Particle Swarm Optimization With Recombination and Dynamic Linkage Discovery

Ying-Ping Chen, *Member, IEEE*, Wen-Chih Peng, *Member, IEEE*, and Ming-Chung Jian

***Abstract*—In this paper, we try to improve the performance of the particle swarm optimizer by incorporating the linkage concept, which is an essential mechanism in genetic algorithms, and design a new linkage identification technique called *dynamic linkage discovery* to address the linkage problem in real-parameter optimization problems. Dynamic linkage discovery is a costless and effective linkage recognition technique that adapts the linkage configuration by employing only the selection operator without extra judging criteria irrelevant to the objective function. Moreover, a recombination operator that utilizes the discovered linkage configuration to promote the cooperation of particle swarm optimizer and dynamic linkage discovery is accordingly developed. By integrating the particle swarm optimizer, dynamic linkage discovery, and recombination operator, we propose a new hybridization of optimization methodologies called *particle swarm optimization with recombination and dynamic linkage discovery (PSO-RDL)*. In order to study the capability of PSO-RDL, numerical experiments were conducted on a set of benchmark functions as well as on an important real-world application. The benchmark functions used in this paper were proposed in the 2005 Institute of Electrical and Electronics Engineers Congress on Evolutionary Computation. The experimental results on the benchmark functions indicate that PSO-RDL can provide a level of performance comparable to that given by other advanced optimization techniques. In addition to the benchmark, PSO-RDL was also used to solve the economic dispatch (ED) problem for power systems, which is a real-world problem and highly constrained. The results indicate that PSO-RDL can successfully solve the ED problem for the three-unit power system and obtain the currently known best solution for the 40-unit system.**

***Index Terms*—Building blocks, dynamic linkage discovery, economic dispatch (ED), genetic algorithms (GAs), genetic linkage, particle swarm optimization (PSO), recombination operator, valve-point effect.**

I. Introduction

THE PARTICLE swarm optimizer (PSO), which was introduced by Kennedy and Eberhart in 1995 [1], [2], emulates the flocking behavior of birds to solve 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. These particles fly through the D-dimensional problem space by learning from the historical information of all the particles. There are global and local versions of PSO. 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 were 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), which were introduced by Holland [3], are stochastic population-based search and optimization algorithms loosely modeled after the paradigm of evolution. GAs 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 has been theorized by researchers [3]–[5] with building block processing, such as creating, identifying, and exchanging. Building blocks are conceptually noninferior subsolutions that 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. Accordingly, the GA's search capability can be greatly improved by identifying building blocks accurately and preventing crossover operation from destroying them [6], [7]. Hence, linkage identification, i.e., the procedure to recognize building blocks, plays an important role in GA optimization.

The two aforementioned optimization techniques are both population based and have been proven successful in solving a variety of difficult problems. However, both models have strengths and weaknesses. Comparisons between GAs and PSO can be found in the literature [8], [9] and suggest that a hybrid of these two algorithms may lead to further advances. As a consequence, a host of studies on the hybridization of GAs and PSO have been proposed and examined. Most of these research works try to incorporate genetic operators into PSO [10], [11], while some try to introduce the concept of linkage into PSO [12]. According to the similar idea employed by linkage PSO [12], this paper tries to introduce the recombination mechanism working on building blocks to enhance the capability of PSO with the linkage concept.

Manuscript received November 20, 2006; revised April 20, 2007. The work was supported in part by the National Science Council of Taiwan, R.O.C., under Grant NSC-95-2221-E-009-092 and Grant NSC-95-2627-B-009-001 and in part by the Aiming for the Top University and Elite Research Center Development Plan (ATU Plan), National Chiao Tung University and Ministry of Education, Taiwan, R.O.C. This paper was recommended by Associate Editor Y. S. Ong.

Y.-P. Chen and W.-C. Peng are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: ypchen@cs.nctu.edu.tw; wcpeng@cs.nctu.edu.tw).

M.-C. Jian is with Airoha Technology Corporation, Hsinchu 300, Taiwan, R.O.C. (e-mail: mcjian@gmail.com).

Digital Object Identifier 10.1109/TSMCB.2007.904019

This paper presents a research project that aims to address the linkage problem in real-parameter optimization and introduces the linkage concept to the particle swarm optimizer. Thus, there are the following three primary objectives.

1) With the hypothesis that linkage also exists in real-parameter optimization, a linkage identification technique is needed to address the linkage problem. This paper provides both the linkage identification mechanism and observations from numerical experiments to support this hypothesis.
2) In order to improve the performance of the particle swarm optimizer, the linkage concept is introduced. An optimization algorithm that incorporates this mechanism is developed, and numerical experiments are conducted to evaluate the performance of the proposed methodology.
3) The economic dispatch (ED) problem, which is an essential topic in power control systems, can be appropriately handled and optimized.

