

A New Dimension Division Scheme for Heterogeneous Multi-Population Cultural Algorithm

Mohammad R. Raeesi N. and Joshua Chittle and Ziad Kobti

School of Computer Science, University of Windsor, 401 Sunset Avenue, Windsor, ON, Canada N9B 3P4
raeesim@uwindsor.ca and chittlej@uwindsor.ca and kobti@uwindsor.ca

Abstract

Heterogeneous Multi-Population Cultural Algorithm (HMP-CA) is a new class of Multi-Population Cultural Algorithms which incorporates a number of local Cultural Algorithms (CAs) designed to optimize different subsets of the dimensions of a given problem. In this article, various dimension decomposition techniques for HMP-CAs are proposed and compared. The concept of using a dimension decomposition scheme which does not result in populations having the same number of dimensions is implemented, and named imbalanced dimension division. All the techniques are evaluated using a number of well-known benchmark optimization functions and two measures are defined in order to compare them including success rate and convergence ratio. The results show that imbalanced dimension division schema works better with a higher number of local CAs, and outperforms all the other evaluated techniques in both measures.

Introduction

Evolutionary Algorithms (EAs) are subset of optimization methods successfully applied to solve problems in various research areas. EAs are population-based approaches incorporating the concept of evolution inspired by natural selection; that individuals which survive for the next generation are more likely to be fitter for their environment. An EA starts with a population of randomly generated individuals called the initial population. New populations are generated by applying evolutionary operators on the individuals from the previous population, followed by incorporating a selection method afterward for finalization. This routine halts when a predefined termination criterion (*e.g.* CPU time, predefined number of generations) has been met.

A Cultural Algorithm (CA) is an EA incorporating knowledge to improve the search mechanism. In a CA, knowledge is extracted, stored, and updated in a space separate from the population, called the belief space. The recorded knowledge in the belief space is then used to direct the search process during evolution.

Although CAs are successfully applied to various types of optimization problems, they suffer from immature con-

vergence. There are a number of strategies introduced to resolve converging to local optima, however researchers have recently been interested in incorporating multiple populations to increase diversity and escape to the global optimum. The multiple populations approach divides the whole population into a number of sub-populations, with each sub-population being evolved using a local CA independently while the local CAs communicate with each other between generations.

In addition to incorporating multiple populations in CAs, we recently showed that incorporating heterogeneous local CAs offers better solutions compared to the homogeneous ones. We proposed a Heterogeneous Multi-Population Cultural Algorithm (HMP-CA) to deal with numerical optimization problems where the goal is to optimize mathematical functions with a D -dimensional vector of continuous parameters (Raeesi N. and Kobti 2013). The results showed that dividing the dimensions among the local CAs improves the convergence rate. Furthermore, by incorporating partial solutions our published method saves both memory and CPU usage while handling the same total number of individuals in each generation.

There are various techniques to divide the dimensions among local CAs in a HMP-CA. In this article, an analysis is conducted to study the performance of these techniques. Our published HMP-CA (Raeesi N. and Kobti 2013) is modified such that it is capable of dealing with a number of dimension division strategies and any number of local CAs. The modified HMP-CA includes a number of heterogeneous sub-populations with a shared belief space. Each sub-population includes only partial solutions corresponding to its assigned dimensions and sends its best partial solutions to the belief space every generation. For each dimension, the shared belief space keeps the best parameters with their corresponding objective value. In order to evaluate the strategies, a number of benchmark numerical optimization functions are incorporated. Each strategy is investigated with variable number of sub-populations.

The structure of this article is as follows. Section *Cultural Algorithm* briefly describes CA and its multi-population version, followed by a concise description of HMP-CA in Section *Heterogeneous Multi-Population Cultural Algorithm*. Section *Proposed Method* illustrates the proposed HMP-CA in detail, followed by representing the investigated dimen-

sion decomposition strategies in Section *Dimension Decomposition Strategies*. Section *Experiments and Results* represents the experiments and the discussion on the results, followed by representing concluding remarks and future directions in Section *Conclusions*.

