Multi-Island Competitive Cooperative Coevolution for Real Parameter Global Optimization

Kavitesh K. Bali and Rohitash Chandra

School of Computing Information and Mathematical Sciences
University of South Pacific, Suva, Fiji.
bali.kavitesh@gmail.com, c.rohitash@gmail.com

Abstract. Problem decomposition is an important attribute of cooperative coevolution that depends on the nature of the problems in terms of separability which is defined by the level of interaction amongst decision variables. Recent work in cooperative coevolution featured competition and collaboration of problem decomposition methods that was implemented as islands in a method known as competitive island cooperative coevolution (CICC). In this paper, a multi-island competitive cooperative coevolution algorithm (MICCC) is proposed in which several different problem decomposition strategies are given a chance to compete, collaborate and motivate other islands while converging to a common solution. The performance of MICCC is evaluated on eight different benchmark functions and are compared with CICC where only two islands were utilized. The results from the experimental analysis show that competition and collaboration of several different island can yield solutions with a quality better than the two-island competition algorithm (CICC) on most complex multi-modal problems.

1 Introduction

Coevolutionary algorithms have gained popularity as a vital extension to the traditional evolutionary algorithms [1]. Cooperative coevolution (CC) is one such evolutionary computation method which solves a problem by dividing it into subcomponents [2]. Essentially, cooperative coevolution has the ability to simplify the complexities of a problem through decomposition [3]. However, the performance of CC algorithms are highly sensitive to problem decomposition strategy [4]. Variable interaction [5] is a major constraint that governs the decomposition of a problem [6]. It is generally believed that placement of interacting variables into separate subcomponents degrades the optimization performance significantly [7, 8]. Inter-dependencies exists amongst decision variables specifically in non-separable and partially separable functions [9, 2, 10]. Grouping of interacting variables accurately into separate subcomponents hence becomes a challenge to CC. An efficient problem decomposition technique identifies and groups variables with inter-dependencies together [9]. For this reason, several decomposition mechanisms have been proposed that automatically capture and group interacting variables together [6, 11, 12, 13].

There is no unique decomposition strategy for some classes of problems such as fully-separable functions, fully-non separable or overlapping functions [14]. For instance, in a fully-separable function, all of the decision variables can be optimized independently, hence any decomposition is viable. Some partially separable functions may also contain a relatively high dimensional fully-separable subcomponent. Poor decomposition of such subcomponents may affect the optimization process and the solution quality [4]. Unfortunately, attempting to determine an effective decomposition strategy for these different classes of functions is a laborious task, which requires extensive experimentation [4].

It was shown that in spite of such automated decomposition strategies, it is still possible to identify a set of near optimal *static* decomposition [4]. However, it comes with the expense of elaborate empirical studies. To remedy the need for identifying the optimal decomposition, a very simple reinforcement learning approach to dynamically adapt the decomposition strategy was utilized [4].

An alternative method called competitive island-based cooperative coevolution (CICC) was proposed recently to eliminate the need for finding an optimal decomposition [15, 16, 17]. CICC has been originally designed for training recurrent neural networks on chaotic time series problems [15, 16]. In such a scenario, neural level and synapse level problem decomposition methods are implemented as islands that compete and collaborate with each other. The competition algorithm ensures that different problem decomposition methods are given an opportunity to compete in different phases during the course of evolution [17]. It was shown that CICC is a promising approach for different classes of global optimization problems [17]. By enforcing competition and collaboration of different problem decomposition strategies, the CICC framework could yield solutions with a quality better than individual decomposition strategies used in isolation [17]. Initially, the CICC method incorporated competition amongst only two different problem decomposition methods [17].

In this paper, a multi-island competitive cooperative coevolution (MICCC) algorithm is proposed in which several different problem decomposition strategies are given a chance to compete, collaborate and motivate other islands while converging to a common solution. The performance of MICCC using three and five islands are evaluated on eight different benchmark functions and compared with CICC where only two islands were used.

The organization of the rest of this paper is as follows. Section 2 describes the proposed method and its application to different classes of problems. Experimental results and their analysis are provided in Section 3. Finally, Section 4 concludes the paper and outlines possible future extensions.

2 Multi-Island Competitive Cooperative Coevolution

In this section, we extend the two-island competition algorithm (CICC) applied in [15, 16, 17] to a multi-island cooperative coevolution (MICCC) algorithm which enforces competition and collaboration between various different problem decomposition strategies that are implemented as islands. In competitive coevolution, individuals of a species show its competitive ability through its fitness scores. The individuals of higher fitness win in the competition and continuously improve their performance through evolution of populations [18].

