Introducing Recombination with Dynamic Linkage Discovery to Particle Swarm Optimization

Ming-chung Jian
Ying-ping Chen

NCLab Report No. NCL-TR-2006004
January 2006

Natural Computing Laboratory (NCLab)
Department of Computer Science
National Chiao Tung University
329 Engineering Building C
1001 Ta Hsueh Road
HsinChu City 300, TAIWAN
`http://nclab.tw/`

Introducing Recombination with Dynamic Linkage Discovery to Particle Swarm Optimization

Ming-chung Jian and Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
{mcjian@gmail.com, ypchen@cs.nctu.edu.tw}

January 31, 2006

Abstract

In this paper, we introduce the recombination operator with the technique of dynamic linkage discovery to particle swarm optimization (PSO) in order to improve the performance of PSO. Dynamic linkage discovery is a costless, effective linkage recognition technique adapting the linkage configuration by utilizing the natural selection without incorporating extra judging criteria irrelevant to the objective function. Furthermore, we employ a specific recombination operator to work with the building blocks identified by dynamic linkage discovery. Numerical experiments are conducted on a set of carefully designed benchmark functions and demonstrate good performance achieved by the proposed methodology.

1 Introduction

Particle swarm optimizer (PSO), introduced by Kennedy and Eberhart in 1995 [1, 2], emulates flocking behavior of birds to solve the optimization problems. The PSO algorithm is conceptually simple and can be implemented in a few lines of codes. In PSO, each potential solution is considered as a particle. All particles have their own fitness values and velocities. The particles fly through the D-dimensional problem space by learning from the historical information of all the particles. In PSO, There exist global and local versions. Instead of learning from the personal best and the best position discovered so far by the whole population as in the global version of PSO, in the local version, each particle's velocity is adjusted according to its own best fitness value and the best position found by other particles within its neighborhood. Focusing on improving the local version of PSO, different neighborhood structures are proposed and discussed in the literature. Moreover, the position and velocity update rules have been modified to enhance the PSO's performance as well.

On the other hand, genetic algorithms (GAs) introduced by John Holland [3, 4], are stochastic, population-based search and optimization algorithms loosely modeled after the paradigms of evolution. Genetic algorithms guide the search through the solution space by using natural selection and genetic operators, such as crossover, mutation, and the like. Furthermore, the GA optimization mechanism is theorized by researchers [3, 4, 5] with building block processing, such as creating, identifying, exchanging, and the like. Building blocks are conceptually non-inferior sub-solutions which are components of the superior complete solutions. The building block hypothesis states that the final solutions to a given optimization problem can be evolved with a continuous process of creating, identifying, and recombining high-quality building blocks. According to the GA's search capability can be greatly improved by identifying building blocks accurately and preventing crossover operation from destroying them [6, 7]. Therefore, linkage identification, the procedure to recognize building blocks, plays an important role in the optimization mechanism of genetic algorithms.

The two optimization techniques are both population-based that have been proven successful in solving a variety of difficult problems. However, both models have strength and weakness. Comparisons between GAs and PSOs can be found in the literature [8, 9] and suggest that a hybrid of these two algorithms may lead to further advances. Hence, a lot of studies on the hybridization of GAs and PSOs have been proposed and examined. Most of these research studies try to incorporate genetic operators into PSO [10, 11]. Moreover, some try to introduce the concept of genetic linkage into the realm of PSO [12]. Based on the similar idea employed by linkage PSO, our work is to introduce recombination working on building blocks to enhance the performance of PSO with the concept of linkage.

Particularly, in this paper, we propose a dynamic linkage discovery technique to effectively detect the building blocks of the objective function. This technique differs from the traditional linkage detection technique in that the evaluation cost is eliminated. The idea is to dynamically adjust the linkage configuration according to the search process and feedback from the environment. Thus, this technique is costless and easy to be integrated into the search algorithm. Our method introduce the linkage concept and the recombination operator to the operation of PSO. The proposed algorithm is designed such that the PSO mechanism facilitates the global search and the recombination operator working on building blocks reinforces the local search.

The paper is organized as follows. Section 2 discusses previous research and gives an overview of both GA and PSO. Section 3 presents the proposed method and how the dynamic linkage discovery technique cooperates with recombination and PSO. Section 4 describes the test problems and the experimental results. Section 5 discusses the results and the future work. Finally, section 6 concludes the study.