Cultural Algorithm

Cultural Algorithm (CA) as developed by Reynolds (1994) is an EA which extracts knowledge during the evolutionary process in order to redirect the search process. CA incorporates two spaces, namely a population space and a belief space. The population space evolves individuals over generations while the belief space is responsible for storing and updating the extracted knowledge. CAs incorporate a two-way communication between the spaces, namely an acceptance function which transfers the best individuals from the population space to the belief space and an influence function which carries the extracted knowledge from the belief space to influence the operations in the population space. The evolution process in population space can be managed by any evolutionary algorithm such as a Genetic Algorithm (GA) or a Differential Evolution (DE).

In spite of their successful application in various fields, CAs suffer from immature convergence. The main reason is the fact that they do not preserve population diversity over generations. There are a number of strategies introduced to overcome this limitation such as rejecting duplicate solutions or having a high mutation rate. Incorporating multiple populations is another such strategy which represents a good performance in order to do so. The first Multi-Population Cultural Algorithm (MP-CA) was introduced by Digalakis and Margaritis (2002) which was incorporated to schedule electrical generators. The main characteristic of a MP-CA is its architecture which determines the knowledge propagation within the local CAs. There are different architectures proposed to implement a MP-CA. The most common one is homogeneous local CAs (Xu, Zhang, and Gu 2010; Guo et al. 2011; Raeesi N. and Kobti 2012) in which there are a number of homogeneous local CAs with their own local belief spaces cooperating to find the best solution.

Heterogeneous local CAs is another class of architectures for MP-CA where the sub-populations are heterogeneous such that each sub-populations is working on different dimensions. Lin *et al.* (Lin, Chen, and Lin 2008; 2009; Lin et al. 2009) introduced heterogeneous local CAs by proposing their Cultural Cooperative Particle Swarm Optimization (CCPSO) to train a Neurofuzzy Inference System (NFIS). In CCPSO, each local CA has its own local belief space and incorporates a Particle Swarm Optimization (PSO) to evolve its corresponding sub-population. In this framework each local CA is responsible for optimizing only one variable.

In addition to the architecture, incorporating multiple populations brings more algorithm parameters to be adjusted in order to get a good performance. The number of sub-populations, the communication topology, and the type of migrated knowledge are three instances of MP-CAs parameters.

Heterogeneous Multi-Population Cultural Algorithm

Heterogeneous Multi-Population Cultural Algorithm (HMP-CA) is an architecture in the class of Heterogeneous local CAs in which there is only one belief space shared among all local CAs (Raeesi N. and Kobti 2013). In this architecture, there are a number of local CAs working on optimization of different subsets of the all dimensions. The only one shared belief space is responsible to keep a track of the best parameters found for each dimension. The focus of this paper is on this architecture where there are various strategies to divide dimensions among local CAs.

In HMP-CA, each local CA is designed to handle only its assigned dimensions. Therefore instead of complete solutions, each sub-population provides a set of partial solutions. For instance, if a local CA is responsible to optimize the first three dimensions of a 30-dimensional optimization problem, it only works with the values for the first three dimensions of a solution meaning only a 3-dimensional vector is used instead of a complete 30-dimensional solution. In other words, in this example a sub-population including 100 partial solutions deals with only 300 parameters in each generation while a sub-population with the same number of complete solutions incorporates 3,000 parameters. Due to this huge deduction in the number of parameters for each generation, the HMP-CA is an efficient method in terms of both CPU time and memory usage.

It should be mentioned here that in order to evaluate a partial solution, it is completed by the complement of its parameters coming from the belief space. This mechanism provides a fair comparison platform for partial solutions such that all the partial solutions are completed with the same parameters complement.

Heterogeneous local CAs incorporating either a shared belief space or local belief spaces are initialized by dividing the dimensions. Lin *et al.* (Lin, Chen, and Lin 2008; 2009; Lin et al. 2009) assigned each dimension to a local CA, and in our published HMP-CA (Raeesi N. and Kobti 2013) a jumping strategy was incorporated for dimension decomposition. There are more dimension division strategies, the most common ones are investigated in this study which are represented in Section *Dimension Decomposition Strategies*.

Proposed Method

The architecture of the proposed method is similar to our recently published HMP-CA (Raeesi N. and Kobti 2013). A number of sub-populations including only partial solutions are incorporated alongside a shared belief space. In the published HMP-CA, sub-populations cooperate with each other by transferring only their best partial solution to the belief space, and the belief space updates its only one parameter for each dimension with its corresponding objective value. In contrast, the architecture of the proposed method provides the flexibility to set the number of best partial solutions to be transferred to the belief space as well as to set the number of parameters to be stored in the belief space for each dimension.

