2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Compensate Information from Multimodal Dynamic Landscapes: An Anti-Pathology Cooperative Coevolutionary Algorithm

Xingguang Peng, *IEEE Member*, Xiaokang Lei and Kun Liu

***Abstract*— Cooperative coevolutionary algorithms (CCEAs) divides a problem into several components and optimizes them independently. Some coevolutionary information will be lost due to the search space separation. This may lead some algorithmic pathologies, such as relative overgeneralization. In addition, according to the interactive nature of the CCEA, the coevolutionary landscapes are dynamic. In this paper, a multi-population strategy is proposed to simultaneously search local or global optima in each dynamic landscape and provide them to the other components. Besides, a grid-based archive scheme is proposed to archive these historic collaborators for reasonable fitness evaluation. Two benchmark problems were used to test and compare the proposed algorithm to three classical CCEAs. Experimental results show that the proposed algorithm effectively counteract relative overgeneralization pathology and significantly improve the rate of converging to global optimum.**

I. INTRODUCTION

COOPERATIVE coevolutionary algorithms (CCEAs) divide a problem into some small components and optimizes them independently. These components are optimized by coevolutionary populations and the fitness evaluation of individuals of the coevolutionary populations is realized through cooperatively interactions. The divide-and-conquer nature of CCEAs implies an promising way to solve distributed optimization problem such as decision-making [1], planning [2] and large scale optimization [3].

In spite of this, simple CCEAs will lose a great amount of information in the process of partitioning the whole search space into separate component search spaces. A sub-population doesn't have access to the landscapes of the other sub-populations due to the search space partition. The fitness evaluation of a sub-population depends on the collaborators sent from the other coevolutionary sub-populations. Since the collaborators (usually the best individuals of the sub-populations) can not sufficiently represent the profile of the landscapes of the other sub-problems, the fitness evaluation of a CCEA always suffers from information loss in contrast to the traditional EA. As a result, simple CCEAs may not just get caught in but gravitate towards suboptimal solutions represented by Nash equilibrium in the joint search space [4]. Several pathologies have been reported and analyzed in the [5, 6].

This work was supported by the National Nature Science Foundation of China under Grant 61105068 and the NPU Foundation for Fundamental Research under Grant JCY20130110.

The authors are with the School of Marine Science and Technology, Northwestern Polytechnical University Xi'an 710072, China. Corresponding author: Xingguang Peng (email: pxg0510@gmail.com).

In order to ensure CCEAs to optimize globally, scientists have made some extensions to the simple CCEA in the literature. Archive is the most popular way to compensate information. Those archives can be classified into two categories: explicit archives and implicit archives. The explicit archives directly store information in specified manners [4, 7-9]. In contrast, the implicit archive strategy does not necessary have an additional archive. The useful information is just stored in the populations or in the genotypes [10].

In addition to the archive method, another effective way is to learn from the evolution process. In [11] the biasing information is learned by estimating the best possible reward for an individual if partnered with its optimal collaborator. The authors of [12] argued that coevolutionary populations should not necessarily explore only their most promising solutions, but also those solutions that provide the other coevolutionary populations with accurate projections of the joint search space. This means that it is also important to learn collaborating information for the other coevolutionary populations.

In this work, the representative solutions of a component's landscape are searched and sent to the other components for their construction of accurate joint search spaces. To this end, each coevolutionary population serves a multimodal optimization to concurrently search several optima (global or local) of the component. These optima are seen as representatives and are sent to the other populations. In addition, the landscapes of populations are indeed dynamic because the fitness evaluation of a population depends on the collaborating information provided by the others and such information may change while the coevolution procedure proceeds.

Therefore, the main task of this work is to extend the behavior of coevolutionary populations from conventional search to multimodal search in dynamic landscapes. The interaction between two coevolutionary populations will not be the random or best-of-N individuals but a set of current global or local optima which may contain more useful coevolutionary information. Besides, an archive scheme will be studied in this work to further compensate information by maintaining the interaction information from the other components.

