so on. By selecting randomly from the collaborators with the 'best' non-domination level in each sub-population we increase the chances of finding more diverse solutions. Early on in the search, there may be only a few individuals from the 'best' non-domination level, so the random selection of collaborators is from a

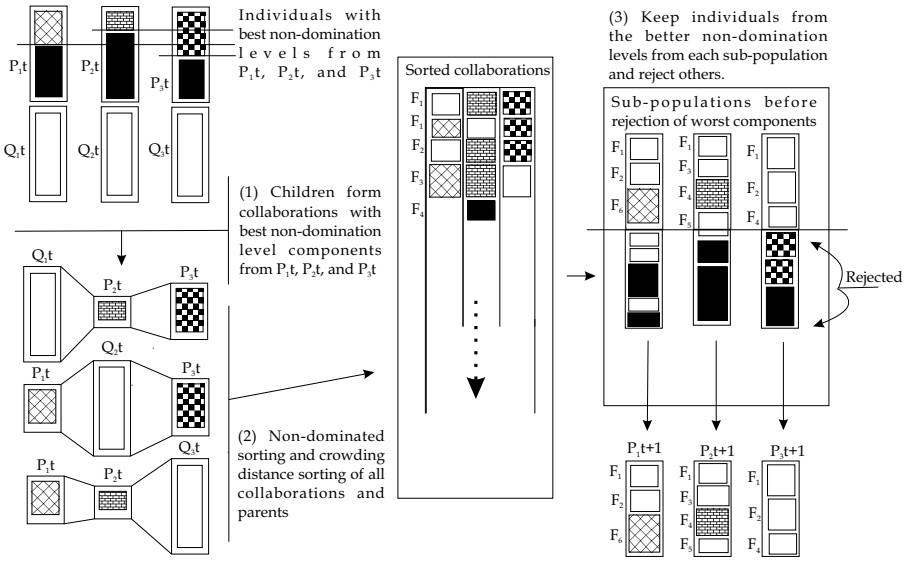

Fig. 1. Outline of the NSCCGA procedure. Q_i^t are the child sub-populations and P_i^t are the parent sub-populations at time t. There are n sub-populations where n is the number of decision variables.

small set in each sub-population. Over successive generations, the number of components from the 'best' non-domination level increases, and as a result, the number of collaborators we can choose from in each sub-population increases significantly. The ranking approaches that were previously used in the multi-objective CCAs (Section 3) are either too exploitive because they only allow the selection of the single 'best' individual for collaboration [6], or they implement a tournament with 'best' and randomly selected collaborators which can potentially waste evaluations [7].

5 Performance Metrics

We use the following performance metrics introduced by Zitzler *et* al. [20]:

$$\mathcal{M}_1^*(Y') := \frac{1}{|Y'|} \sum_{p' \in Y'} \mathbf{min}\{||\mathbf{p}' - \bar{\mathbf{p}}||^*; \bar{\mathbf{p}} \in \bar{\mathbf{Y}}\} \tag{1}$$

$$\mathcal{M}_2^*(Y') := \frac{1}{|Y' - 1|} \sum_{p' \in Y'} |\{q' \in Y'; ||p' - q'||^* > \sigma^*\}| \tag{2}$$

$$\mathcal{M}_3^*(Y') := \sqrt{\sum_{i=1}^{n} \mathbf{max}\{||\mathbf{p}'_\mathbf{i} - \mathbf{q}'_\mathbf{i}||^*; \mathbf{p}', \mathbf{q}' \in \mathbf{Y}'\}} \tag{3}$$

where Y' is the set of objective vectors corresponding to the non-dominated solutions found, and $\bar{Y}$ is a set of uniform Pareto-optimal objective vectors. A niche neighbourhood size, $\sigma^* > 0$, is used in equation (2) to calculate the distribution of the non-dominated solutions. $\mathcal{M}_1^*(Y')$ gives the average distance from Y' to $\bar{Y}$. $\mathcal{M}_2^*(Y')$ describes how well the solutions in Y' are distributed. $\mathcal{M}_2^*(Y')$ should produce a value between $[0, |Y'|]$ as it estimates the number of niches in Y' based on σ^*. The higher the value, the better the distribution is according to σ^*. $\mathcal{M}_3^*(Y')$ measures the spread of Y'.