Table 1: A sample belief space with size 3.

Dimensions		1	2	3	4	5
Set 1	Parameters	0.0	0.0	-0.88	-0.23	-0.43
	Objective	1.01	1.01	1.01	1.01	1.01
Set 2	Parameters	0.01	-0.26	1.06	-0.24	-0.44
	Objective	1.02	1.09	1.37	1.01	1.01
Set 3	Parameters	0.39	-0.31	-0.73	-0.62	0.79
	Objective	1.19	1.12	1.49	1.35	1.79

Furthermore, the published HMP-CA incorporates only one static dimension division strategy, while the proposed method is capable to deal with any dimension decomposition technique such as sequential, logarithmic, and even customized ones.

Like the published HMP-CA, in order to evolve the sub-population a simple DE is incorporated which benefits from $DE/rand/1$ mutation represented in Equation 1, binomial crossover illustrated in Equation 2, and a selection mechanism depicted in Equation 3. Due to the extensive experiments conducted in the published HMP-CA, the scale factor for Equation 1 is selected randomly from intervals $[0.5, 2.5]$ for each generation and the crossover probability for Equation 2 is set to 0.5 for all generations.

$$V_{i,g} = X_{r1,g} + F \times (X_{r2,g} - X_{r3,g}) \quad (1)$$

where $X_{r1,g}$, $X_{r2,g}$, and $X_{r3,g}$ are three randomly selected target vectors within the same generation g . F is a scale factor to determine how much the base vector $X_{r1,g}$ should be perturbed by the difference of the other two.

$$z_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } r_j \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (2)$$

where r_j is a real number randomly selected within the interval $[0, 1)$ for j^{th} dimension, and Cr is the crossover probability. In order to ensure that the trial vector $Z_{i,g}$ differs from target vector $X_{i,g}$ at least in one component, j_{rand} is randomly selected as the index of the different dimension.

$$X_{i,g+1} = \begin{cases} Z_{i,g} & \text{if } f(Z_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} & \text{otherwise} \end{cases} \quad (3)$$

where $X_{i,g}$ and $Z_{i,g}$ denote a target vector and its corresponding trial in generation g , respectively and $X_{i,g+1}$ represents the selected target vector for the next generation.

A shared belief space is incorporated in the proposed HMP-CA which tracks the best found parameters with their corresponding objective values for each dimension. A sample belief space of size 3 is illustrated in Table 1 which is incorporated to solve a 5-dimensional sphere model.

The framework of the belief space starts with receiving a number of best partial solutions from a local CA. For each parameter of each partial solution, the corresponding records are reviewed in order to avoid pushing duplicate parameters. Therefore, for the existing parameters only the objective value is updated and the new parameters are inserted into the belief space if they have a better objective value.

The belief space influences the search direction only in fitness evaluations. When a local CA requires a partial solution to be evaluated the belief space is queried for the complement of its set of parameters, the belief space returns the complement parameters which are randomly selected from its recorded parameters.

The number of local CAs in the proposed HMP-CA and their corresponding assigned dimensions are dependent to its dimension division strategy. Therefore the proposed HMP-CA starts by receiving a dimension division strategy. Based on the given strategy, each local CA is initialized with its corresponding sub-population of randomly generated partial solutions. The local CAs are evolved incorporating the previously mentioned DE for the maximum number of generations such that they transfer their best partial solutions to the belief space in every generation and they request the belief space for the complement parameters for each fitness evaluation.

In order to avoid trapping into local optimal regions within a sub-population, a re-initialization mechanism is incorporated such that if a local CA cannot find a better solution for a number of generations its sub-population will be re-initialized with random partial solutions while the belief space continues with its recorded parameters. It should be mentioned that in the proposed HMP-CA, there is no local search method incorporated to speed up the convergence.

Dimension Decomposition Strategies

In this article, a number of different dimension division strategies are studied which can be categorized into two classes including balanced and imbalanced techniques. The division strategies which assigns the same number of dimensions to each local CA is classified as balanced divisions. In balanced divisions, the difference of the number of dimensions between any two local CAs is at most one. All decomposition strategies where this difference could be greater than one are considered to be imbalanced division methods.

Within the balanced category, sequential division, jumping division, and overlapped sequential division are investigated. The jumping strategy assigns the adjacent dimensions to different local CAs while the sequential strategy assigns the adjacent dimensions to the same local CA. The overlapped sequential division is a modified version of sequential division such that each dimension is assigned to two different local CAs.