II. PROPOSED ALGORITHM

Over the past two decades, many algorithms have been proposed to enhance conventional EAs to work in dynamic landscapes. Comprehensive surveys can be found in [13, 14]. The main idea of these algorithms is to persistently compensate population diversity for dynamic optimization, which could also be seen as a kind of information compensation. However, some of these algorithms were designed to efficiently work for dynamic optimization problems with special property. For example, memory based methods [15] are designed to persistently optimize for cyclic problem; Prediction based methods [16] can only work efficiently in predictable dynamic optimization problems. As for the CCEAs, the property of the dynamic landscapes of the coevolutionary populations is hard to be known in priori due to the random characteristic of evolution process. Thus, in this work, we need a kind of dynamic EA which can concurrently search multiple optima in dynamic landscapes without any knowledge about the problems. To this end, we borrow the idea of multi-population based dynamic EAs to improve the coevolutionary populations' ability of searching representative information. Multi-optima dynamic search just enhances the algorithm from interaction-sending aspect. As for interaction-receiving aspect, we propose an archive scheme to utilize the interactive information. Moreover, this archive scheme is benefit for simplifying the multi-optima dynamic search.

A. The Multi-population Strategy for Dynamic Landscapes

The problem-independent and multi-optima-tracking properties of multi-population based dynamic EAs have attracted researchers to develop effective algorithms [17, 18] in recent years. In this work, we modify Self-Organizing Scout (SOS) algorithm [19] and integrate it into the conventional CCEA. The main reason why we choose SOS is that it is a relatively early and well-known algorithm developed from genetic algorithm. The reason why we have to modify the origin SOS algorithm is that the landscape in SOS is different from that of the proposed multi-population strategy. In the original SOS, the landscape changes randomly. At one site in the landscape, its height may not only go up but also go down arbitrarily. In this work, we will design a scheme (as will be described in the Section B) to maintain and utilize interactive information for fitness evaluation. This special scheme can maintain a monotonous landscape. The framework of multi-population strategy is given in Fig. 1. In general, the multi-population strategy works as follows:

1) Compute next generation of base and child populations: Consider a coevolutionary population, in our multi-population strategy, it is divided into one base population P_0 and several child populations $P_i, i = 1, \ldots, N_{co}$. The only base population searches high-value peaks in the whole search space of the components. When a peak is confirmed, several individuals around it will be cut off to generate a child population and the search space that covers the child population will also be split off from the whole search space. The base and child populations run independently within their search spaces. The offspring of base population must not fall into the search spaces of child populations. The search spaces of child populations can overlap a little. However, when two child spaces overlap to some extent they will merge together.

```
REPEAT
  Compute next generation of the base population and
    child populations;
  Management of child populations;
  Check forking criterions to create new child populations;
  IF (sending interaction is needed)
    Output representative individuals of child populations;
  Recycle individuals abandoned in the above steps;
UNTIL termination criterion;
```

Fig. 1. General framework of multi-population strategy.

2) Forking criteria: The concept of forking population in SOS and this work comes from Forking Genetic Algorithm (FGA) [20]. The forking populations were used to locate more than one static optima in multi-modal optimization problem. We use two forking criteria to judge whether a child population should be split off. Similar to FGA, at each generation, the following two criteria are used to confirm a child population as a forking population.

- The base population has met the stopping criterion, e.g. the best fitness has not increased for a number of generations.
- Second, There have been at least individuals in the area determined by the center (the best individual) and the maximal radius of child populations (denote r_{max}).

If a cluster of individuals are confirmed as forking population, a number of P_{sinit} individuals will be split off from the based population as a new child population and the corresponding searching area will also be cut off from the main search space. Note that, the base population should at least remain a minimal population P_{0min} after a new child population has been split off.

The forking criteria of our multi-population strategy are simple and easy to be implemented in contrast to that of SOS. The original SOS can scout new peaks and track existing peaks in an arbitrary dynamic landscape. The fitness value of the best individual may increase or decrease. Therefore one cannot determine whether the best individual has really changed only by its fitness value. Thus, at the forking generation, SOS has to confirm several clusters of individuals according to the boundary values of number of individuals, range of child search space and relative fitness value of new and existing forking populations. Among these clusters only the one with maximal ratio of number of individuals to diameter is selected to fork.

Comparing with SOS, our multi-population strategy just uses the stopping criterion of ordinary genetic algorithms to judge the converging degree. This is benefited by the interaction utilizing scheme (which will be described in Section B), the landscape changes monotonously rather than arbitrarily in SOS. Accordingly, the criterion such as that the fitness of best fitness has not increased for a number of generations could be effective in our multi-population strategy.