2 Related work in the literature

The traditional PSO algorithm, described in [1], consists of a number of particles, representing a possible solution to a numerical problem, moving around in the search space, In an iteration, the velocity of each particle is updated according to the best position encountered by the particle itself and by any of the particles as

$$\vec{v}_i = w\vec{v}_i + \vec{\varphi}_{1i}(\vec{p}_i - \vec{x}_i) + \vec{\varphi}_{2i}(\vec{p}_g - \vec{x}_i) \ ,$$

where w is the inertia weight described in [13] and $\vec{p}_g$ is the best position known for all particles. $\vec{\varphi}_1$ and $\vec{\varphi}_2$are random values different for each particle as well as for each dimension. The velocity update rule with constriction coefficients is proposed in [14]. The position of each particle is also updated in each iteration by adding the velocity vector to the position vector, i.e.,

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_i \ .$$

The particles in this paper have no neighborhood restriction, which means each particle can affect all other particles. In the local version of PSO, The $\vec{p}_g$ has been replaced by $\vec{p}_l$, the best position achieved by a particle within its neighborhood. Focusing on improving the local version of PSO, different neighborhood structures have been proposed and discussed [15, 16, 17]. Furthermore, studies on modifying the rule of updating position and velocity are also conducted [12, 18, 19].

Devicharan and Mohan [12] first computed the elements of linkage matrix based on observation of the results of perturbations performed in some randomly generated particles. These elements of the linkage matrix were used in a modified PSO algorithm in which only strongly linked particle positions were simultaneously updated. Liang et al [18, 19] proposed a learning strategies where each dimension of a particle learned from just on particle's historical best information, while each particle learned from different particles' historical best information for different dimensions.

In order to enhance the performance of PSO by introducing the genetic operators and/or mechanisms, many hybrid GA/PSO algorithms have been proposed and tested on function minimization problems [10, 11, 20, 21]. Løvbjerg et al [10] incorporated a breeding operator into the PSO algorithm, where breeding occurred inline with the standard velocity and position update rules. Robinson et al [20] tested a hybrid which used the GA algorithm to initialize the PSO population and another in which the PSO initialized the GA population. Shi et al [21] proposed two approaches. The main idea of the proposed algorithm was to parallelly integrate PSO and GA. Settles and Soule [11] combined the standard velocity and position update rules of PSO with the concepts of selection, crossover, and mutation from GAs. They employed an additional parameter, the breeding ratio, to determine the proportion of the population which underwent breeding procedure (selection, crossover, and mutation) in the current generation.

Moreover, the importance of learning genetic linkage has long been discussed and recognized in the field of genetic algorithms [3, 6, 5, 7]. Because it is hard, if not impossible, to guarantee the user-designed chromosome representation provides tightly linked building blocks when the problem domain knowledge is unavailable, a variety of genetic linkage learning techniques have been proposed and developed to handle the linkage problem, which refers to the need of good building-block linkage. The issue of learning problem-specific linkages has been addressed in the genetic algorithm literature [22, 23, 24]. Furthermore, some try to introduce the linkage concept to PSO and formulate linkage-sensitive PSO algorithms [12, 18, 19].

Based on the brief literature review, we know that a combination of GA and PSO can produce a very effective search strategy. The linkage recognition is closely related to the genetic operator. Hence, we incorporate the linkage information with a recombination operator to improve the performance of PSO.

3 PSO with Recombination and Dynamic Linkage Discovery

The main purpose in this study is to enhance the PSO's performance by introducing the genetic operator with linkage concept. In order to make good use of linkage information, we design a special recombination operator. In the recombination process, there is a building block pool composed by selected individuals. Every offspring is created by choosing and recombining building blocks from the pool at random. We use this recombination process to generate the whole next population. An illustration of how a new individual is generated is shown as Figure 1. A seemingly similar operator has been proposed by Smith and Fogarty [25]. In [25], the representation on which the recombination operator works takes the form of markers on the chromosome which specify whether or not a gene is linked to its neighbors. Different chromosomes form different numbers of building blocks. However, our recombination operator keeps a global linkage configuration such that every individual in the pool is decomposed into the same building blocks.

In the present work, we assume that the relation between different dimensions is dynamically changed along with the search process. Thus, the linkage configuration should be updated accordingly. Instead of incorporating extra artificial criteria for linkage adaptation, we, again, entrust the task to the mechanism of natural selection. As a consequence, we propose the dynamic linkage discovery technique and we call the PSO combined with recombination and