A logarithmic division method is considered as well, belonging to the category of imbalanced strategies. It starts by assigning all the dimensions to the first local CA. Two additional local CAs are generated by selecting each half of the dimensions of the first local CA. Remaining local CAs are generated in a similar fashion by recursively dividing the dimensions in half.

Table 2 illustrates the described dimension decomposition strategies dividing 30 dimensions among 15 local CAs. As mentioned before, the balanced strategies assign almost the same number of dimensions to each local CA, while the imbalanced ones assign various number of dimensions to each local CA ranging from one dimension to the all dimensions.

Table 2: Dividing 30 dimension into 15 local CAs incorporating different strategies.

Local CAs	Balanced			Imbalanced
	Jumping	Sequential	Overlapped Sequential	Logarithmic
1	1,16	1, 2	1, 2, 3, 4	1, 2, 3, ..., 30
2	2, 17	3, 4	3, 4, 5, 6	1, 2, 3, ..., 15
3	3, 18	5, 6	5, 6, 7, 8	16, 17, 18, ..., 30
4	4, 19	7, 8	7, 8, 9, 10	1, 2, 3, ..., 8
5	5, 20	9, 10	9, 10, 11, 12	9, 10, 11, ..., 15
6	6, 21	11, 12	11, 12, 13, 14	16, 17, 18, ..., 23
7	7, 22	13, 14	13, 14, 15, 16	24, 25, 26, ..., 30
8	8, 23	15, 16	15, 16, 17, 18	1, 2, 3, 4
9	9, 24	17, 18	17, 18, 19, 20	5, 6, 7, 8
10	10, 25	19, 20	19, 20, 21, 22	9, 10, 11, 12
11	11, 26	21, 22	21, 22, 23, 24	13, 14, 15
12	12, 27	23, 24	23, 24, 25, 26	16, 17, 18, 19
13	13, 28	25, 26	25, 26, 27, 28	20, 21, 22, 23
14	14, 29	27, 28	27, 28, 29, 30	24, 25, 26, 27
15	15, 30	29, 30	29, 30, 1, 2	28, 29, 30

For a 30-dimensional optimization problem, the logarithmic strategy can generate up to 59 local CAs including one local CA with 30 dimensions, two local CAs with 15 dimensions, 4 local CA with 7 or 8 dimensions, 8 local CAs with 3 or 4 dimensions, 14 local CAs of 2 dimensions, and 30 local CAs including only one dimension.

In this article, a new strategy similar to the logarithmic strategy is also defined which is called customized logarithmic. The new strategy incorporates 35 local CAs for a 30-dimensional problem which includes 5 local CAs with the following 5 dimension subsets in addition to the 30 local CAs with one dimension each.

$$\{1, 2, 3, \dots, 30\}, \{1, 2, 3, \dots, 15\}, \{16, 17, 18, \dots, 30\}, \\ \{1, 2, 3, \dots, 8\}, \{16, 17, 18, \dots, 23\}$$

Experiments and Results

In order to evaluate the mentioned strategies, they are incorporated by the proposed HMP-CA considering the total population size to be set to 1000 individuals which can be divided into any number of sub-populations evenly. Considering 15 local CAs in the proposed HMP-CA, each local CA will have 66 solutions within its sub-population. The size of the belief space is set to 3 and sub-populations transfer their 3 best solutions to the belief space in each generation. The maximum generation to find the optimal solution is considered to be 10,000 generations, and if an optimal solution is not obtained before this upper limit the run is considered to be an unsuccessful one. For each experiment, 100 independent runs were conducted to provide a statistically significant sample size.

A number of well-known numerical optimization functions were considered for the experiments which have been used by various researchers (Yu and Zhang 2011; Mezura-Montes, Velazquez-Reyes, and Coello Coello 2006). The 12 considered benchmark problems are listed as follows and are detailed in the aforementioned references:

- f_1 - Sphere Model.
- f_2 - Generalized Rosenbrock's Function.
- f_3 - Generalized Schwefel's Problem 2.26.
- f_4 - Generalized Rastrigin's Function.
- f_5 - Ackley's Function.
- f_6 - Generalized Griewank's Function.
- f_7 and f_8 - Generalized Penalized Functions.
- f_9 - Schwefel's Problem 1.2.
- f_{10} - Schwefel's Problem 2.21.
- f_{11} - Schwefel's Problem 2.22.
- f_{12} - Step Function.