Focusing on the three objectives, in this paper, we propose a dynamic linkage discovery technique to dynamically and effectively detect the building blocks of the objective function during the whole evolutionary optimization process. This technique differs from those traditional linkage detection techniques in that the evaluation cost is eliminated. The idea is to dynamically adapt the linkage configuration according to the search process and the feedback from the environment. Thus, this technique is costless and easy to be integrated into existing search algorithms. Our method introduces the linkage concept and the recombination operator to the operation of the particle swarm optimizer. Furthermore, in order to efficiently solve the ED problem, we incorporate the proposed algorithm with a new constraint handling technique. We use the three-unit [13] and 40-unit [14] problems with the nonsmooth fuel cost function considering valve-point effects [13] found in the literature of power systems as the experiments.

This paper is organized as follows. Section II provides a survey of related work in the literature. Section III describes the proposed method in detail. The three mechanisms including the particle swarm optimizer, dynamic linkage discovery technique, and recombination operator are introduced. The framework consisting of the three components is then presented. Section IV shows the experimental results that evaluate the performance of the proposed algorithm in the 2005 Institute of Electrical and Electronics Engineers Congress on Evolutionary Computation (IEEE CEC 2005) benchmark. Section V applies the proposed framework to the ED problem, which is a significant topic in power systems. Section VI gives a summary of this paper. The future work and the main conclusions of this paper are provided.

II. Related Work

The traditional PSO algorithm, which is described in [1], consists of a number of particles representing possible solutions to a numerical problem and 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

$$v_i^{t+1}(j) = wv_i^t(j) + c_1\varphi_{1i}(j)\left[p_i^t(j) - x_i^t(j)\right] + c_2\varphi 2i(j)\left[p_g^t(j) - x_i^t(j)\right]$$

where t is the time index, i is the particle index, and j is the dimension index. $\vec{p_i}$ is the individual best position. $\vec{p_g}$ is the known global best position. w is the inertia weight described in [15]. c_1 and c_2 are the acceleration rates of the cognitive and social parts, respectively. $\vec{\varphi_1}$ and $\vec{\varphi_2}$ are random values different for each particle i as well as for each dimension j. The velocity update rule with constriction coefficients is proposed in [16]. The position of each particle is also updated in each iteration by adding the velocity vector to the position vector, i.e.,

$$x_i^{t+1}(j) = x_i^t(j) + v_i^t(j).$$

The particles used in this paper have no neighborhood restriction, which means each particle can affect all other particles. In the local version of PSO, $\vec{p_g}$ is replaced by $\vec{p_n}$, i.e., 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 [17]–[19]. Furthermore, studies on modifying the rule of updating position and velocity are also implemented [12], [20], [21]. Devicharan and Mohan [12] first computed the elements of the 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.* [20], [21] proposed learning strategies where each dimension of a particle learned from just one 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 genetic operators and/or mechanisms, many hybrid GA/PSO algorithms have been proposed and tested on function minimization problems [10], [11], [22], [23]. 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.* [22] tested a hybrid that used the GA algorithm to initialize the PSO population and another in which the PSO initialized the GA population. Shi *et al.* [23] proposed two approaches. The main idea of the proposed algorithm was to 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, i.e., the breeding ratio, to determine the proportion of the population that 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 GAs [3], [4], [6], [7]. Because it is hard, if not impossible, to guarantee the user-designed chromosome representation that provides tightly linked building blocks when the problem domain knowledge is unavailable, a variety of 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 linkage has been addressed in the GA literature [24]–[26]. Furthermore, some researchers try to introduce the linkage concept to PSO and formulate linkage-sensitive PSO algorithms [12], [20], [21].

III. FRAMEWORK

The proposed algorithm introduces the recombination operator with the dynamic linkage discovery technique to the particle swarm optimization (PSO). Dynamic linkage discovery adapts the linkage configuration by utilizing natural selection without incorporating extra judging criteria. Furthermore, a specifically designed recombination operator is employed to work with the identified building blocks.

A. Dynamic Linkage Discovery

In the literature, most linkage identification techniques were proposed and tested on trap functions [27]. There are relatively fewer studies on handling linkage in real-number optimization problems. From the survey of linkage learning, Tezuka *et al.* identified linkage by nonlinearity check on real-coded GA [25]. Such a technique has also been incorporated with the particle swarm optimizer [12]. Different from this perturbation-based linkage identification technique, we propose the *dynamic linkage discovery* in this paper.

In PSO, particles are encoded as real-number vectors. Because of this representation, we use the term "linkage" to indicate the interrelation among dimensions as in the literature [12], [21], [28]. At different stages of an optimization process, the linkage configuration may be different according to the fitness landscape and the corresponding population distribution. Hence, in this paper, we make a hypothesis that the relation between different dimensions is dynamically changed along with the optimization process from the viewpoint of the population. Acting on this hypothesis, the linkage configuration should be updated accordingly such that the obtained information regarding the function structure embedded in the population distribution can be extracted and utilized.