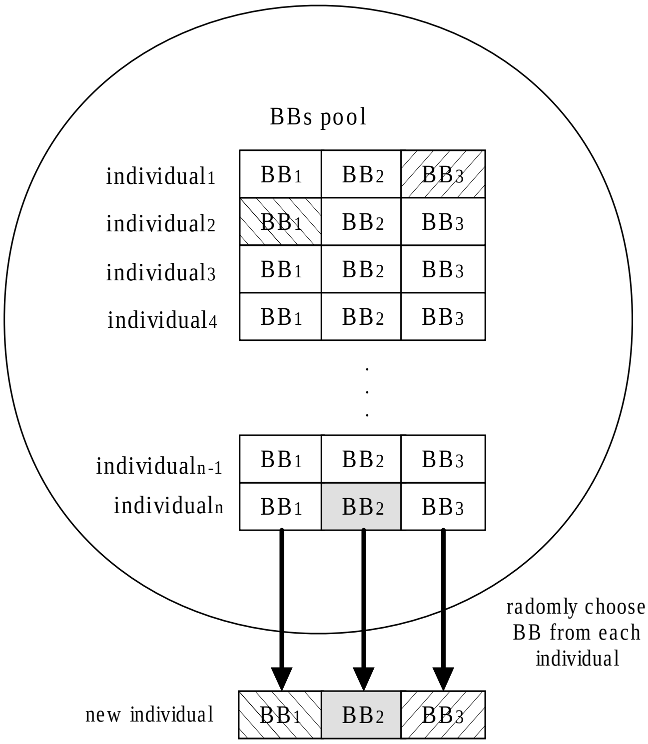

Figure 1: The procedure of how a new particle is generated through the recombination operator

dynamic linkage discovery as PSO-RDL. The dynamic linkage discovery technique is costless, effective, and easy to implement. The idea is to update the linkage configuration according to the fitness feedback. During the whole search process, PSO-RDL first assigns a set of random linkage groups and then adjusts the linkage groups according to the fitness feedback from the optimization problem. If the best fitness value of the current population is improved over a specified threshold, the current linkage configuration is considered appropriate and stays unchanged. Otherwise, the linkage groups will be reassigned at random.

In the proposed algorithm, we repeat the PSO procedure for a certain number of generations, we term such a period a *PSO epoch* in the rest of this paper. After each PSO epoch, we select the N best particles from the population to construct the building block pool and conduct a recombination operation according to the building blocks identified by dynamic linkage discovery. After the recombination process, the linkage discovery step is executed if necessary. We calculate the average fitness of the current epoch, compare the average with the one calculated during last epoch, and check if the improvement is great enough. When the specified threshold is reached, the current linkage groups are suitable and remain unchanged for the next PSO epoch. Otherwise, it is considered that the building blocks do not work well for the current search stage. Thus, the linkage discovery process restarts, and the linkage group is randomly reassigned. The pseudo code and complete flow of the algorithm are shown in Figures 2 and 3, respectively.

Similar research studies have been done in the literature, such as PSO with learning strategy [18, 19] and PSO with adaptive linkage learning [12]. The main difference between the proposed algorithm and them is that we introduce the recombination operator specifically designed to work with the identified building blocks. In addition, we propose a new linkage discovery technique to dynamically adapt the linkage during search process.

PSO w/ Recombination & Dynamic Linkage Discovery
Step 1: Do Finding the linkage group.
Step 2: Do PSO algorithm on the population.
Step 3: Do Recombination to generate next population.
Step 4: If fitness value improved, then go to step 2, else go to step 1. Repeat until the maximum iteration is reached.
Step 5: Do local search on the best particle.

Figure 2: Pseudocode of PSO-RDL

4 Experiments

The computer simulations are conducted to demonstrate the performance of PSO- RDL. The experiments are focused on the real-valued parameter optimization. The test problems are proposed in the special session on real-parameter optimization in CEC2005 aimed at developing high-quality benchmark functions to be publicly available to the researchers around the world for evaluating their algorithms. The description of test problems and parameter settings is provided in section 4.1. Section 4.2 shows the numerical results of the experiments as well as the linkage dynamics during optimizing several functions of different characteristics.

4.1 Test Problems

The newly proposed set of test problems includes 25 functions of different characteristics. 5 of them are unimodal problems, and other 20 are multimodal problems [26]. Due to the page restriction, we hereby present only the first 14 test problems results, including the unimodal functions, the basic multimodal functions, and the expanded functions. Experiments are conducted on the 10-D problems. In this benchmark, it is predefined that the problem is considered solved when the error is 1e-6 for problems 1-5 and 1e-2 for problems 6-14. To conduct the experiments, the number of particles is set to 20, $0.8 \leq w \leq 0.9$, $0.5 \leq \vec{\varphi}_1 \leq 2.0$,$0.5 \leq \vec{\varphi}_2 \leq 2.0$, and $V_{\max}$ restricts the particles' velocity, where $V_{\max}$ is equal to 25% of the search range. N, the number of particles selected for the recombination, is set to 25% of the swarm size. The threshold which decides if the linkage configuration should be changed is set to 5% of the previous best fitness value.

4.2 Experimental Results

The complete experiment results are listed in Tables 1, 2, and 3. In the experimental results, PSO-RDL successfully solved problems 1, 2, 4, 5, 6, and 12. Moreover, comparable results are achieved in solving problems 3, 7 ,8 , 11, 13, and 14. Unfortunately, PSO-RDL failed to solve problems 9 and 10. Figures 4, 5, 6, and 7 show how the dynamic linkage discovery technique changes the linkage configuration during the optimization process. Detailed discussion on the experimental results is presented in the next section.