The generalized Rosenbrock's function (f_2), for instance is represented in Equation 4:

$$f_2(x) = \sum_{i=1}^{D-1} \left| 100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right| \quad (4)$$

$$-30 \leq x_i \leq 30$$

$$\min(f_2) = f_2(1, \dots, 1) = 0$$

In addition to these 12 functions, a modified version of the function f_2 is considered in this analysis which is called modified Rosenbrock's function (f_{2M}) illustrated in Equation 5:

$$f_{2M}(x) = \sum_{i=1}^{D-4} \left| 100 (x_{i+4} - x_i^2)^2 + (x_i - 1)^2 \right| \quad (5)$$

In our experiment, the number of dimensions is set to 30 which is a common number in the area of numerical optimization.

In order to evaluate and compare the strategies using the benchmark functions, two measures are defined including *Success Rate (SR)* and *Convergence Ratio (CR)*. The former measure refers to the percentage of runs which find the optimal solution within one experiment (100 independent runs). An experiment with the 100% SR is referred as a reliable experiment. The later measure deals with the average number of generations required to find the optimal solution. Within the conducted experiments, the experiment which has the minimum average is considered to be the base of the CR calculation, which is as follows.

$$ConvergenceRatio = \frac{AVG - AVG_{min}}{AVG_{max} - AVG_{min}} \times 100\%$$

where AVG refers to the average number of generations required to find the optimal solution for an experiment, and AVG_{min} and AVG_{max} denote the minimum and the maximum averages obtained over all experiments with respect to each optimization function. Therefore, the zero CR value represents the experiment which requires the minimum average number of generations to find the optimal solution.

Each dimensions division strategy is evaluated with different number of local CAs. The sequential, jumping, and

Table 3: The CR and SR values with respect to each experiments in percentile form alongside their average in the last two rows.

Style		Balanced Strategies										Imbalanced Strategies							
		Seq				Jump				Seq/Jump	Over Seq	Log					Cust		
Local CAs		5	7	10	15	5	7	10	15	30	15	5	7	10	15	30	59	35	
f_1	CR	25.08	18.48	8.62	4.25	26.30	18.13	8.60	4.29	0.50	9.29	100.00	49.14	42.40	11.69	3.61	0.00	0.42	
	SR	99	100	100	100	100	100	100	100	100	100	38	95	91	100	100	100	100	
f_2	CR	24.23	11.90	11.17	24.59	-	-	-	100.00	99.55	7.85	52.78	31.43	22.43	7.70	1.90	0.00	1.20	
	SR	95	100	100	100	0	0	0	6	9	100	33	72	73	100	100	100	100	
f_{2M}	CR	-	-	-	-	-	-	-	-	-	-	90.18	84.22	100.00	59.62	18.52	7.43	0.00	
	SR	0	0	0	0	0	0	0	0	0	0	33	52	48	100	100	100	100	
f_3	CR	26.98	16.84	7.82	3.62	27.61	16.41	7.89	3.63	0.20	8.75	100.00	54.49	44.31	11.31	3.33	0.00	0.36	
	SR	100	100	100	100	100	100	100	100	100	100	20	100	94	100	100	100	100	
f_4	CR	25.02	15.36	5.89	2.34	32.12	16.22	5.98	2.35	0.00	7.01	100.00	48.93	48.94	9.51	1.84	0.54	0.15	
	SR	100	99	100	100	100	100	100	100	100	100	49	90	46	99	100	100	100	
f_5	CR	32.83	20.98	10.13	5.20	33.41	19.98	10.41	5.13	0.61	11.26	100.00	61.21	53.42	14.79	4.24	0.00	0.58	
	SR	100	100	100	100	100	100	100	100	100	100	33	96	89	100	100	100	100	
f_6	CR	31.96	18.03	7.16	2.98	24.36	15.19	6.67	2.91	0.00	7.12	100.00	42.18	38.73	9.21	2.28	0.06	0.29	
	SR	100	100	100	100	100	100	100	100	100	97	54	98	85	100	100	100	100	
f_7	CR	27.65	17.49	8.63	4.36	25.88	18.08	8.47	4.35	0.48	9.40	100.00	46.73	49.34	11.83	3.54	0.00	0.45	
	SR	100	100	100	100	100	100	100	100	100	100	31	94	93	100	100	100	100	
f_8	CR	31.81	19.73	9.78	5.05	32.23	21.14	9.88	4.97	0.59	10.91	100.00	52.55	53.56	13.63	4.16	0.00	0.56	
	SR	100	100	100	100	100	100	100	100	100	100	21	92	86	100	100	100	100	
f_9	CR	35.08	24.62	10.12	4.25	32.08	20.64	8.99	4.29	0.00	10.22	100.00	51.90	51.95	13.09	3.28	0.03	0.38	
	SR	99	100	100	100	100	100	100	100	100	100	39	92	88	100	100	100	100	
f_{10}	CR	33.17	20.85	5.92	2.43	31.70	20.52	5.95	2.41	2.52	9.03	100.00	53.89	53.92	10.12	1.68	0.00	2.15	
	SR	93	97	99	100	97	99	98	100	100	100	2	71	68	95	100	100	100	
f_{11}	CR	38.11	24.82	13.27	6.66	41.96	25.83	13.25	6.73	0.74	14.72	100.00	60.93	59.53	19.06	5.75	0.00	0.73	
	SR	100	100	100	100	99	100	100	100	100	100	16	94	85	100	100	100	100	
f_{12}	CR	12.98	6.53	3.15	1.29	12.49	7.50	3.21	1.27	0.00	3.86	100.00	19.33	23.25	5.15	1.35	0.41	0.10	
	SR	100	100	100	100	100	100	100	100	100	100	69	100	100	100	100	100	100	
Average	CR	28.74	17.97	8.47	5.58	29.10	18.15	8.12	11.86	8.77	9.12	95.61	50.53	49.37	15.13	4.27	0.65	0.57	
	SR	91.2	92.0	92.2	92.3	84.3	84.5	84.5	85.1	85.3	92.1	33.7	88.2	80.5	99.5	100.0	100.0	100.0	