For most problems, it is difficult to exactly identify the linkage configuration, especially when the linkage configuration may change over time. Instead of incorporating extra artificial criteria for linkage adaptation, we entrust the task to natural selection. As a consequence, we propose the dynamic linkage discovery technique, and we call the PSO combined with recombination and 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 evolutionary optimization process, PSO-RDL assigns a set of random linkage groups and then adjusts the linkage configuration according to the objective values. If the average fitness value of the current population is improved above a specified threshold, the current linkage configuration is considered appropriate and remains unchanged. Otherwise, the linkage groups will be reassigned at random. Dynamic linkage discovery may seem similar to the mechanism of "random neighborhood" defined in *Standard PSO 2006* and downloadable at Particle Swarm Central [29] due to the concept of "if no improvement, redefine the neighborhoods." The essential difference is that random neighborhood works on particles while dynamic linkage discovery works on dimensions. In this sense, the two methodologies are orthogonal and might cooperate with each other. Finally, the pseudocode of dynamic linkage discovery is shown in Fig. 1.

```
procedure DYNAMIC-LINKAGE-DISCOVERY(δ)
    // δ is the fitness improvement
    if δ > threshold then
        Linkage configuration remains unchanged
    else
        Call Dynamic-Linkage-Group
    end if
end procedure
procedure DYNAMIC-LINKAGE-GROUP
    Generate a group number G, a random integer uniformly distributed in [1, Dim]
    // Dim is the number of dimension
    for each dimension do
        Assign a random integer uniformly distributed in [1, G]
    end for
    // Dimensions assigned the same integer belong to the same building block
end procedure
```

Fig. 1. Pseudocode of dynamic linkage discovery.

B. Recombination Operator

For evolutionary algorithms, the merits of crossover have been an essential research topic. Instead of the traditional two-parent recombinatory chromosome reproduction, there has been considerable discussion of multiparent crossover mechanisms [30]–[32]. Work on multiparental recombination techniques (with a fixed number of parents) [31] showed that n-parental inheritance (with n greater than 2 but less than the size of the population) can be advantageous. Based on previous research work, we develop a multiparental recombination operator for constructing the offspring population.

In this paper, since we have discovered the linkage configuration in order to make good use of the obtained information, we specifically design a new recombination operator. After selection, we consider the selected individuals as a building block pool. In the recombination process, every offspring is created by choosing and recombining building blocks from the pool at random. We use this recombination process to generate the next population. Fig. 2 shows an illustration of how a new individual is generated. By repeating the process, we can construct a new population in which each particle is composed of the building blocks. The pseudocode for constructing a new population is shown in Fig. 3.

C. Recombination With Dynamic Linkage Discovery in PSO

For the convenience of analysis and development, in this paper, we use a modified version of PSO [33]. In the proposed algorithm, we repeat PSO for a certain number of generations, which is called a *PSO epoch*. After each PSO epoch, we select the n best particles to establish the building block pool and conduct the recombination operation. After the recombination process, the linkage discovery step is executed if necessary.

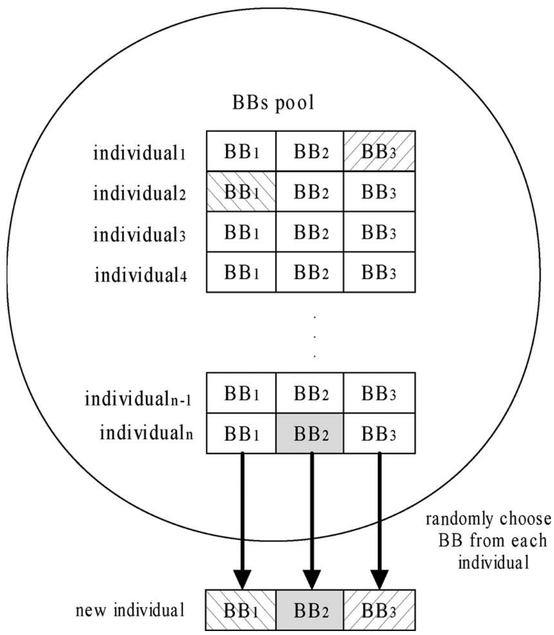

Fig. 2. New particle is generated through the recombination operator.

```
procedure NEW-POPULATION
    Select n best individuals from the current population
    Establish a building block pool from the selected individuals
    i ← 0
    while i < population size do
        i ← i + 1
        Construct a new particle with recombination
    end while
end procedure
```

Fig. 3. Constructing a new population by using recombination.

```
procedure PSO-RDL
    δ ← a value larger than threshold
    repeat
        Call Dynamic-Linkage-Discovery(δ)
        repeat
            Do PSO on the population
        until a PSO epoch is due
        Call New-Population
        Calculate the fitness improvement δ
    until the maximum iteration is reached
    Do local search on the best particle
end procedure
```

Fig. 4. Pseudocode of PSO-RDL.