3) Management of child populations: During the run of multi-population strategy, each child population tracks its corresponding best individual (center of the search space) with gradually shrinking search space. Considering the i-th child population P_i, the radius of the search space r_i decreases from maximal radius r_{max} to minimal radius r_{min} with respect to the generation number g_i of the child population and a shrinking factor f_r like follows [19]:

$$r_i(t+1) = r_{min} + (r_i(t) - r_{min})^{\frac{g_i}{-fr}} \tag{1}$$

Each time, when r_i decreases some individuals may locate outside the new search space. These individuals will be discarded and stored in a recyclable individuals archive Ψ subjected to the minimal population size P_{smin}.

The center c_i of the search space is the best individual of P_i. During the searching procedure of P_i the c_i may change its location, which will lead the movement of P_i's search space. This may cause overlapping of different search spaces. Usually this is allowed, only when the center of a search space falls in the search space of another child population, then both child populations are merged.

When merging two child populations P_1 and P_2, the child population with smaller fitness (assume P_2) is integrated into the other one (P_1). The search space of P_1 is enlarged to enclose some individuals in P_2. The radius of P_1 is calculated as follow with subject to the maximal radius r_{max}.

$$r_1 = min\left(\sqrt[2]{r_1^2 + r_2^2}, r_{max}\right) \tag{2}$$

The individuals of P_2 that fall in the enlarged area of P_1 are remained and the else are discarded and stored in the archive Ψ. Note that, the size of P_1 should be no more than P_{smax}, if the size of P_1 had reached its upper limit, the individuals are also moved into the archive Ψ although they locate in the updated area of P_1 .

4) Retrieve the individuals in recyclable archive: It can be seen in the above, individuals may be moved into the recyclable archive Ψ when child populations moving, shrinking and merging. Ψ is just a temporary archive for the discarded individuals. The archived individuals are regenerated randomly and retrieved into the base population at the end of an evolutionary cycle. Note that, the regenerated individuals are not allowed locating in the search spaces of child populations.

B. Grid-based Interactive Information Utilizing Scheme

In this work, we propose a grid-based archive scheme. As shown in Fig. 2, for decision variable x_i, the search space is divided into N_a segments and each segment is assigned an archive element $e_i^\alpha, \alpha = 1, \ldots, N_e$. Each e_i^α stores collaborative individuals provided by the other components. Besides, each e_i^α is stored together with an index value $\hat{x}_i^\alpha$ which performs the best in the α-th segment of i-th component when combining with e_i^α. In other words, the complete solution $\{\hat{x}_i^\alpha \bigcup e_i^\alpha\}$ has the best fitness value in the α-th segment of i-th component. When evaluating the fitness of individual $I_k \in P_i$, the result is the better one between $\{\hat{x}_i^\alpha \bigcup e_i^\alpha\}$ and $\{I_k \bigcup e_i^\alpha\}$ (assume that I_k locates in the α-th segment). In addition, the archive elements are updated during the run to provide better collaborator for fitness evaluation. Therefore, the landscape of each component is monotonously dynamic.

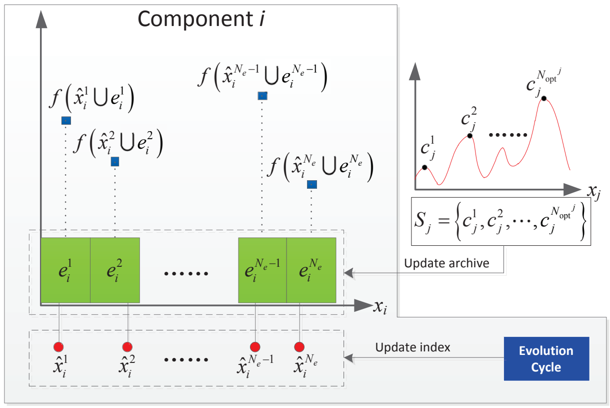

Fig. 2. Demonstration of grid-based archive scheme.

According to the description given above, an archive element is composed of a received collaborative individual from the other components and a local current-best index value. To update the archive, it is not only the received interactions but also the evolution state of decision variable should be considered.

Assume j-th component ($j \neq i$) sends its current global or local optima $\{c_j^1, \ldots, c_j^{N_{opt}^j}\}$ to the j-th component, N_{opt}^j denotes the quantity of optima found in j-th component. The pseudo-code of grid-based archive scheme is given in Fig. 3 (considering a maximum optimization problem).