logarithmic strategies are experimented by 5, 7, 10, 15, and 30 local CAs. Additionally, the logarithmic strategy with 59 local CAs, overlapped sequential with 15 local CAs, and customized logarithmic with 35 local CAs are evaluated. It should be mentioned here that the sequential and jumping strategies with 30 local CAs are equivalent. Therefore, there are 17 different experiments conducted in this study.

The detail results of all experiments are presented in Table 3 where all the numbers are in percentile form. In this table the Seq, Jump, Log, Over Seq, and Cust refer to the sequential, jumping, logarithmic, overlapped sequential, and customized logarithmic techniques, respectively. For each optimization function, the first row represents CR value, followed by SR value in the second row. The average CR and SR value with respect to the different experiments are presented in the last two rows. The zero CR values emphasizes with bold face illustrate that the logarithmic strategy with 59 sub-populations offers the zero CR for 8 functions, the sequential and jumping strategies with 30 sub-populations offers the zero CR for 4 other functions and the zero CR for the last one which is function f_{2M} is offered by the customized logarithmic strategy with 35 local CAs.

It should be mentioned here that CR value for each experiment is calculated based on the runs where an optimal solution is found. For instance, the sequential strategy with

30 local CAs can find an optimal solution in only 9 independent runs out of 100, meaning that its CR value is calculated only based on those 9 successful runs. Consequently, for the experiments where no optimal solution is found there is no CR value, such as the jumping strategy with 10 local CAs applied on function f_2 .

Table 3 illustrates that the sequential strategy works nearly equivalently to the jumping strategy with the exception of function f_2 , where the difference value of the adjacent dimensions is a key point. Therefore, for this function the strategies assigning adjacent dimensions to the same local CA work better. In order to prove this claim, function f_{2M} is proposed and the results show that if this key point is removed the sequential strategy works the same as the jumping one. The results also show that the logarithmic strategy works well only with higher number of sub-populations.

Generally, the only configurations that offer the SR of 100% for all 13 optimization functions include: (1) the customized logarithmic strategy with 35 local CAs, (2) the logarithmic strategy with 59 local CAs, and (3) the logarithmic strategy with 30 local CAs. Within these three configurations, the first two offer average CR values less than 1.00% which is much lower than 4.27% of the third one. This is due to the fact that the third one does not include the assignment of each dimension to one local CA.