We calculate the average fitness of the current epoch, compare the average to that calculated during the last epoch, and check whether the improvement is significant enough. When the specified threshold is reached, the current linkage configuration is considered appropriate and remains unchanged for the next PSO epoch. Otherwise, the linkage discovery process starts. The pseudocode is shown in Fig. 4.

Similar studies have been done in the literature, such as PSO with learning strategy [20], [21] and PSO with adaptive linkage learning [12]. The main difference between PSO-RDL and these previously proposed methods is the introduction of the recombination operator specifically designed to work with the identified building blocks. In addition, a new linkage discovery technique to dynamically adapt the linkage configuration during the search process is proposed. Furthermore, since the local search is utilized in the proposed framework, PSO-RDL can also be regarded as a memetic algorithm. Interested readers can refer to the related publications [34]–[37].

IV. EXPERIMENTAL RESULTS ON BENCHMARK

Computer simulations were conducted to evaluate the performance of PSO-RDL. The test problems [38] are proposed in the special session on real-parameter optimization in IEEE CEC 2005. The reason we use the benchmark to test PSO-RDL is to understand how well PSO-RDL can do on the artificial test functions and to observe the runtime dynamics. The method we adopt in this paper to conduct the comparison is to directly cite the numerical results published and available in the literature. In this section, the test problems, parameter settings, and numerical results are presented.

A. Test Functions

The set of test problems proposed in IEEE CEC 2005 includes in total 25 functions of different characteristics. Five of them are unimodal problems, and the others are multimodal problems [38]. However, because some of the 25 functions are solved by none of the algorithms compared in this paper, we use only 13 test functions from the benchmark of ten dimensions, i.e., ten decision variables, in this paper.

- Unimodal Functions (5):
  1) F_1: Shifted Sphere Function;
  2) F_2: Shifted Schewefel's Problem 1.2;
  3) F_3: Shifted Rotated High Conditioned Elliptic Function;
  4) F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness;
  5) F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds.
- Multimodal Functions (8):
  — Basic Functions (7):
    1) F_6: Shifted Rosenbrock's Function;
    2) F_7: Shifted Rotated Griewank's Function without Bounds;
    3) F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds;
    4) F_9: Shifted Rastrigin's Function;
    5) F_{10}: Shifted Rotated Rastrigin's Function;
    6) F_{11}: Shifted Rotated Weierstrass Function;
    7) F_{12}: Schwefels' Problem 2.13.
  — Hybrid Composition Function (1):
    1) F_{15}: Hybrid Composition Function.

Please note that we keep the function number assigned in the original benchmark for reference. The bias of fitness value for each function $f(x^*)$, the search ranges $[X_{\min}, X_{\max}]$, and the initialization range of each function are given in Table I.

TABLE I
GLOBAL OPTIMA, SEARCH RANGES, AND INITIALIZATION RANGES OF THE ADOPTED TEST FUNCTIONS

f	$f(x^*)$	Search Range	Initialization Range
f_1	-450	[-100,100]	[-100,100]
f_2	-450	[-100,100]	[-100,100]
f_3	-450	[-100,100]	[-100,100]
f_4	-450	[-100,100]	[-100,100]
f_5	-310	[-100,100]	[-100,100]
f_6	390	[-100,100]	[-100,100]
f_7	-180	$[-\infty,\infty]$	[0,600]
f_8	-140	[-32,32]	[-32,32]
f_9	-330	[-5,5]	[-5,5]
f_{10}	-330	[-5,5]	[-5,5]
f_{11}	90	[-0.5,0.5]	[-0.5,0.5]
f_{12}	-460	$[-\pi,\pi]$	$[-\pi,\pi]$
f_{15}	120	[-5,5]	[-5,5]

TABLE II
PARAMETER SETTINGS IN THE NUMERICAL EXPERIMENTS

Parameter	Value
Swarm size	20
Inertia weight (w)	0.65
Cognitive acceleration rate (c_1)	1.48
Social acceleration rate (c_2)	1.48
Maximum velocity	25% of the search range
PSO epoch	50
Selected particles for recombination	5
Improvement threshold	2% of the best fitness

B. Parameter Settings

The parameter settings for PSO-RDL employed in this paper are described as follows: the number of particles is 20, $w = 0.65$, $c_1 = 1.48$, $c_2 = 1.48$, and $V_{\max}$ is equal to 25% of the search range. The PSO epoch is 50 generations. If the PSO epoch is too long, the whole swarm may have converged, and no further information exchange is needed. On the other hand, if the PSO epoch is too short, the linkage discovery mechanism may not be able to learn from a group of random particles. The number of particles selected for the recombination is 5. The threshold that determines whether the linkage configuration should be changed is set to 2% of the previous best fitness value. A high fitness improvement threshold may lead to an ever-changing linkage configuration, while a low threshold may not be helpful to guide the search and becomes useless. A list of the parameter settings is shown in Table II.