6 Experiments

6.1 Function Optimization

For the purposes of this comparative study we have selected five well known problems from the literature; ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, and one rotated problem proposed by Deb [9]. These test problems are two-dimensional multidimensional minimization problems. ZDT1 is a 30 decision variable problem with a convex Pareto front which is continuous and uniformly distributed. ZDT2 is also a 30 dimensional problem, but has a non-convex Pareto front. ZDT3 is a 30 dimensional problem with 5 discontinuous non-convex fronts. ZDT4 is a 10 decision variable problem with 100 Pareto optimal fronts, only one of which is the global front. ZDT6 is a 10 dimensional problem with a non-uniform mapping between the objective space and parameter space.

6.2 Parameter Settings

We intend to compare NSCCGA's performance with the current state of the art, NSGA-II algorithm, therefore, we have used exactly the same real-coded recombination and selection processes as NSGA-II [9]. For each test function the NSGA-II executed 80,000 solution evaluations each run, where a single solution evaluation involves the calculation of all objective values for the individual. To make the comparison fair we attempted to use approximately the same number of solution evaluations for the NSCCGA. For the 30 and 10 dimensional problems we performed 80,240 evaluations and 80,080 evaluations respectively for each run of the NSCCGA. 30 runs of each algorithm were performed to acquire the necessary statistical significance for the performance metrics. The crowded tournament selection operator used a tournament size of 2.

A mutation rate of $\frac{1}{n}$ was used for the NSGA-II, where n is the number of parameters. This mutation rate was chosen because it is typically used within the literature reporting on the performance of this algorithm. η_c and η_m control the distribution of the crossover and mutation probabilities respectively and were assigned values of 20 each.

High mutation rates can disrupt smaller populations. The collaboration formation within the NSCCGA produces a relatively large number of candidate solutions each generation. Therefore, we might expect significant improvements

in performance with a higher mutation rate at the sub-population level, by exploring more of the search space without significant disruption to the exploitation process. Therefore, we also conducted experiments with relatively high mutation rates with the NSCCGA, and a mutation rate of 0.6 was found to demonstrate generally good performance across the ZDT test problems. We have not explored the optimality of control parameters further because this paper is primarily concerned with demonstrating the utility of a new collaborator selection mechanism.

A population size of 100 was used for the NSGA-II, where 100 children were added each generation. The NSCCGA used a population size of 200 individuals for the sub-population evolving x_1, and added 200 children to this population each generation. For each of the other populations 40 individuals were used, with 40 new individuals added to each population each generation. A constant sub-population size is maintained through an elitism operator which culls a proportion of the least fit individuals from each sub-population. A crossover rate of 0.9 was used for both the NSCCGA and NSGA-II. For the $\mathcal{M}_2^*$ metric, σ^* was set to 0.01.

7 Results and Discussion

Figure 2 shows the typical non-dominated fronts, and Table 1 tabulates the results acquired using the performance metrics $\mathcal{M}_1^*$, $\mathcal{M}_2^*$, and $\mathcal{M}_3^*$. From this table it is apparent that the NSCCGA is comparable in performance to the NSGA-II upon the ZDT test functions. The NSCCGA also has the advantage of being able to acquire a large number of diverse non-dominated solutions for an equivalent number of evalutions of the NSGA-II (Metric $\mathcal{M}_2^*$ in Table 1) when the mutation rate is sufficiently high. This is by virtue of the collaboration formation mechanism within the NSCCGA, where n populations with m individuals can potentially form nm solutions through collaboration.

Table 1. $\mathcal{M}_1^*$, $\mathcal{M}_2^*$, $\mathcal{M}_3^*$, and the number of evaluations (averaged over 30 runs).

Metric	Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{M}_1^*$	*NSCCGA*	1.15E-03 ±3.54E-05	8.66E-04 ±5.35E-05	3.27E-04 ±9.50E-05	1.39E-03 ±2.63E-04	2.58E-03 ±7.76E-04
	real-coded NSGA II	1.06E-03 ±1.46E-04	6.76E-04 ±2.76E-04	2.80E-04 ±3.09E-05	1.26E-03 ±3.21E-04	7.02E-02 ±7.24E-03
$\mathcal{M}_2^*$	*NSCCGA*	1.34E+03 ±8.71E+00	1.33E+03 ±9.01E+00	1.21E+03 ±1.32E+01	5.53E+02 ±1.96E-01	5.52E+02 ±4.82E-01
	real-coded NSGA II	9.94E+01 ±7.70E-02	8.61E+01 ±3.44E+01	9.81E+01 ±5.14E-01	9.62E+01 ±1.82E+01	9.91E+01 ±8.16E-02
$\mathcal{M}_3^*$	*NSCCGA*	1.42E+00 ±3.97E-03	1.42E+00 ±2.08E-02	1.62E+00 ±4.39E-03	1.42E+00 ±1.67E-02	1.29E+00 ±1.51E-02
	real-coded NSGA II	1.41E+00 ±2.26E-05	1.23E+00 ±4.89E-01	1.60E+00 ±5.10E-02	1.37E+00 ±2.58E-01	1.25E+00 ±3.11E-03