Table 4: The average SR and the number of experiments with the CR of 100% obtained for each optimization function.

Functions	f_1	f_2	f_{2M}	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
Average SR	95.5%	64.0%	31.4%	94.9%	93.1%	95.2%	96.1%	95.2%	94.1%	95.2%	89.4%	93.8%	98.2%
Reliable Experiments	13	8	4	15	12	14	13	14	14	13	7	13	16

In fact assigning only one dimension to a local CA helps to find an optimal solution when the dimensions of the problem are independent to each other, and assigning a number of dimensions to a local CA is more useful where the dimensions are inter-dependent. Consequently, incorporating a hybrid mechanism works better when there is no prior knowledge about the given optimization function. Therefore, generally the imbalanced dimension division techniques outperform the balanced strategies.

The overall results are represented in Table 4 illustrating the average SR obtained by all the configurations for each optimization function and the number of reliable experiments capable to obtain the SR of 100%. This information determines the most challenging optimization functions which are f_{2M} , f_{10} , and f_2 . For the rest of the functions, more than 12 experiments out of 17 are able to find the optimal solutions in all of their 100 independent runs.

Conclusions

HMP-CA is a new architecture for MP-CA which incorporates heterogeneous local CAs with a shared belief space. In HMP-CA, local CAs are working to optimize different subsets of dimensions. Therefore, the mechanism that divides the dimensions among local CAs plays a key role in the performance of the method. In this article, a study is conducted to analyze different dimension division strategies.

The evaluated strategies are classified into two categories; namely balanced and imbalanced. Within the former category, sequential, jumping, and overlapped sequential strategies are taken into account in addition to the logarithmic and customized logarithmic strategies from the latter category.

The results show that generally the imbalanced techniques offer better performance compared to the balanced schemes. More specifically, the logarithmic strategy with 59 local CAs and the customized logarithmic technique with 35 local CAs present the best success rates as well as the minimum convergence ratios.

Future work may include exploring additional dimension division schemes to discover even better success rates and convergence ratios. The proposed algorithms may also be tested in the future against other optimization functions, particularly on functions with inter-dependent variables.

Acknowledgment

This work is made possible by a grant from the National Science Foundation and NSERC Discovery No. 327482.

References

Digalakis, J., and Margaritis, K. 2002. A multipopulation cultural algorithm for the electrical generator schedul-

ing problem. *Mathematics and Computers in Simulation* 60(3-5):293–301.

Guo, Y.-N.; Cheng, J.; Cao, Y.-Y.; and Lin, Y. 2011. A novel multi-population cultural algorithm adopting knowledge migration. *Soft Computing* 15(5):897–905.

Lin, C.-J.; Weng, C.-C.; Lee, C.-L.; and Lee, C.-Y. 2009. Using an efficient hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy network design. In *International Conference on Machine Learning and Cybernetics*, 3076–3082.

Lin, C.-J.; Chen, C.-H.; and Lin, C.-T. 2008. Efficient self-evolving evolutionary learning for neurofuzzy inference systems. *IEEE Transactions on Fuzzy Systems* 16(6):1476–1490.

Lin, C.-J.; Chen, C.-H.; and Lin, C.-T. 2009. A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 39(1):55–68.

Mezura-Montes, E.; Velazquez-Reyes, J.; and Coello Coello, C. A. 2006. A comparative study of differential evolution variants for global optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, 485–492.

Raeesi N., M. R., and Kobti, Z. 2012. A knowledge-migration-based multi-population cultural algorithm to solve job shop scheduling. In *The 25th Florida Artificial Intelligence Research Society Conference (FLAIRS-25)*, 68–73.

Raeesi N., M. R., and Kobti, Z. 2013. Heterogeneous multi-population cultural algorithm. In *IEEE Congress on Evolutionary Computation (CEC)*, 292–299.

Reynolds, R. G. 1994. An introduction to cultural algorithms. In Sebald, A. V., and Fogel, L. J., eds., *Thirs Annual Conference on Evolutionary Programming*, 131–139. River Edge, New Jersey: World Scientific.

Xu, W.; Zhang, L.; and Gu, X. 2010. A novel cultural algorithm and its application to the constrained optimization in ammonia synthesis. *Communications in Computer and Information Science* 98(1):52–58.

Yu, W.-j., and Zhang, J. 2011. Multi-population differential evolution with adaptive parameter control for global optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, 1093–1098.