C. Experimental Results

Table III shows the number of successfully solved problems for PSO-RDL and other evolutionary algorithms. According to the criteria specified in the IEEE CEC 2005 benchmark, PSO-RDL successfully solved problems 1, 2, 4, 5, 6, 7, 12, and 15. Moreover, comparable results are achieved in solving problems 3, 8, and 11. Unfortunately, PSO-RDL failed to solve problems 9 and 10. Figs. 5–8 demonstrate how the dynamic linkage discovery technique changes the linkage configuration during the optimization process.

TABLE III
PROBLEMS SOLVED BY PSO-RDL AND OTHER ADVANCED EVOLUTIONARY ALGORITHMS. UNIMODAL: UNIMODAL FUNCTIONS; BASIC: BASIC MULTIMODAL FUNCTIONS; HYBRID: HYBRID COMPOSITION FUNCTION

Method	Unimodal (5)	Basic (7)	Hybrid (1)
DMS-PSO	1,2,3,5 (4)	6,7,9,12 (4)	15 (1)
PSO-RDL	1,2,4,5 (4)	6,7,12 (3)	15 (1)
PSO	1,2,4,5 (4)	6,7,12 (3)	*
LR-CMA-ES	1,2,3,4,5 (5)	6,7,12 (3)	*
SPC-PNX	1,2,4,5 (4)	6,7,11 (3)	*
DE	1,2,3,4,5 (5)	6,9 (2)	*
Sa-DE	1,2,4 (3)	9,12 (2)	15 (1)

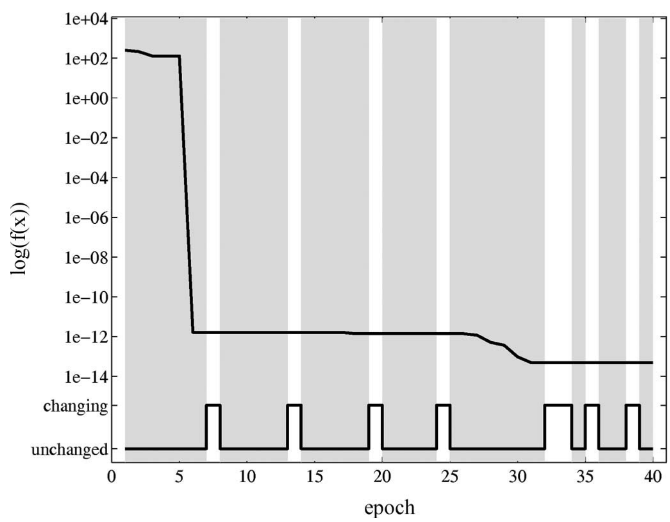

Fig. 5. Convergence and linkage dynamics for the Sphere function.

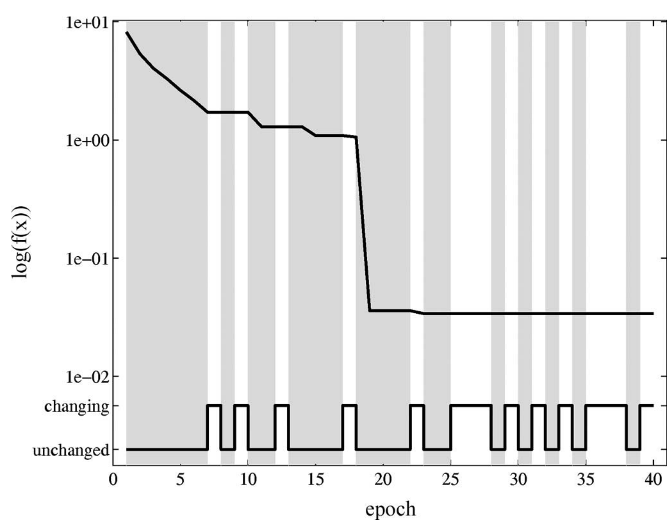

Fig. 6. Convergence and linkage dynamics for the Shifted Rotated Griewank's function.

D. Discussion

According to the results listed in Table III, it can be considered that PSO-RDL delivers a similar level of performance compared to other advanced evolutionary algorithms on the artificial test functions. For the total number of solved problems, PSO-RDL can solve eight problems and is ranked top two. Although slightly inferior to dynamic multi-swarm particle

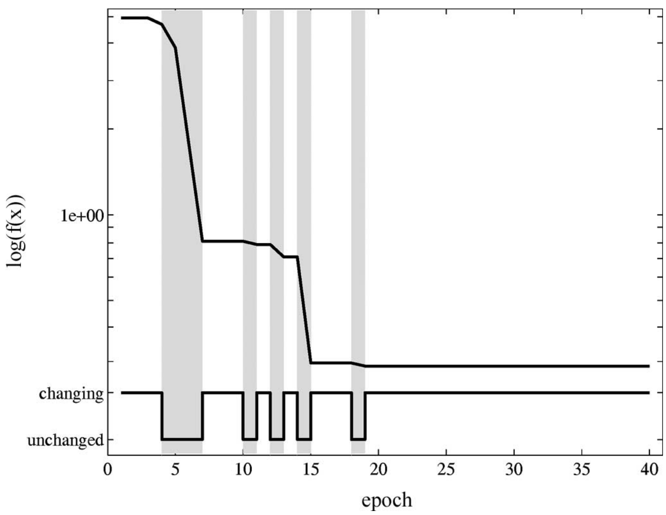

Fig. 7. Convergence and linkage dynamics for the Shifted Expanded Griewank's plus Rosenbrock's function (a multimodal function).