Algorithm 1: Multi-Island Competitive Cooperative Coevolution

```
Stage 1: Initialization:
while Island-n ≤ MaxNumIslands do
 | Cooperatively evaluate Island-n
end

Stage 2: Evolution:
while FE ≤ Global-Evolution-Time do
    while Island-n ≤ MaxNumIslands do
        while FE ≤ Island-Evolution-Time do
            foreach Sub-population at Island-n do
                foreach Cycle in Max-Cycles do
                    foreach Generation in Max-Generations do
                        Create new individuals using genetic operators
                        Cooperative Evaluation of Island-n
                    end
                end
            end
        end
    end

    Stage 3: Competition: Compare and mark the island with the best fitness.

    Stage 4: Collaboration: Inject the best individual from the winner island into all the
    other islands.
end
```

In MICCC [17], several different uniform problem decomposition decompositions are constructed as islands that compete and collaborate to optimize a function. These islands are evolved in isolation by independent G3-PCX [19] algorithm. The islands enforce competition by comparing their solutions after a fixed time (implemented as fitness evaluations), and exchange the best solution between the islands. The solution migration occurs between the different islands when evolutionary processes carry on for defined fitness evaluations or generations; by migrating feasible solutions of the winner island into those who lose the competition. The key aspects of the proposed MICCC algorithm are initialization, evolution, competition and collaboration.

2.1 Initialization

Each different problem decomposition strategy (island) is constructed with uniform problem decomposition strategies. To enforce an unbiased competition, all the different islands begin search with the same genetic materials in their subpopulation.

Initially, all the sub-populations of Island One are initialized with random-real number values from a domain specified in Table 1. Next, these real values (from Island One) are copied into the sub-populations of the rest of the islands that are constructed with unique problem decompositions. In MICCC algorithm, the number of fitness evaluations depends on the number of sub-populations implemented in an island. An island with higher number of subcomponents will basically acquire more fitness evaluations for each cycle. Since each island is simultaneously evolved in isolation for complete cycles, the number of fitness evaluations cannot be exactly the same for each island due to unique problem decomposition strategies.

2.2 Cooperative Coevolution

After initialization, each of the islands are evolved in isolation simultaneously for a predefined time in the round robin fashion of the cooperative co-evolution framework. This predefined time is the *island-evolution time* that is established by the number of cycles that makes the required number of fitness evaluations for each of the respective islands. A single cycle completes when all the sub-populations of the respective island have been cooperatively evolved for a depth of n generations. The individuals from each of the sub-populations are cooperatively evaluated by concatenating the chosen individual from a given sub-population with the best individuals from the rest of the sub-populations [8].

2.3 Competition

In the competition phase of the MICCC algorithm, fitness ranking and comparison of all the cooperative evolved islands are conducted. A ranking process is adapted in order to identify the better and poor performing islands. This is done by quantifying the fitness of each of the islands at certain time intervals. The islands with higher fitness are ranked high while the poor performing islands with lower fitness are ranked lower. In MICCC, the island producing the highest fitness is the winner and ranked as the best island at that point of check. For the case where the fitness is the same, the winner island is randomly selected to encourage a fair competition.

2.4 Collaboration - Solution Migration

Collaboration is the core feature of the MICCC framework whereby the actual interaction and migration between different islands occur. Here, the best solution of the winner island is copied and injected into to the runner-up islands. This migration of the best feasible solution is able assist and motivate the other islands to compete fairly in the next round.

The transfer of best solutions from one island to the rest is done via the context vector [20]. As an island wins, the best individuals from each of the subcomponents need to be carefully concatenated into a context vector. The best solutions are then split from the context vector and are then injected into the respective subcomponents of each of the runner-up islands. The runner-up island which receives the best (injected) solution is cooperatively evaluated to ensure that the newly injected solution has a fitness. The best fitness of the winning island is also transferred alongside the best solution to the rest of the islands. Moreover, since the fitness of the best solution from the last sub-population carries a stronger solution, this fitness value is transferred and is used to override the fitness of the best solutions of all the sub-populations of the runner-up islands.

3 Simulation and Analysis

In this section, we evaluate the performances the multi-island instances; three and five island algorithms of MICCC and compare them with the standalone CC implementations. Next, we compare these multi-island instances with the original two island algorithm [17].