If interaction receiving generation **do**
  for each archive element e_i^α, compare it to each
    $c_j^l (l = 1, \ldots, N_{opt}^j)$
    if $f\{\hat{x}_i^\alpha \bigcup c_j^l\} > f\{\hat{x}_i^\alpha \bigcup e_i^\alpha\}$ **do**
      $e_i^\alpha = c_j^l$
    end
else for each individual $I_k \in P_i$ **do**
  confirm segment index α for I_k
  if $e_i^\alpha = NULL$ **do**
    e_i^α equals to the best individual of component j
    $\hat{x}_i^\alpha = I_k$
  else if $f\{I_k \bigcup e_i^\alpha\} > f\{\hat{x}_i^\alpha \bigcup e_i^\alpha\}$ **do**
    $\hat{x}_i^k = I_k$
  end
end

Fig. 3. Pseudo-code of grid-based archive scheme.

C. Implementation

The flowchart of the proposed CCEA (termed mCCEA in the following context) in a certain component is illustrated in Fig. 4. At each generation, the multi-population strategy is executed for computing the next generation of both base and child populations, managing child populations and checking for creating new child populations. Besides, the archive index values are also updated at each generation. When new interaction is received, the archive elements which are used for evaluating populations are updated according to the index values and interaction. At every N_δ generations, the centers of current child populations are sent to the other components for cooperating their evolution. The algorithm terminates when stopping criteria met and each component outputs its best-performing archive (including the components best-performing index value and the corresponding collaborator from the other components).

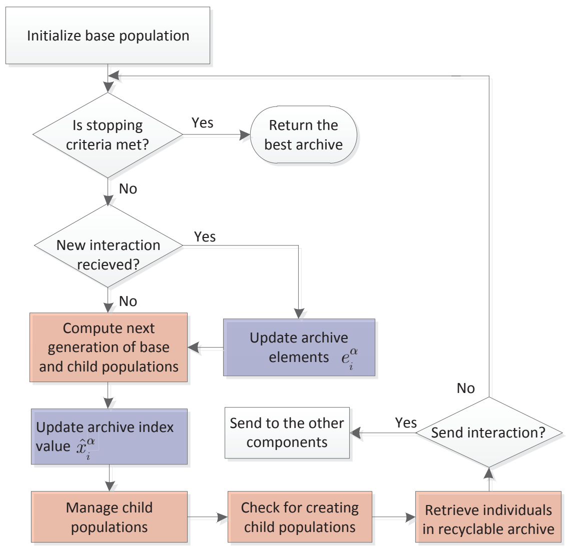

Fig. 4. Flowchart of proposed algorithm.

For the sake of infinite Nash equilibriums problems, when constructing an executing solution for the objective problem from the output of each component, a comparing logic has to be applied as follows. Firstly, compute the fitness of two kinds of complete solutions, one is the combination of the best local index value and the corresponding archived collaborator of each component, the other one is the combination of the best local index values of each component. Then, among these combinations (complete solutions) choose the one with the best fitness value as the final executing solution.

III. Experimental Results and Analysis

Relative overgeneralization is a pathological coevolution behavior that occurs when populations in the system are attracted towards areas of the space in which there are many strategies that perform well relative to the interacting partner [6]. In this work, we compare our proposed algorithm with several other typical CCEAs on benchmark problems that are designed for testing relative overgeneralization pathology.

A. Algorithms for Comparison

To validate the effectiveness of mCCEA, the following CCEAs are used for comparision:

1) Traditional CCEA (tCCEA): tCCEA evaluates the fitness of an individual as the maximum when partnered with collaborators from the teammates population: chosen at random, plus the fittest individual from the teammates previous-generation population. In the following experiments is set to 4.

2) Biased CCEA (bCCEA): bCCEA [11] is developed from tCCEA. The fitness will be based partly on the maximum of collaborations with randomly chosen collaborators, and partly on the reward obtained when collaborating with the optimal collaborator from the other components. For fair comparison, the is set to 3. The biasing rate is set to 0.5.

3) Complete CCEA (cCCEA): cCCEA is an extreme term of tCCEA and was used as a peer algorithm in [12]. When evaluating the fitness of an individual, a component accesses all individuals of the other components to obtain the maximum-of-all evaluation.