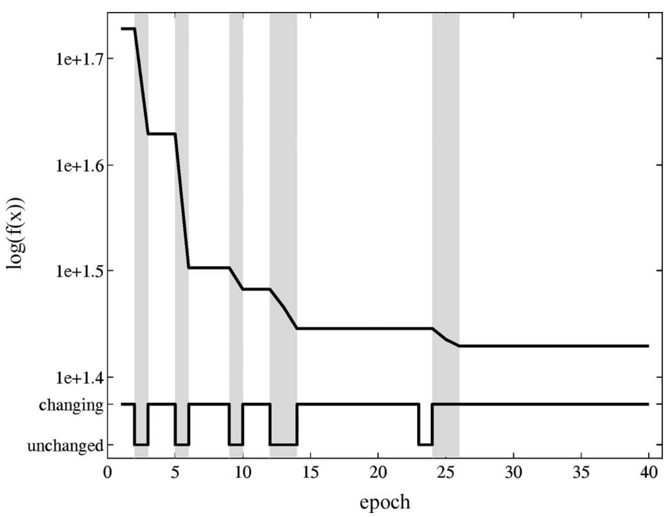

Fig. 8. Fitness convergence and linkage dynamics of PSO-RDL for the Shifted Rastrigin's function (a multimodal function with many local optima).

swarm optimizer (DMS-PSO), PSO-RDL performs well on the artificial test functions.

Because the main purpose of using the benchmark is to understand the behavior of PSO-RDL, observations for fitness convergence and linkage dynamics are provided in Figs. 5–8. The gray areas represent the time frames when a proper linkage configuration is assisting the optimization mechanism. When the linkage configuration is not appropriate for the current optimization stage, i.e., the linkage configuration fails to assist the search, the linkage group composition will start to vibrate for some iterations until the next proper set of linkage groups is found, as shown in Figs. 5 and 6. The phenomenon evidently verifies our assumption that the building block's composition is dynamically changed during the search process in the real-parameter optimization problem. Thus, it is reasonable that we hand over linkage adaptation to the mechanism of natural selection. Moreover, Fig. 8 shows the linkage dynamics for the function that PSO-RDL failed to solve. It can be seen that the linkage configuration keeps changing all the time. As discussed above, this phenomenon indicates that when the function may have no building block structure or is too difficult for PSO-RDL, an appropriate linkage configuration cannot be found, and the optimization task cannot be accomplished.

Focusing on the time ratio of the linkage status (changing versus unchanged), we can observe that for Figs. 5 and 6, the linkage configuration remains unchanged for most of the time. Correspondingly, PSO-RDL provides good results on these two functions. On the contrary, the linkage configuration keeps changing in Figs. 7 and 8. Thus, our algorithm does not work well on these two functions, although PSO-RDL can obtain numerical results comparable to those obtained by other algorithms on the shifted expanded Griewank's plus Rosenbrock's function. Hence, we can conclude that when the linkage configuration changes too often, PSO-RDL may fail to solve the problem with a high probability. Such a "signal" may be a performance indicator for PSO-RDL and worth further investigation.

V. Real-World Applications

In this section, we employ PSO-RDL to handle a real-world application, i.e., the ED problem, which is an essential topic in power systems. Thanks to the importance of the ED problem, researchers have been making a host of attempts to find better solutions. Among the promising sets of evolutionary optimization methods for tackling the ED problem are GAs [13], [39]–[42], evolutionary programming [14], [43]–[45], and PSO [46]–[49]. In order to observe the performance as well as to obtain better solutions, we will apply PSO-RDL on the ED problem. For this purpose, the following topics will be covered in this section.

- ED problem: A brief introduction to the ED problem.
- PSO-RDL for ED: PSO-RDL used to solve the ED problem.
- Experimental results: The numerical results for the 3- and 40-unit ED problems. The comparisons of PSO-RDL with other algorithms are also presented.

A. ED Problem

With the development of modern power systems, the ED problem has received increasing attention because many aspects of power systems are involved. The ED problem consists of allocating the total generation required among the available generation units, assuming that a unit commitment is previously determined. The objective aims to minimize fuel cost subject to physical and operational constraints. As a result, the ED problem is to find the optimal output combination of the power generations that minimizes the total generation cost while satisfying the equality and inequality constraints. In order to model the ED problem, a simplified cost function [50] of each generator that is represented as a quadratic function can be put as

$$C = \sum_{j \in J} F_j(P_j) \tag{1}$$

$$F_j(P_j) = a_j P_j^2 + b_j P_j + c_j \tag{2}$$

where

C	total generation cost;
J	set for all generators;
P_j	electrical output of generator j;
F_j	cost function for generator j;
a_j, b_j, c_j	cost coefficients for generator j.