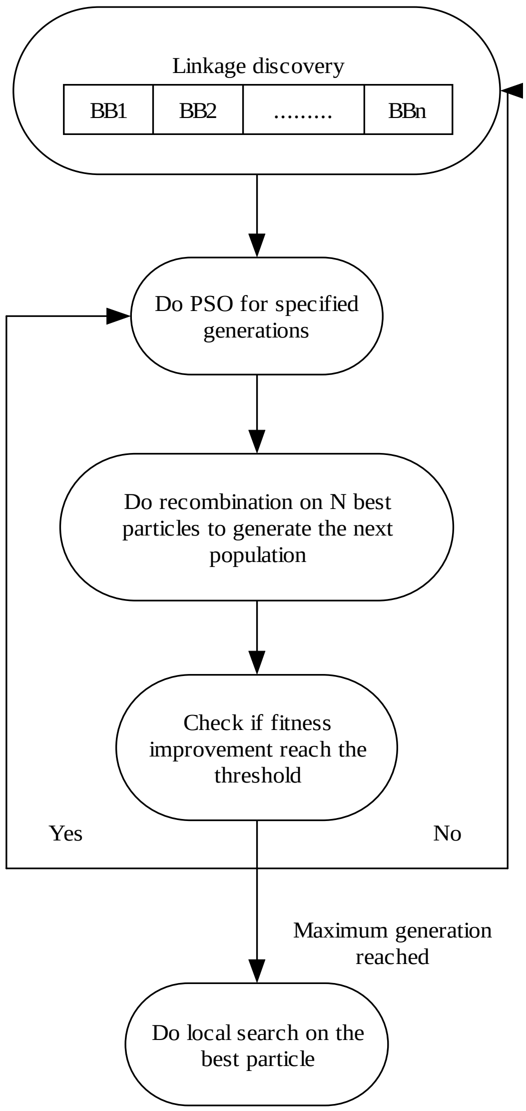

Figure 3: The flow of the PSO-RDL

5 Discussion

From the experimental results listed in Table 1, it can be considered that the proposed algorithm is able to provide good results in the benchmark. The first five functions are unimodal functions. Function 1 is shifted sphere function, Function 2 is shifted Schwefel's problem 1.2, and Function 3 is shifted rotated high condition elliptic function. These three functions have different condition numbers which make function 3 much harder than functions 1 and 2. Function 4 is shifted Schwefel's problem 1.2 with noise in fitness. Function 5 is Schwefel's problem 2.6 with global optimum on bounds. From the results, we can observe that PSO-RDL reaches the predefined tolerance level for functions 1, 2, 4, and 5. For function 3, PSO-RDL achieves an error of 1e-4 but does not meet the 1e-6 criterion. It may be caused by a multiplicator 10^6 in this objective function which greatly amplifies the error. In summary, PSO-RDL provides a sufficiently good

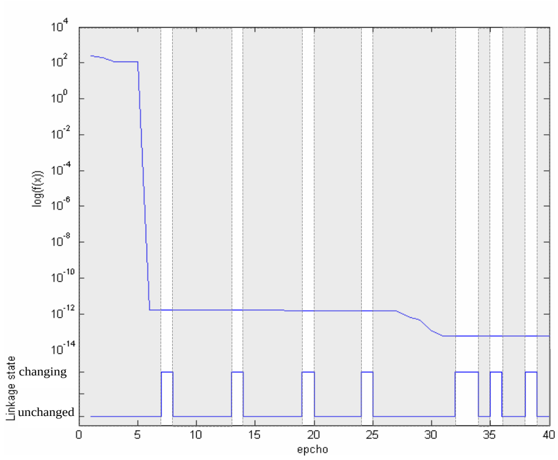

Figure 4: Fitness convergence and linkage dynamics of the Sphere function. A unimodal function which PSO-RDL solved successfully. The gray area in the figure represents the proper building blocks can improve the fitness and stay unchanged. Once the building blocks do not work well, the linkage configuration will change until the next suitable set is found.

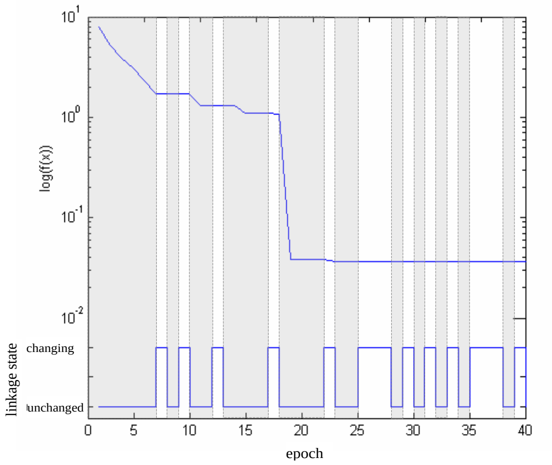

Figure 5: Fitness convergence and linkage dynamics of the Shifted Rotated Griewank's function. A multimodal function which PSO-RDL produced comparable results. The gray area in the figure represents the proper building blocks can improve the fitness and stay unchanged. Once the building blocks do not work well, the linkage configuration will change until the next suitable set is found.

performance for the unimodal functions in this benchmark.