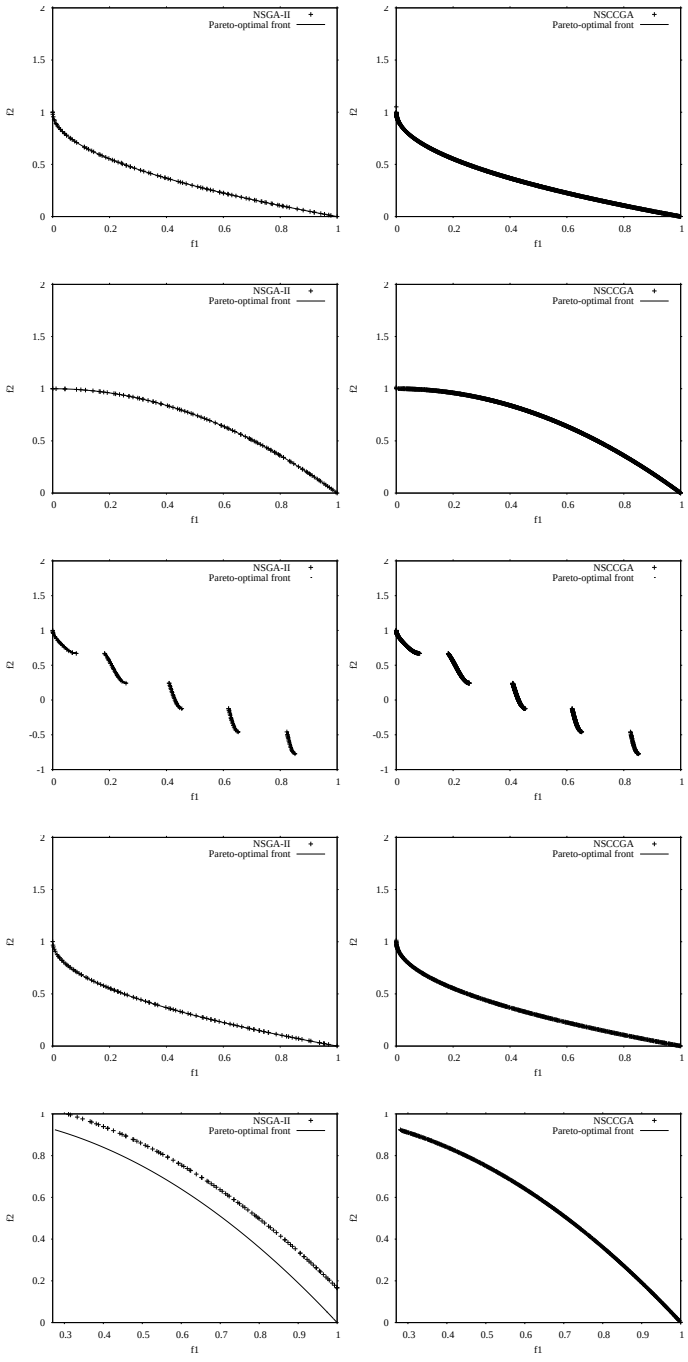

Fig. 2. From top to bottom are the typical non-dominated fronts of ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for the NSGA-II and NSCCGA.