B. Benchmark Problems

We will test these algorithms using a class of problem domains called the maximum of two quadratics (or MTQ). These problems include a global optimum and a local suboptimum, where the suboptimum covers a much wider range of the search space and is thus difficult to escape. The problems have been used before by [10-12, 21]. The joint reward function for the MTQ class is defined as:

$$MTQ(x,y) \leftarrow \max \begin{cases} H_1 * \left(1 - \frac{16*(x-X_1)^2}{S_1} - \frac{16*(y-Y_1)^2}{S_1}\right) \\ H_2 * \left(1 - \frac{16*(x-X_2)^2}{S_2} - \frac{16*(y-Y_2)^2}{S_2}\right) \end{cases} \quad (3)$$

where x and y may take values (actions) ranging between 0 and 1. Different settings for H_1, H_2, X_1, Y_1, X_2, Y_2, S_1, and S_2 affect the difficulty of the problem domain in one of the following aspects. H_1 and H_2 affect the heights of the two peaks: higher peaks may increase the chance that the algorithm converges there. S_1 and S_2 affect the area that the two peaks cover: a higher value for one of them results in a wider coverage of the specific peak. This makes it more probable that the coevolutionary search algorithm will converge to this peak, even though it may be suboptimal. Different values for X_1, Y_1, X_2, and Y_2 result in changes in the locations of the centers of the two quadratics, which also affect the relatedness of the two peaks: similar values of the x or y coordinates for the two centers imply higher overlaps of the projections along one or both axes.

In the following experiments, we use the following parameter settings in [10]: $H_1 = 50$, $H_2 = 150$, $X_1 = 0.75$, $Y_1 = 0.75$, $X_2 = 0.25$, $Y_2 = 0.25$, $S_1 = 1.6$, $S_2 = 0.03125$. We will denote this function MTQ50. We will also consider a second function MTQ125 which is MTQ50 but with $H_1 = 125$.

C. Experimental Parameter Settings

In the following experiments, all algorithms are realized based on GA toolbox [22] with almost the default settings: the population size of each component (i.e. the sum of the size of the base population and child populations) is 50, tournament selection size is 2, simulated-binary crossover and mutation rates are 0.8 and 0.1 respectively, the maximal generation number is 1000. All algorithms are run 50 times independently. The stopping criteria used in fork checking is that the best fitness of the base population keeps the same for more than 5 generations.

The following parameters related to multi-population strategy are the same to that of SOS or scale down according to the variable domain of MTQ: the maximal, minimal and initial size of child populations (i.e. P_{smax}, $P_{smin}$and P_{sinit}) are 20, 4 and 10 respectively. The minimal size of the base population P_{0min} is 10. The maximal and minimal radius of child populations (i.e. r_{max} and r_{min}) are 0.2 and 0.1 respectively. Shrinking factor f_r is 2.0. In addition, the quantity of segments of grid-based archive (i.e. N_e) is 10 and the interaction interval N_e is 10 generations.

D. Comparison on Rate of Converging to Global Optimum

The results of the rate of converging to global optimum (the Euclidean distance from the resulting best solution to the global optimum is less than 0.01) are shown in Table I.

TABLE I
RATE OF CONVERGING TO GLOBAL OPTIMUM

Problems	mCCEA	tCCEA	bCCEA	cCCEA
MTQ50	98%	8%	74%	68%
MTQ150	98%	0%	44%	44%

It can be seen that tCCEA suffers from relative overgeneralization seriously. bCCEA, benefits from heuristic estimation of the best collaborator, greatly improves the converging rate. However, it is difficult for bCCEA to exactly estimate the best collaborator and adaptively adjust the biasing rate, which is an obstacle to further improve its performance. As for the cCCEA, it is essentially the same to tCCEA because it also uses the maximum-of-N collaborating scheme. Without any memory or archive that maintains the up-to-date optimal collaborator like bCCEA, the fitness of an individual depends on the combination to the current collaborators of the other components. The main reason that cCCEA performances better than tCCEA is the great amount of information exchange (interacting the whole population).

mCCEA nearly successes to converge to the global optimum in all tests and greatly outperforms the other three algorithms. This is the result of information compensation which is achieved by the multi-population strategy and grid-based archive scheme. The former help the algorithm search representative individuals rather than some randomly chosen ones in the dynamic landscape. These representative individuals (local or global optima) contain more information of the state of a component and provide more useful information to the other components. The later aims to better utilize these representative information. The up-to-date optimal collaborators are stored in the segments of grid-based archive according to local index values. Since for each segment of the search space of a component there is an up-to-date optimal collaborator stored in the archive, more historic coevolutionary information is maintained for coevolution.