Functions 6-14 are multimodal problems. Function 6 is shifted Rosenbrock's function, a problem with a very narrow valley from the local optimum to the global optimum, and solved by PSO-RDL. Function 7 is shifted rotated Griewank's function without bounds, and this function makes the search easily away from the global optimum. Fortunately, PSO-RDL can achieves a comparable result for this function. Function 8 is shifted rotated Ackley's function with global optimum on bounds, which has a very narrow global basin and half of the dimensions of this basin are on the bounds. Hence, the search algorithm cannot easily find the global basin when the recombination operator is used. The PSO-RDL failed on this problem in all 25 runs. Functions 9, 10, and 11 are shifted Rastrigin's function, shifted rotated Rastrigin's function, and shifted rotated Weierstrass function, respectively, all of which have a huge number of local optima. The PSO-RDL has a relatively bad performance on the first two problems comparing with traditional PSO [27] and DMS-PSO [28]. Comparable results were obtained on function 11. It may be because when the number of local optima is huge, the dissimilar individuals would likely to have similar fitness values. Although they could provide good building blocks, when different building blocks are combined to create new individuals, the offspring could have worse fitness values instead. Once the building blocks cannot be identified correctly, the genetic operator cannot work well, either. Function 12 is Schwefel's problem, and PSO-RDL achieves a 100% success rate. Functions 13 and 14 are extended functions, and the PSO-RDL produces comparable results in solving these two functions.

Observe the fitness convergence and linkage dynamics in Figures 4, 5, 6, and 7. The gray area represents the time frame when a proper linkage configuration can assist the optimization process. When the current linkage groups are not suitable, i.e. the linkage configuration cannot

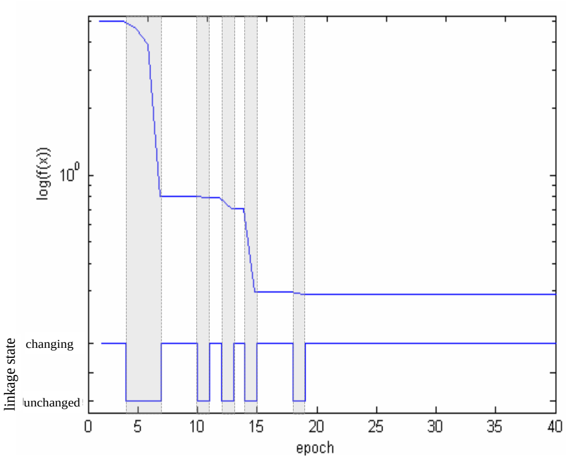

Figure 6: Fitness convergence and linkage dynamics of the Shifted Expanded Griewank's plus Rosenbrock's function. A multimodal function which PSO-RDL produced comparable results. The gray area in the figure represents the proper building blocks can improve the fitness and stay unchanged. Once the building blocks do not work well, the linkage configuration will change until the next suitable set is found.

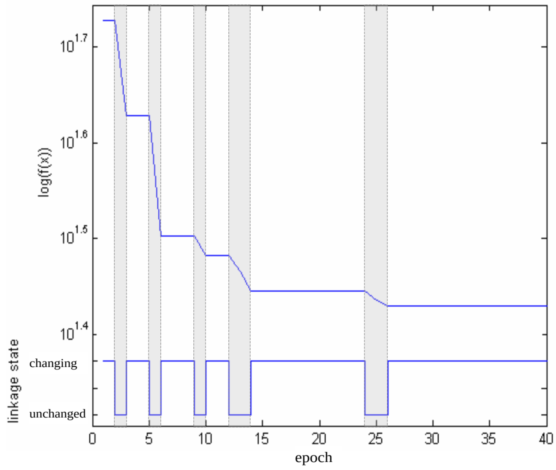

Figure 7: Fitness convergence and linkage dynamics of the Shifted Rastrigin's function. A multimodal function with large number of local optima and PSO-RDL failed to solve. The gray area in the figure represents the proper building blocks can improve the fitness and stay unchanged. Once the building blocks do not work well, the linkage configuration will change until the next suitable set is found.

assist the search, the linkage group composition will start to vibrate for some iterations until the next proper set of linkage groups is found. The phenomenon can explain the assumption that the building block's composition is dynamically changed during the search process in the real-parameter optimization problem. Thus, it is considered reasonable that we hand over the linkage adaptation to the mechanism of natural selection. Moreover, Figure 7 shows the function with a large number of local optima and PSO-RDL failed. It is clearly that the linkage configuration keeps changing all the time. As discussed above, this phenomenon indicates that when the function has a large number of local optima, it is hard to recognize the building blocks because totally different individuals may have similar fitness values. In such a case, different individual may provide their own good building blocks, but worse individuals may be created by the recombining these incompatible pieces of solutions.