In the real world, the total generation should be equal to the total system demand plus the transmission network loss. However, in a number of studies reported in the literature [13], [39], [40], [42], the network loss is not considered for simplicity. For the purpose to directly compare the numerical results obtained by PSO-RDL with those available in the literature, the transmission network loss is also not considered in this paper. Hence, the constraints of the problem include two main parts. The first part is the equality constraint. The sum of the output of all generators must be equal to the total system demand, which can be described as

$$D = \sum_{j \in J} P_j \tag{3}$$

where D is the total system demand.

Second, the generation output of each unit has to be within its minimum and maximum limits. Such a requirement introduces the inequality constraint. The inequality constraint for generator j can be put as

$$P_{j\min} \leq P_j \leq P_{j\max} \tag{4}$$

where $P_{j\min}$ and $P_{j\max}$ are the minimum and maximum outputs of generator j, and P_j is the desired output.

In reality, the objective function of the ED problem is more complicated than (2) due to the valve-point effect and the change of fuels. Therefore, the nonsmooth cost functions should be considered instead of (2), which is the most simplified form. The inclusion of the valve-point loading effect makes the modeling of the incremental fuel cost function of the generators more practical and increases the nonlinearity as well as number of local optima. The incremental fuel cost function of the generating units with valve-point loadings [13] can be represented as

$$F_j(P_j) = a_j P_j^2 + b_j P_j + c_j + \left| e_j \sin\left(f_j \times (P_{j\min} - P_j)\right)\right| \tag{5}$$

where e_j and f_j are the coefficients for generator j to reflect the valve-point effect.

In this paper, we focus on solving the ED problem with the valve-point effect, which is modeled as (5). Thus, the combination of (1) and (5) is the objective function for PSO-RDL to optimize, and a solution to the ED problem is a set of generation outputs specified for each generator in question. In addition, a solution is called a *feasible solution* if it satisfies the equality constraint (3) as well as the inequality constraint (4). Otherwise, it is called an *infeasible solution*. The regions in the search space composed of feasible solutions are called *feasible regions*, and the others are called *infeasible regions*.

B. PSO-RDL for ED

In order to solve the ED problem with PSO-RDL, the equality and inequality constraints have to be appropriately handled. To address this issue, we devise a constraint handling technique based on the concepts of repair and penalty. With the repair mechanism, infeasible solutions are modified to be feasible ones. For the ED problem, although the inequality constraints (4) might need to be handled in traditional mathematical programming methods, they can be ignored in this paper because the control of decision variable ranges is a built-in functionality of evolutionary algorithms.

As for the equality constraint (3), we repair infeasible solutions with the following procedure. We generate an integer sequence from one to the number of generators with a uniformly random order. Each integer in the sequence represents one generator that needs processing. The sequence indicates the order in which the denoted generator is processed. For example, for four generators, if we randomly generate a sequence 3, 1, 2, 4, the sequence means that we will first process unit 3, then unit 1, unit 2, and unit 4. In this order, we check the equality constraint, i.e., the sum of the total generation output must be equal to the system demand. If the equality constraint is not satisfied, the output of the generator under processing is modified according to

$$P_i' = \min\left(\text{UBound}(P_i), \max\left(\left(D - \sum_{j \in J, j \neq i} P_j\right), \text{LBound}(P_i)\right)\right) \tag{6}$$

where D is the system power demand, J is the set for all generators, and $\text{LBound}(P_i)$ and $\text{UBound}(P_i)$ are the lower bound and upper bound of P_i.

The aforementioned repair procedure is conducted with a probability of 0.4 and adjusts the output of the generators one by one until the solution is feasible. However, if all infeasible solutions are repaired, the population diversity may be greatly reduced. Therefore, to preserve the diversity of the population, only a portion of infeasible solutions is repaired. For the rest of the infeasible solutions, we use a penalty function to indicate the infeasibility. The penalty function was designed as

$$\text{Objective value with penalty} = C + w_p \left| D - \sum_{j \in J} P_j \right| \tag{7}$$

where C is the total generation cost, and the $w_p = 10^5$ is the penalty coefficient.

By incorporating the constraint handling technique, PSO-RDL is capable of solving the ED problem. In order to verify the feasibility and demonstrate the performance of PSO-RDL on the ED problem, numerical experiments were conducted, and the results are given in Section V-C.