3.1 Benchmark Problems and Configuration

The experimental results in this paper are based on eight benchmark functions used in [17]. These functions are selected considering the level of difficulty, the scope of separability and the nature of problem,i.e. unimodal or multi-modal listed by Table 1. Furthermore, we use different problem decomposition strategies of 100 dimensions as inputs for competition in the multi-island algorithms. According to [21], these values that represent low, medium and high dimensional subcomponent sizes allow us to approximately determine the optimal subcomponent size.

The generalized generation gap with parent-centric crossover evolutionary algorithm (G3-PCX) [19] is used as the subcomponent optimizer. We use a pool size of 2 parents and 2 offspring as presented in [19]. The mean and standard deviation of function errors *(f(x)-f(*x))* of 25 runs for each of the experiments are reported in the next subsection. The maximum number of fitness evaluations was set to 1500000 as suggested in [17]. The number of individuals in each of the respective sub-populations are fixed at 100.

3.2 Results and Analysis

In this section, we evaluate the performance of the three island algorithm on the different classes of functions f_1 to f_8. We observe that the proposed method has produced better quality solutions than the respective standalone CC counterparts according to Table 2. It has improved the solution quality of the unimodal and fully-separable functions f_1 - f_3. It has performed fairly well for the multi-modal non-separable Rosenbrock instances of f_4 and f_5. The three island algorithm performed better than the standalone CC implementations on f_5. For

Table 1. Problem Definitions based on [22, 10, 23]

Problem	Name	Optimum	Range	Multi-modal	Fully Separable
f_1	Ellipsoid	0	[-5,5]	No	Yes
f_2	Shifted Sphere	-450	[-100,100]	No	Yes
f_3	Schwefel's Problem 1.2	0	[-5,5]	No	Yes
f_4	Rosenbrock	0	[-5,5]	Yes	No
f_5	Shifted Rosenbrock	390	[-100,100]	Yes	No
f_6	Rastrigin	0	[-5,5]	Yes	Yes
f_7	Shifted Rastrigin	-330	[-5,5]	Yes	Yes
f_8	Shifted Griewank	-180	[-600,600]	Yes	No

Table 2. Comparison of MICCC-3 Island results against individual decomposition strategies used in isolation (CC)

		Standard CC			
Functions	Stats.	20×5	10×10	4×25	**MICCC - 3 Island**
f_1	Mean	6.83e+00	4.80e-98	3.76e-98	2.51e-101
	StDev	1.83e+00	1.70e-98	1.82e-98	1.41e-101
f_2	Mean	3.43e+05	9.03e-03	1.08e-12	4.84e-13
	StDev	1.40e+04	2.03e-03	1.06e-12	1.31e-13
f_3	Mean	5.00e-03	7.98e-51	1.13e-50	0.00e+00
	StDev	1.00e-04	2.01e-50	1.12e-50	0.00e+00
f_4	Mean	2.59e+02	5.16e+01	1.11e+02	5.90e+01
	StDev	2.49e+02	2.01e+00	1.01e+01	1.02e+00
f_5	Mean	7.95e+10	3.80e+01	9.01e+01	0.00e+00
	StDev	1.01e+09	1.02e+01	1.60e+01	0.00e+00
f_6	Mean	1.87e+01	2.70e+02	4.86e+02	0.60e+01
	StDev	1.08e+00	1.02e+01	1.89e+01	0.66e+00
f_7	Mean	8.83e+02	5.02e+02	7.56e+02	3.53e+02
	StDev	5.04e+01	2.02e+01	4.02e+01	1.04e+01
f_8	Mean	2.81e+3	5.11e-13	3.63e-03	1.98e-13
	StDev	2.08e+02	1.14e-13	1.07e-03	2.03e-14

the multi-modal, separable Rastrigin functions f_6 and f_7, the three island algorithm achieved better results than the standalone counterparts and recorded similar performance for f_8.

We observe a similar trend while analyzing the performance of the five island algorithm. The results in Table 3 suggest that we attain better quality solutions while competing the five islands than each of those evolved in isolation as a standalone CC. This can be observed for all the test functions f_1 - f_8. The five island competition has shown improved and near optimal solutions for the unimodal f_1 - f_3, and multi-modal functions f_5 and f_8. Moreover, it generated better solutions than the standalone CCs' for the Rastrigin instances f_6 and f_7. For, f_4, the five island algorithm provided better solution quality (error) than 4 out of 5 standalone decompositions tested.