The analysis above can also be verified by the dynamic performance of these algorithms in decision and objective spaces. Fig. 5 and Fig. 6 show the comparison of best fitness value curves (average curves over 50 runs) against generation numbers. It can be seen that the tCCEA and cCCEA suffers from the relative overgeneralization, there fitness curves go flat at incorrect levels (suboptimal solution). bCCEA needs a long period to learn the estimation of optimal collaborator. In contrast, mCCEA can evolve to the global optimum in a short period due to the effective information compensation.

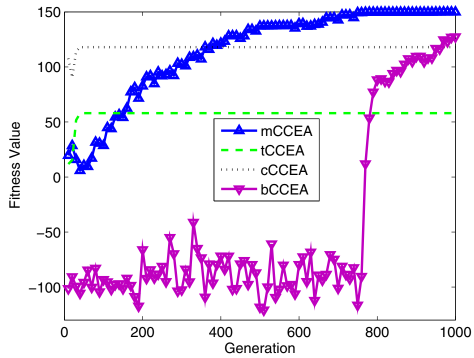

Fig. 5. Average curves of best fitness for MTQ50.

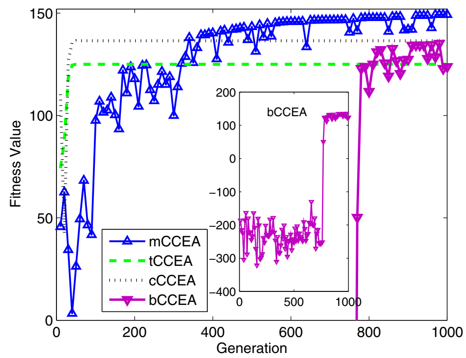

Fig. 6. Average curves of best fitness for MTQ125.

The similar conclusions could be drawn with respect to the decision space. Fig. 7 and Fig. 8 show the average trajectories of the best solutions. These trajectories go along the direction from suboptimum to global optimum because suboptimum has a larger attractive basin comparing with that of global optimum. It can be seen that only the best solution of mCCEA has enough power to reach the global optimum. This power comes from the continuous and effective information compensation provided by the proposed multi-population strategy and grid-based archive scheme. As for bCCEA, the right direction is just found after a long and disordered learning period. As for cCCEA and tCCEA, the best solutions poor moving power is due to the lack of effective information compensation.

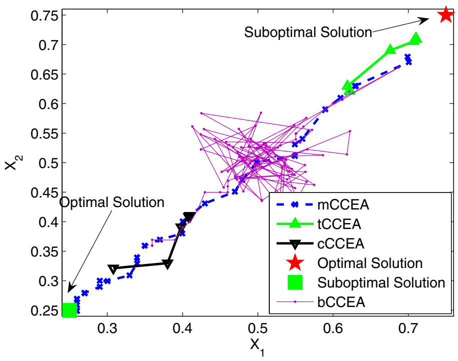

Fig. 7. Average trajectories of best solutions for MTQ50.

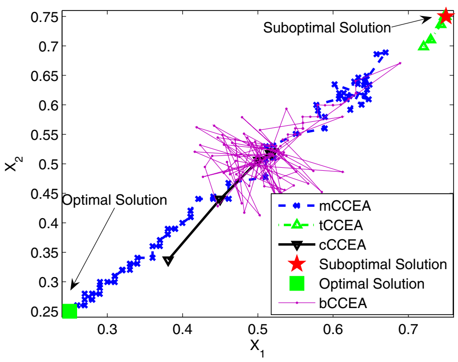

Fig. 8. Average trajectories of best solutions for MTQ125.

E. Validation of Executing Solution Construction on Infinite Nash Equilibriums Problems

As mentioned in Section II-C, when mCCEA terminates each component may supply more than one optimal collaborators for constructing the final executing solution. A solution construction method has been proposed for the sake of infinite Nash equilibriums problems. The following experiments are designed to validate the solution construction method. All of the experimental and algorithmic parameters are the same to the above expecting that we will use a second class of problems with infinite Nash equilibriums.