Focus on the time ratio of the linkage status (changing vs. unchanged), we can observe that for Figures 4 and 5, the linkage configuration stay unchanged most of the time. Correspondingly, the proposed algorithm provide good results on these two functions. On the contrary, the linkage configuration keeps on changing in the Figures 6 and 7. Thus, our algorithm do not work very well on these two functions, although we mentioned that PSO-RDL can obtain comparable results on the shifted expanded Griewank's plus Rosenbrock's function. This is because there does not exist a very efficient algorithm for this problem so far. Hence, we can conclude that when the linkage configuration changes too often, the algorithm will fail to solve the problem with a high probability.

In this paper, we proposed a new framework by introducing the recombination mechanism with the dynamic linkage discovery technique to PSO. From the experiment, the proposed

FES		1	2	3	4	5
	1st(Min)	2.532750E+02	4.560870E+02	1.493440E+07	9.026380E+02	2.003030E+02
	7th	1.686600E+02	4.557260E+02	5.153480E+06	8.377340E+02	7.584360E+01
	13th(Median)	5.109340E+02	5.003960E+02	9.281540E+06	7.127830E+02	1.633820E+02
1E+03	19th	4.331560E+02	7.901100E+02	3.557060E+06	2.561870E+03	1.167430E+02
	25th(Max)	2.154320E+02	5.215150E+02	6.801890E+06	1.359570E+03	1.637060E+02
	mean	2.761047E+02	5.568583E+02	6.668280E+06	1.088976E+03	5.799859E+02
	Std	1.605565E+02	1.236481E+02	5.745819E+06	5.979742E+02	1.573093E+03
	1st(Min)	1.705300E-12	6.948170E-01	2.336850E+05	2.549360E+02	7.975790E-01
	7th	1.477930E-12	9.846390E-02	4.743090E+05	4.024540E+02	9.953610E-02
	13th(Median)	1.648460E-12	6.167260E-04	9.124280E+05	2.998050E+02	1.136450E+00
1E+04	19th	1.818990E-12	5.787250E-01	3.322040E+05	1.789100E+03	1.140130E+00
	25th(Max)	1.477930E-12	8.862120E-01	4.038570E+04	6.052760E+02	7.659790E+00
	mean	1.589343E-12	2.749070E-01	4.183295E+05	2.983205E+02	1.701576E+00
	Std	1.543518E-13	2.381502E-01	3.035168E+05	3.333319E+02	2.551875E+00
	1st(Min)	0.000000E+00	5.684340E-14	4.417010E-04	7.389640E-13	0.000000E+00
	7th	0.000000E+00	5.684340E-14	4.427520E-04	3.410610E-13	0.000000E+00
	13th(Median)	0.000000E+00	5.684340E-14	4.693550E-04	7.275960E-12	0.000000E+00
1E+05	19th	0.000000E+00	1.136870E-13	4.715320E-04	1.136870E-13	0.000000E+00
	25th(Max)	0.000000E+00	1.136870E-13	4.643300E-04	1.665510E-11	0.000000E+00
	mean	6.821208E-15	8.185456E-14	4.555010E-04	6.985737E-10	1.455192E-13
	Std	1.885282E-14	3.314516E-14	2.746542E-05	3.133031E-09	7.275960E-13

Table 1: Best function error values achieved when FES = 1e+3, 1e+4, and 1e+5 for functions 1-5. The predefined error is 1e-6 for these five functions. These functions are all unimodal problems, and PSO-RDL successfully solved functions 1, 2, 4, and 5. Comparable results for function 3 were obtained.

algorithm can provide a good performance on a carefully designed benchmark function set. The future research may include applying the dynamic linkage discovery technique to other evolutionary optimization algorithms, using this algorithm as an optimization tool to solve other real-world problems, and developing other linkage discover techniques for real-parameter optimization problems.

6 Conclusions

In this paper, we first surveyed on the recent studies. We recognize the importance of the linkage concept of GA and that the correct combination of GA and PSO can lead to the further algorithmic advance. We then introduced the dynamic linkage discovery technique into PSO by incorporating the recombination operator to work on the identified building blocks. We adopted the benchmark functions defined in CEC2005 to evaluate the performance of the proposed algorithm. The experimental results indicated that the proposed algorithm can provide a good performance on the benchmark functions of different characteristics.