The $\mathcal{M}_1^*$ metrics demonstrate comparable performance between the algorithms on all but the ZDT6 function, where the NSCCGA was able to converge closer to the Pareto-optimal front than the NSGA-II. It is also apparent that the NSCCGA was able to achieve a slightly better measure of spread across all the functions from the $\mathcal{M}_3^*$ metric. This can be understood in terms of the much

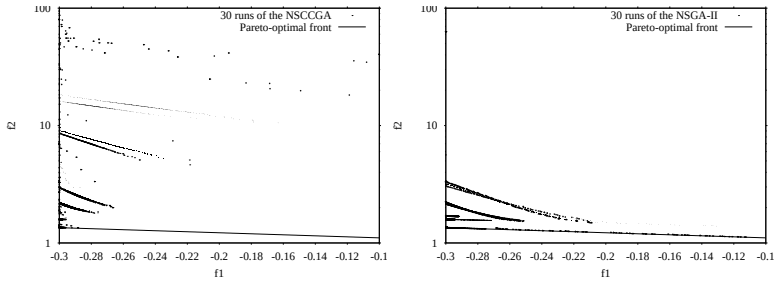

Fig. 3. The left and right plots respectively show 30 runs of the NSCCGA and the NSGA-II algorithm on the rotated problem after 80,000 evaluations. A crossover and mutation rate of 0.9 and $(1/n)$ was used for the NSGA-II, and a crossover and mutation rate of 0.9 and 0.6 was used with the NSCCGA.

larger number of collaborations which are formed within the NSCCGA, resulting in a greater likelyhood of good coverage across the front, including the extreme end points of the front.

7.1 Rotated Problems

Rotated problems introduce significant parameter interactions [21]. On the rotated problem, we have observed that the NSCCGA performed much worse than the NSGA-II. This is understandable because the NSCCGA assumes a problem that is decomposable, and can be solved by breaking it down into components. The CCA this work is based on has difficulties converging to good solutions with these types of problems (Figure 3). We have also observed that high mutation rates with the NSCCGA on the rotated problem improved the performance a little, because it increases the chance of simultaneous improvements in parameters [21], thereby enabling better solutions to be found more often.

8 Conclusions

This paper has demonstrated a novel collaboration formation mechanism within a cooperative coevolutionary algorithm, which utilises the unique aspects of multiobjective optimization. Through the cooperative coevolution of components which participate in candidate non-dominated solutions it is possible to find collaborators which cooperate in good solutions. This paper has demonstrated that

the NSCCGA compares well with the NSGA-II on a variety of test problems exhibiting diverse characteristics. However, the NSCCGA experienced difficulty in handling the rotated problem. More work is necessary on the NSCCGA in order to apply it to difficult real world problems, which may have significant parameter interactions. Two areas which we are considering to explore in the future are problem decomposition [22] and self-adaptation of control parameters [23]. Preliminary work in these areas has suggested that they are effective techniques.

References

1. Hillis, D.: Coevolving Parasites Improves Simulated Evolution as an Optimization Procedure. In: Langton, C. G., Taylor, C., Farmer, J. D., Rasmussen, S. (eds.): Artificial Life II - Proc. of the Workshop on the Synthesis and Simulation of Living Systems. Addison Wesley, Redwood City CA (1990) 313–324
2. Coello, C. A. C., Sierra, M. R.: A Coevolutionary Multi-objective Evolutionary Algorithm. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K. C., McKay, B., Essam, D., Gedeon, T. (eds.): Proc. 2003 Congress on Evolutionary Computation (CEC'03). IEEE Press, Piscataway NJ (2003) 482–489
3. Potter, M. A., De Jong, K. A.: Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. In: Evolutionary Computation, Vol. 8, No. 1. (2000) 1–29
4. Eriksson, R., Olsson, B.: Cooperative Coevolution in Inventory Control Optimisation. In: Smith, G. D., Steele, N. C., Albrecht, R. F. (eds.): Proc. of the Third International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA'97). Springer-Verlag, Berlin Germany (1997) 583–587
5. Potter, M. A., De Jong, K. A., Grefenstette, J. J.: A Coevolutionary Approach to Learning Sequential Decision Rules. In: Eshelman, L. (ed.): Proc. of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, (1995) 366–372
6. Tan, K. C., Yang, Y. J., Lee, T. H.: A Distributed Cooperative Coevolutionary Algorithm for Multiobjective Optimization. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K. C., McKay, B., Essam, D., Gedeon, T. (eds.): Proc. 2003 Congress on Evolutionary Computation (CEC'03). IEEE Press, Piscataway NJ (2003) 2513–2520
7. Keerativuttitumrong, N., Chaiyaratana, N., Varavithya V.: Multi-objective Cooperative Coevolutionary Genetic Algorithm. In: Merelo Guervs, J.J., Adamidis, P., Beyer, H.-G., Fernndez-Villacaas, J.-L., Schwefel, H.-P.(eds.): Proc. of Parallel Problem Solving From Nature VII (PPSN'02). Lecture Notes in Computer Science, Vol 2439. Springer-Verlag, Berlin Germany (2002) 288–297
8. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou, K.C., Tsahalis, D.T., Periaux, J., Papailiou, K.D., Fogarty, T. (eds.): EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. International Center for Numerical Methods in Engineering (Cmine), Barcelona Spain (2001) 95–100
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," In: IEEE Trans. Evol. Comput., Vol. 6, No. 2. (2002) 182–197