This second class problems are developed from the MTQ problem and are termed SMTQ. They are defined as [12]:

$$MTQ(x,y) \leftarrow \max \begin{cases} H_1 * \left(1 - \frac{16*(x_1^r - X_1)^2}{S_1} - \frac{12*(y_1^r - Y_1)^2}{S_1}\right) \\ H_2 * \left(1 - \frac{16*(x_2^r - X_2)^2}{S_2} - \frac{12*(y_2^r - Y_2)^2}{S_2}\right) \end{cases} \quad (4)$$

where x_1^r, x_2^r, y_1^r and y_2^r are the original x and y values (which ranged between 0 and 1) rotated around the centers of the two peaks by $\frac{\pi}{4}$:

$$\begin{cases} x_1^r = (x - X_1) * \cos\frac{\pi}{4} + (y - Y_1) * \sin\frac{\pi}{4} + X_1 \\ y_1^r = (x - X_1) * \cos\frac{\pi}{4} + (y - Y_1) * \sin\frac{\pi}{4} + Y_1 \\ x_2^r = (x - X_2) * \cos\frac{\pi}{4} + (y - Y_2) * \sin\frac{\pi}{4} + X_2 \\ y_2^r = (x - X_2) * \cos\frac{\pi}{4} + (y - Y_2) * \sin\frac{\pi}{4} + Y_2 \end{cases} \quad (5)$$

Observe that the two Nash equilibria from the MTQ class have now become an infinity of Nash equilibria in the SMTQ class. This creates an additional difficulty for the coevolutionary search. Especially for CCEAs like the mCCEA, each component may archive several best component solutions, improper combinations of component solutions may lead poor-performance executing solution. This is because each component solution is searched with an archived collaborator. When combining the component solutions, one doesnt have to combine with another solution which is similar to its collaborator. This may result in improper combination of executing solution whose fitness is rather low.

Fig. 9 and Fig. 10 show the comparison of mCCEA and mdCCEA (mCCEA but doesnt uses the comparing logic in executing solution construction but directly combines the best archive solutions of each component) on SMTQ problems (the parameters are the same to that of MTQ).

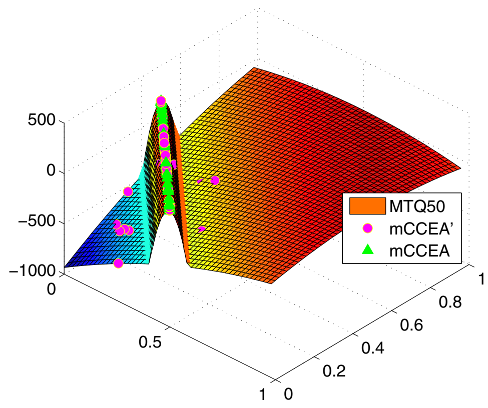

Fig. 9. Executing solutions of mCCEA and mdCCEA for SMTQ50.

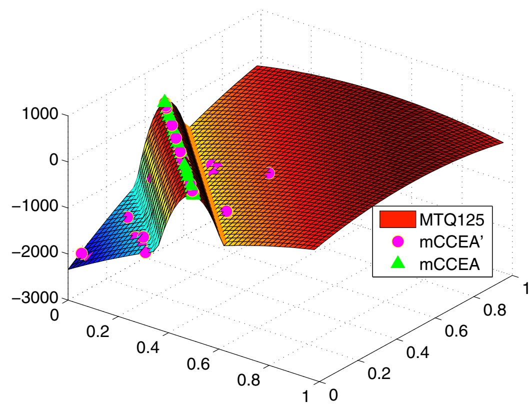

Fig. 10. Executing solutions of mCCEA and mdCCEA for SMTQ125.

The experimental result over 50 runs show that the comparing logic used in the construction of executing solution of mCCEA can significantly improve the rate of converging to the global optima comparing with the mdCCEA (100% vs 80% on SMTQ50 and 100% vs 72% on SMTQ125). It can be seen that some executing solutions of mdCCEA locate at the poor-performance area in the overall landscape due to the improper combination of component solutions. In contrast, mCCEA guarantees its executing solutions locating at the highest ridge in the overall landscape, which validate the effectiveness of our executing solution construction method.

IV. Conclusions

In this work, we proposed a multi-population strategy to learn more useful information for compensating information from dynamic landscape. Within this strategy, the main population is divided into a base and several child populations. These populations behave a dynamic multimodal search so as to obtain and interact local or global optima as the representative information. We also proposed a grid-based archive scheme to maintain more historic coevolutionary information by storing up-to-date the optimal collaborator for each segment of search space. Besides, this archive scheme can maintain a monotonously dynamic landscape which is benefit for simplifying the child population forking. Accordingly, we simplified and modified the SOS algorithm as the realization of the multi-population strategy. In addition, a comparing logic was proposed for constructing the executing solutions for the sake of infinite Nash equilibriums.