Furthermore, the present work on PSO-RDL gives us two observations. First, in the literature, it is rarely discussed about the building blocks in real-parameter optimization problems. This work may shed light on the existence of building blocks in real-parameter optimization problems. Secondly, if building blocks do exist, then why these building blocks cannot be de-

FES		6	7	8	9	10
	1st(Min)	2.095290E+05	7.913570E+00	2.071810E+01	5.917200E+01	7.040100E+01
	7th	1.004910E+06	7.795590E+00	2.068090E+01	3.346290E+01	8.032340E+01
	13th(Median)	2.625730E+06	3.039740E+00	2.079820E+01	6.455240E+01	6.806960E+01
1E+03	19th	1.758970E+05	6.686890E+00	2.061660E+01	6.492440E+01	5.296840E+01
	25th(Max)	2.532690E+06	7.688750E+00	2.042040E+01	5.312520E+01	6.702910E+01
	mean	1.439999E+07	8.739664E+00	2.073800E+01	5.604120E+01	6.090967E+01
	Std	6.375029E+07	3.305699E+00	1.234300E-01	1.005663E+01	9.357552E+00
	1st(Min)	4.090850E+02	1.315700E+00	2.019960E+01	3.035510E+01	2.487390E+01
	7th	2.312340E+02	7.873120E-01	2.016270E+01	9.949580E+00	2.188910E+01
	13th(Median)	2.776480E+02	1.350170E+00	2.023330E+01	1.193950E+01	1.790920E+01
1E+04	19th	5.469470E+02	1.599990E+00	2.019980E+01	8.954630E+00	8.954630E+00
	25th(Max)	8.794220E+01	1.132830E+00	2.016870E+01	3.979840E+00	9.949590E+00
	mean	1.558660E+03	1.023968E+00	2.021504E+01	1.239753E+01	1.699393E+01
	Std	2.739473E+03	3.055044E-01	1.242222E-01	6.588169E+00	5.750982E+00
	1st(Min)	4.673380E-09	3.693100E-02	2.000030E+01	2.686380E+01	2.487390E+01
	7th	1.067920E-09	4.430740E-02	2.000040E+01	9.949580E+00	2.188910E+01
	13th(Median)	9.943620E-10	4.928150E-02	2.000010E+01	8.954630E+00	1.790920E+01
1E+05	19th	7.719340E-11	2.951780E-02	2.000030E+01	8.954630E+00	6.964710E+00
	25th(Max)	2.379580E-09	1.477980E-02	2.001460E+01	3.979840E+00	9.949590E+00
	mean	2.522122E-08	6.744376E-02	2.000148E+01	1.014857E+01	1.568052E+01
	Std	1.101030E-07	5.439797E-02	3.008915E-03	5.233374E+00	5.519394E+00

Table 2: Best function error values achieved when FES = 1e+3, 1e+4, and 1e+5 for functions 6-10. The predefined error is 1e-2 for these five functions. The functions are all multimodal problems, and PSO-RDL successfully solved function 6 and gave comparable results on functions 7 and 8 . However, worse results were obtained on functions 9 and 10 due to the large number of local optima.

tected by the linkage detection techniques previously proposed in the literature? According to the information obtained in this study, perhaps in a real-parameter optimization problem, the configuration of building blocks dynamically changes along with the search stage. Thus, those traditional, static linkage detection techniques failed to accomplish the task.

In this study, we introduce recombination with dynamic linkage discovery to PSO and consider the integration as a promising research direction. By combining the strength of different optimization models, we create the PSO-RDL algorithm with intriguing features and properties. We will continue to work on understanding and analyzing the real number optimization problem in order to design better evolutionary optimization algorithms in the future.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grant NSC-94-2213-E-009-120. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

FES		11	12	13	14
1E+03	1st(Min)	7.163920E+00	1.947410E+05	2.407150E+00	3.831320E+00
	7th	8.096430E+00	1.916540E+05	3.916860E+00	4.208820E+00
	13th(Median)	6.790020E+00	2.033770E+05	3.717910E+00	4.517640E+00
	19th	1.179240E+01	2.629030E+05	2.992830E+00	3.996120E+00
	25th(Max)	8.614150E+00	2.776750E+05	3.684660E+00	3.873040E+00
	mean	9.025970E+00	1.946428E+05	3.967534E+00	4.022600E+00
	Std	1.461248E+00	6.610987E+04	8.144143E-01	2.947659E-01
1E+04	1st(Min)	4.331310E+00	8.418040E+01	7.276860E-01	3.145140E+00
	7th	3.164750E+00	8.418040E+01	1.090090E+00	4.024910E+00
	13th(Median)	5.706250E+00	8.418040E+01	9.649410E-01	4.507050E+00
	19th	6.810890E+00	8.418040E+01	1.007700E+00	3.654990E+00
	25th(Max)	5.471430E+00	8.418040E+01	9.827740E-01	3.488360E+00
	mean	5.375676E+00	8.418040E+01	9.501498E-01	3.586252E+00
	Std	1.300417E+00	1.450389E-14	4.728880E-01	3.330765E-01
1E+05	1st(Min)	1.336160E+00	2.140760E-08	4.057560E-01	2.712020E+00
	7th	1.668220E+00	2.140760E-08	5.777020E-01	4.017370E+00
	13th(Median)	2.098220E+00	2.140760E-08	8.943220E-01	4.474510E+00
	19th	2.948160E+00	2.140760E-08	6.815150E-01	3.077510E+00
	25th(Max)	4.734480E+00	2.140760E-08	8.410920E-01	2.932240E+00
	mean	2.516470E+00	2.140760E-08	7.443845E-01	3.199979E+00
	Std	1.542868E+00	1.130485E-16	4.122971E-01	5.012267E-01