10. Fonsea, C. M., Fleming, P. J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization. In: Forrest S. (ed.): Proc. of the Fifth International Conference on Genetic Algorithms, Vol. 26. Morgan Kaufmann, Los Altos CA (1993) 30–45
11. Potter, M. A., De Jong K.: A Cooperative Coevolutionary Approach to Function Optimization. In: Proc. of Parallel Problem Solving From Nature III (PPSN III), Springer-Verlag, Berlin Germany (1995) 249–257
12. Mao, J., Hirasawa, K., Hu, J., Murata, J.: Genetic Symbiosis Algorithm for Multiobjective Optimization Problems. In: Beyer, H., Cantu-Paz, E., Goldberg, D., Parmee, I., Spector, L., Whitley, D. (eds.): Proc. 2001 Genetic and Evolutionary Computation Congress (GECCO'01). Morgan Kaufmann, (2001) 771
13. Barbosa, H. J. C., Barreto, A. M. S.: An Interactive Genetic Algorithm with Coevolution of Weights for Multiobjective Problems. In: Beyer, H., Cantu-Paz, E., Goldberg, D., Parmee, I., Spector, L., Whitley, D. (eds.): Proc. 2001 Genetic and Evolutionary Computation Congress (GECCO'01). Morgan Kaufmann, (2001) 203–210
14. Parmee, I. C., Watson, A. H.: Preliminary Airframe Design Using Co-evolutionary Multiobjective Genetic Algorithms. In: Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., Smith, R.E. (eds.): Proc. 1999 Genetic and Evolutionary Computation Congress (GECCO'99). Morgan Kaufmann, (1999) 1657–1665
15. Lohn, J., Kraus, W. F., Haith, G. L.: Comparing a Coevolutionary Genetic Algorithm for Multiobjective Optimization. In: Fogel, D., *et. al.* (eds.): Proc. 2002 Congress on Evolutionary Computation (CEC'02). IEEE Press, Piscataway NJ (2002) 1157–1162
16. Lohn, J., Haith, G., Colombano, S., Stassinopoulos, D.: A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers. In: Stoica, A., Keymeulen, D., Lohn, J. (eds.): The First NASA/DoD Workshop on Evolvable Hardware. IEEE Press, (1999) 87–92
17. Deb, K., Agrawal, R. B.: Simulated Binary Crossover for Continuous Search Space. In: Complex Systems, Vol. 9, No. 2. (1995) 115–148
18. Deb, K., Kumar, A.: Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multi-modal and Multi-objective Problems. In: Complex Systems, Vol. 9, No. 6. (1995) 431–454
19. Deb, K., Goyal, M.: A Combined Genetic Adaptive Search (GENEAS) for Engineering Design. In: Computer Science and Informatics, Vol. 26, No. 4. (1995) 30–45
20. Zitzler, E., Deb, K. and Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, **8**(2):173-195, April (2000).
21. Salomon, R.: Re-evaluating Genetic Algorithm Performance Under Coordinate Rotation of Benchmark Functions: A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms. In: Bio Systems, Vol. 39, No. 3. (1996) 263–278
22. Weicker, K., Weicker, N.: On the Improvement of Coevolutionary Optimizers by Learning Variable Interdependencies. In: Proc. 1999 Congress on Evolutionary Computation (CEC'99). IEEE Press, (1999) 1627–1632
23. Iorio, A. W., Li, X.: Parameter Control Within a Cooperative Coevolutionary Genetic Algorithm. In: Merelo Guervs, J.J., Adamidis, P., Beyer, H.-G., Fernndez-Villacaas, J.-L., Schwefel, H.-P.(eds.): Proc. of Parallel Problem Solving From Nature VII (PPSN'02). Lecture Notes in Computer Science, Vol 2439. Springer-Verlag, Berlin Germany (2002) 247–256