In the experiments, we validated the effectiveness of our proposed algorithm, i.e mCCEA, by comparing it with another three CCEAs. The comparing results on the rate of converging to global optimum, best fitness curves and the trajectories of the best solution according to MTQ problems show the proposed information compensation methods can greatly improve the anti-pathology ability. Besides, SMTQ problems were used to validate the effectiveness of constructing executing solutions on infinite Nash equilibriums problems.

References

[1] R. Subbu and A.C. Sanderson, "Network-Based Distributed Planning Using Coevolutionary Agents: Architecture and Evaluation," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 2, 2004, pp. 257-269.

[2] J. Berger, J. Happe, C. Gagn and M. Lau, "Co-evolutionary Information Gathering for a Cooperative Unmanned Aerial Vehicle Team," in *proc. 12th International Conference on Information Fusion*, 2009, pp. 347-354.

[3] Z. Yang, J. Zhang, K. Tang and X. Yao, "An adaptive coevolutionary differential evolution algorithm for large-scale optimization," in *proc. 2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 102-109.

[4] L. Panait, S. Luke and J.F. Harrison, "Archive-Based Cooperative Coevolutionary Algorithms," in *proc. GECCO'06*, 2006, pp. 345-352.

[5] E. Popovici, A. Bucci, R.P. Wiegand and E.D. Jong, "Coevolutionary Principles," *Handbook of Natural Computing*, G. Rozenberg, et al., eds., Springer-Verlag, 2012, pp. 987-1033.

[6] R.P. Wiegand, *An Analysis of Cooperative Coevolutionary Algorithms*, Ph.D. dissertation, Department of Computer Science, George Mason University, Fairfax, Virginia, 2003.

[7] S.G. Ficici and J.B. Pollack, "A Game-Theoretic Memory Mechanism for Coevolution," in *proc. GECCO'03*, 2003, pp. 286-297.

[8] E.P. Manning, "Coevolution in a Large Search Space using Resource-limited Nash Memory," in *proc. GECCO'10*, 2010, pp. 999-1006.

[9] M. Shi and H. Wu, "Pareto Cooperative Coevolutionary Genetic Algorithm Using Reference Sharing Collaboration," in *proc. GECCO'09*, 2009, pp. 867-874.

[10] A. Bucci and J.B. Pollack, "On Identifying Global Optima in Cooperative Coevolution," in *proc. GECCO'05*, 2005, pp. 539-544.

[11] L. Panait, S. Luke and R.P. Wiegand, "Biasing Coevolutionary Search for Optimal Multiagent Behaviors," I*EEE Transactions on Evolutionary Computation*, vol. 10, no. 6, 2006, pp. 629-645.

[12] L. Panait and S. Luke, "Selecting Informative Actions Improves Cooperative Multiagent Learning," in *proc. 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006, pp. 760-766.

[13] Y. Jin and J. Branke, "Evolutionary Optimization in Uncertain Environments ł A Survey," *IEEE Transaction on Evolutionary Computation*, vol. 9, no. 3, 2005, pp. 303-317.

[14] T.T. Nguyen, S. Yang and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, no. October, 2012 pp. 1-24.

[15] X. Peng, X. Gao and S. Yang, "Environment identification based memory scheme for estimation of distribution algorithms in dynamic environments," *Soft Computing*, vol. 15, no. 2, 2011, pp. 311-326.

[16] A. Zhou, Y. Jin and Q. Zhang, "A Population Prediction Strategy for Evolutionary Dynamic Multiobjective Optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, 2013, pp. 40 - 53.

[17] D. Parrott and X. Li, "Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation," *IEEE Transaction on Evolutionary Computation*, vol. 10, no. 4, 2006, pp. 440-458.

[18] C. Li and S. Yang, "A General Framework of Multi-Population Methods with Clustering in Undetectable Dynamic Environments," *IEEE Transaction on Evolutionary Computation*, vol. 16, no. 4, 2012, pp. 556-577.

[19] J. Branke, *Evolutionary Optimization in Dynamic Environments*, Kluwer Academic Pub, 2002.

[20] S. Tsutsui and Y. Fujimoto, "Forking Genetic Algorithms GAs with Search Space Division Schemes," *Evolutionary Computation*, vol. 5, no. 1, 1997, pp. 61-80.

[21] L. Panait, "Theoretical Convergence Guarantees for Cooperative Coevolutionary Algorithms," *Evolutionary Computation*, vol. 18, no. 4, 2010, pp. 581-615.

[22] K. Sastry, Single and Multiobjective Genetic Algorithm Toolbox in C++, *IlliGAL Report No. 2007017*, 2007.