Table 3. Comparison of MICCC-5 Island results against individual decomposition strategies used in isolation (CC)

Functions	Stats.	Standard CC					**MICCC - 5 Island**
		20×5	10×10	4×25	5×20	50×2	
f_1	Mean	6.83e+00	4.80e-98	3.76e-98	5.62e-98	2.01e-98	5.59e-99
	StDev	2.03e+00	2.36e-98	1.45e-98	1.63e-98	1.70e-98	1.02e-99
f_2	Mean	3.43e+05	9.03e-03	1.08e-12	8.98e-13	0.27e+01	1.71e-13
	StDev	1.45e+04	1.03e-03	2.03e-13	1.21e-13	2.34e-01	1.03e-14
f_3	Mean	5.00e-03	7.98e-51	1.13e-50	1.18e-50	1.54e-50	0.00e+00
	StDev	1.00e-03	6.98e-51	1.03e-50	1.07e-50	2.01e-50	0.00e+00
f_4	Mean	2.59e+02	5.16e+01	1.11e+02	6.31e-01	1.57e+02	7.43e+01
	StDev	2.30e+01	1.01e+01	1.30e+01	2.03e-01	1.34e+01	0.78e+01
f_5	Mean	7.95e+10	3.80e+01	9.01e+01	3.71e+01	5.67e+04	0.00e+00
	StDev	1.96e+10	1.02e+01	1.20e+01	0.34e+01	3.49e+03	0.00e+00
f_6	Mean	1.87e+01	2.70e+02	4.86e+02	4.73e+02	1.10e+02	9.95e-01
	StDev	0.43e+01	1.20e+02	2.21e+02	1.03e+02	0.70e+02	1.45e-01
f_7	Mean	8.83e+02	5.02e+02	7.56e+02	6.91e+02	1.22e+02	9.75e+01
	StDev	2.23e+02	1.34e+02	2.78e+02	0.45e+02	0.94e+02	1.45e+01
f_8	Mean	2.81e+03	5.11e-13	3.63e-03	5.12e-13	0.43e+01	8.53e-14
	StDev	1.02e+03	2.09e-13	1.67e-03	2.87e-13	0.23e+01	1.04e-14

3.3 Discussion

This paper proposed multi-island competitive cooperative coevolution that involves increasing the number of islands in the original competitive island cooperative coevolution, CICC [17] which was limited to two islands. The experimental results show that we get better quality solutions than the CCs' with standalone decomposition strategies. Table 4 provides a set of comparative data for two, three and five island setups of CICC tested on eight benchmark functions f_1-f_8. According to Table 4, it is evident that all the three different setups performed equally well on unimodal and fully separable functions f_1-f_3 as they recorded similar solution errors. However, we observe an interesting trend with the complex multi-modal problems f_4, f_6, f_7 and f_8. We notice that the quality and precision of the solutions improve while utilizing more islands in the competition. The five island algorithm outperformed the three and two island algorithms on the multi-modal problems. Moreover, the three island algorithm also performed better than two island algorithm for the same set of problems. The performance improvement with the multi-island algorithm is mainly because many different problem decomposition strategies are given a chance to compete and at the same time collaborate to help and motivate the poorly performing problem decompositions through solution migrations of the winning island. For instance, few standalone problem decomposition strategies such as *(*25×5*) and (*50×2*)* in Table 3 have been inefficient and possibly been victims of premature convergence for most multi-modal problems (f_5-f_8) because it recorded poor quality solutions overall when evolved in isolation as a standalone CC. However, if the same low-performing problem decomposition strategies are incorporated as com-

Table 4. Comparison of CICC against MICCC

Functions	Stats.	CICC Versions		
		CICC[17]	**MICCC - 3 Island**	**MICCC - 5 Island**
f_1	Mean	3.76e-99	2.51e-101	5.59e-99
	StDev	2.06e-99	1.41e-101	1.02e-99
f_2	Mean	7.78e-13	4.84e-13	1.73e-13
	StDev	1.34e-13	1.31e-13	1.03e-14
f_3	Mean	0.00e+00	0.00e+00	0.00e+00
	StDev	0.00e+00	0.00e+00	0.00e+00
f_4	Mean	7.93e+01	5.90e+01	7.43e+01
	StDev	1.78e+01	1.02e+01	0.78e+01
f_5	Mean	0.00e+00	0.00e+00	0.00e+00
	StDev	0.00e+00	0.00e+00	0.00e+00
f_6	Mean	1.40e+01	0.60e+01	9.95e-01
	StDev	1.34e+00	0.66e+00	1.45e-01
f_7	Mean	3.92e+02	3.53e+02	9.75e+01
	StDev	2.09e+01	1.04e+01	1.45e+01
f_8	Mean	1.99e-13	1.98e-13	8.53e-14
	StDev	1.98e-13	2.03e-14	1.04e-14

petition inputs to the multi-island algorithm, it performs better and recorded substantial improvement in the quality of fitness solutions. As we increase the number of islands, different problem decomposition strategies compete and collaborate with the exchange of the best genetic materials from the winning islands during the evolution. The performance of the island with sub-populations with a lower diversity is also revived during the migration of individuals. To maintain high quality solutions, it is essential to have distinct problem decomposition strategies implemented as islands.