Table 3: Best function error values achieved when FES = 1e+3, 1e+4, and 1e+5 for functions 11-14. The predefined error is 1e-2 for these four functions. The functions are all multimodal problems, and functions 13 and 14 are extended functions. PSO-RDL successfully solved function 12 and obtained comparable results on functions 11, 13, and 14.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 1942–1948, 1995.

[2] J. Kennedy and R. C. Eberhart, "A new optimizer using paritcle swarm theory," in *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.

[3] J. H. Holland, *Adaptation in natural and artificial systems.* University of Michigan Press, 1975.

[4] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, 1989.

[5] ——, *The design of innovation: Lessons from and for competent genetic algorithms*, ser. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, 2002, vol. 7.

[6] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first result," *Complex Systems*, vol. 3, pp. 493–530, 1989.

[7] D. E. Goldberg, K. Deb, and B. Korb, "Messy genetic algorithms revisited: Studies in mixed size and scale," *Complex Systems*, vol. 4, pp. 415–444, 1990.

[8] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Lecture Notes in Computer Science*, vol. 1447, pp. 611–616, 1998.

[9] P. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," *Lecture Notes in Computer Science*, vol. 1447, pp. 601–610, 1998.

[10] M. Løvbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," in *Proceedings of Genetic and Evolutionary Computation Conference 2001 (GECCO-2001)*, 2001.

[11] M. Settles and T. Soule, "Breeding swarms: A GA/PSO hybrid," in *Proceedings of Genetic and Evolutionary Computation Conference 2005 (GECCO-2005)*, Washington, DC, USA, 2005, pp. 161–168.

[12] D. Devicharan and C. K. Mohan, "Particle swarm optimization with adaptive linkage learning," in *Proceedings of the 2004 Congress on Evolutionary Computation (CEC2004)*, Portland, Oregon, 2004, pp. 530–535.

[13] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," *Lecture Notes in Computer Science*, vol. 1447, pp. 591–600, 1998.

[14] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the IEEE Congress of Evolutionary Computation*, vol. 3, 1999, pp. 1951–1957.

[15] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *IEEE Congress on Evolutionary Computation*, 1999, pp. 1931–1938.

[16] R. C. Eberhart and X. Hu, "Multiobjective optimization using dynamic neighrborhood particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, Hawaii, 2002, pp. 1677–1681.

[17] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle optimizer," in *IEEE International Swarm Intelligence Symposium*, 2005, pp. 124–129.

[18] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Particle swarm optimization algorithm with novel learning strategies," in *International Conference on Systems, Man and Cybernetics*, The Netherlands, 2004.

[19] J. J. Liang, P. N. Suganthan, A. K. Qin, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transaction on Evolutionary Computation*, 2006, (To appear).

[20] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna," in *IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting*, San Antonio TX, 2002.

[21] X. H. Shi, Y. H. Lu, C. G. Zhou, H. P. Lee, W. Z. Lin, and Y. C. Liang, "Hybrid evolutionary algorithms based on PSO and GA," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003)*, 2003, pp. 2393–2399.

[22] A. Singh, D. E. Goldberg, and Y.-p. Chen, "Modified linkage learning genetic algorithm for difficult non-stationary problems," in *Proceedings of Genetic and Evolutionary Computation Conference 2002 (GECCO-2002)*. Morgan Kaufmann Publishers Inc, 2002, p. 699.

[23] M. Tezuka, M. Munetomo, and K. Akama, "Linkage identification by nonlinearity check for real-coded genetic algorithms," in *Proceedings of Genetic and Evolutionary Computation Conference 2004 (GECCO-2004)*, 2004, pp. 222–233.

[24] G. R. Harik, "Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Tech. Rep. 97005, 1997.

[25] J. Smith and T. C. Fogarty, "An adaptive poly-parental recombination strategy," in *Proceedings of AISB-95 Workshop on Evolutionary computing*, 1995, pp. 48–61.

[26] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-p. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," National Chiao Tung University, Natural Computing Laboratory, Hsinchu, Taiwan, NCLab Report No. NCL-TR-2005001, 2005.

[27] M. F. Tas getiren, Y.-C. Liang, G. Gencyilmaz, and I. Eker, "Global optimization of continuous functions using particle swarm optimization," Tech. Rep., 2005.

[28] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC2005)*, vol. 1, 2005, pp. 522–528.