4 Conclusions and Future Work

In this paper, we extended the two-island competitive cooperative coevolution algorithm (CICC) to multiple island approaches where we evaluated the performances of the competitive methods for three and five islands with substantial analyses. The experimental results show that enforcing competition with a wider pool of problem decomposition strategies can considerably improve the performance during the course of the optimization phase, and also shows substantial enhancements in the quality of the overall fitness solutions. As we increase the number of islands for competition, more diversity is introduced. Different islands compete and cooperate through the transfer of the best genetic materials from the winner island. We observe an appealing trend with MICCC for most multi-modal problems, whereby we attain higher quality solutions and basically escape the vulnerable fitness stagnation trap. In future, we are interested in further improving the solutions by allowing more individuals to share their solutions which would improve diversity and intensify selection pressure. This proposed multi-island algorithm can also be applied to large scale global optimization and extended to combinatorial optimization problems.

References

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Institute of Physics Publishing, Bristol, and Oxford University Press, New York, 1997.

[2] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel problem solving from naturePPSN III*. Springer, 1994, pp. 249–257.

[3] R. Chandra, M. Frean, and M. Zhang, "On the issue of separability for problem decomposition in cooperative neuro-evolution," *Neurocomputing*, vol. 87, pp. 33–40, 2012.

[4] M. N. Omidvar, Y. Mei, and X. Li, "Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms," in *Proc. of IEEE Congress on Evolutionary Computation*, 2014, pp. 1305–1312.

[5] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms," *BioSystems*, vol. 39, pp. 263–278, 1995.

[6] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 378–393, June 2014.

[7] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2. IEEE, 2001, pp. 1101–1108.

[8] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving from Nature*. Springer, 1994, pp. 249–257.

[9] R. Salomon, "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms," *BioSystems*, vol. 39, no. 3, pp. 263–278, 1996.

[10] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization," RMIT University, Melbourne, Australia, Tech. Rep., 2013, http://goanna.cs.rmit.edu.au/ xiaodong/cec13-lsgo.

[11] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Cooperative co-evolution with a new decomposition method for large-scale optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014*, 2014, pp. 1285–1292.

[12] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Proc. of International Conference on Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 6239. Springer Berlin / Heidelberg, 2011, pp. 300–309.

[13] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Proc. of IEEE Congress on Evolutionary Computation*, 2010, pp. 1762–1769.

[14] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Information Sciences*, 2015.

[15] R. Chandra, “Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction,” *Neural Networks and Learning Systems, IEEE Transactions on*, p. In Press, 2015.

[16] ——, “Competitive two-island cooperative coevolution for training Elman recurrent networks for time series prediction,” in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 565–572.

[17] R. Chandra and K. Bali, “Competitive two island cooperative coevolution for real parameter global optimization,” in *IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 2015, pp. 93–100.

[18] W. Li and L. Wang, “A competitive-cooperative co-evolutionary optimization algorithm based on cloud model,” in *Advanced Computational Intelligence (IWACI), 2011 Fourth International Workshop on.* IEEE, 2011, pp. 662–669.

[19] K. Deb, A. Anand, and D. Joshi, “A computationally efficient evolutionary algorithm for real-parameter optimization,” *Evolutionary computation*, vol. 10, no. 4, pp. 371–395, 2002.

[20] F. Van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 225–239, 2004.

[21] M. N. Omidvar, Y. Mei, and X. Li, “Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on.* IEEE, 2014, pp. 1305–1312.

[22] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, , and Z. Yang, “Benchmark functions for the CEC’2008 special session and competition on large scale global optimization,” Nature Inspired Computation and Applications Laboratory, USTC, China, Tech. Rep., 2007, http://nical.ustc.edu.cn/cec08ss.php.

[23] F. Herrera, M. Lozano, and D. Molina, “Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems,” 2010.