TABLE IV
PARAMETERS FOR TEST CASE I (THREE-UNIT SYSTEM) WITH VALVE-POINT LOADING EFFECT. a, b, c, e, AND f ARE THE COST COEFFICIENTS IN THE FUEL COST FUNCTION: $F_j(P_j) = a_jP_j^2 + b_jP_j + c_j + |e_j \sin(f_j \times (P_{jmin} - P_j))|$

Unit	P_{min}	P_{max}	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	100	400	0.00482	7.97	78	150	0.063
3	50	200	0.00194	7.85	310	200	0.042

C. Experiments

In this section, the test problems will be described followed by the presentation of the numerical results.

1) Test Problems: We focus on solving the ED problem with nonsmooth functions considering the valve-point effect. We employed PSO-RDL to solve two ED problems, one with three generators and the other with 40 generators. The input data for the three-unit system are given in [13], and those for the 40-unit are given in [14]. The detail parameters include each generator output range, and related coefficients in both systems are listed in Tables IV and V. The total demands for the 3- and 40-unit systems are 850 and 10 500 MW, respectively. The global optimum solution for the three-unit system is proven to be 8 234.07 [51]. For the 40-unit system, the global optimum has not been determined. The best known solution in the literature is 122 252.265 [48]. The parameter settings for PSO-RDL are identical to those listed in Table II. The probability to repair infeasible solutions is 0.4.

2) Experimental Results: The experiments were conducted for 100 independent runs to evaluate the performance of PSO-RDL on the ED problem. The numerical results for the three-unit system are given in Table VI, and the results are compared to those of GA [13], improved evolutionary programming (IEP) [52], evolutionary programming (EP) [44], and modified particle swarm optimization (MPSO) [48]. The results indicate that PSO-RDL has successfully found the reported global optimum solution [51] as EP and MPSO. For the 40-unit system, the results are compared to those obtained by other advanced methods described in [14], such as classical EP (CEP), fast EP (FEP), modified FEP (MFEP), improved FEP (IFEP), as well as MPSO in [48]. The best solution obtained by PSO-RDL is 121 468.820, which is better than the previously known best solution, i.e., 122 252.265 reported in [48]. The best solutions obtained by each method are shown in Table VII. In order to statistically compare the results, we show the numbers of solutions for the 100 independent runs in each range of cost in Table VIII. Finally, the generation outputs and the corresponding costs of the best solution obtained by PSO-RDL are provided in Table IX for verification.

For the ED problem, we also follow the method to conduct the comparison by directly citing the numerical results published and available in the literature. Moreover, as pointed out in [13], [14], and [39], the test problem instances cannot be handled by classical LaGrangian techniques due to the lack of the monotonically increasing nature. As a consequence, the algorithms performing well on the artificial functions such as DMS-PSO [28] and local restart covariance matrix adaptation evolution strategy (LR-CMA-ES) [53] as well as the classical LaGrangian techniques are not included in the comparison.

TABLE V
PARAMETERS FOR TEST CASE II (40-UNIT SYSTEM) WITH VALVE-POINT LOADING EFFECT. a, b, c, e, AND f ARE THE COST COEFFICIENTS IN THE FUEL COST FUNCTION: $F_j(P_j) = a_jP_j^2 + b_jP_j + c_j + |e_j \sin(f_j \times (P_{jmin} - P_j))|$

Unit	P_{min}	P_{max}	a	b	c	e	f
1	36	114	0.0069	6.73	94.705	100	0.084
2	36	114	0.0069	6.73	94.705	100	0.084
3	60	120	0.2028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.6	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.2	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.4	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.1	801.32	300	0.035
26	254	550	0.00277	7.1	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.0114	5.35	148.89	120	0.077
31	60	190	0.0016	6.43	222.92	150	0.063
32	60	190	0.0016	6.43	222.92	150	0.063
33	60	190	0.0016	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

TABLE VI
COMPARISON OF THE EXPERIMENTAL RESULTS OBTAINED BY VARIOUS METHODS ON THE NONSMOOTH COST FUNCTION CONSIDERING THE VALVE-POINT LOADING EFFECT. FOR THE THREE-UNIT SYSTEM, IEP, EP, MPSO, AND PSO-RDL WERE ABLE TO FIND THE GLOBAL OPTIMUM [51]

Unit	GA	IEP (pop=20)	EP	MPSO (par=20)	PSO-RDL (par=20)
1	300	300.23	300.26	300.27	300.267
2	400	400	400	400	400
3	150	149.77	149.74	149.73	149.733
TP	850	850	850	850	850
TC	8237.6	8234.09	8234.07	8234.07	8234.07

TABLE VII
COMPARISON OF THE EXPERIMENTAL RESULTS OBTAINED BY VARIOUS METHODS ON THE NONSMOOTH COST FUNCTION CONSIDERING THE VALVE-POINT LOADING EFFECT. FOR THE 40-UNIT SYSTEM, PSO-RDL WAS ABLE TO FIND THE BEST SOLUTION

	CEP	FEP	MFEP	IFEP	MPSO	PSO-RDL
Minimum cost	123488.3	122679.7	122647.6	122624.35	122252.3	121468.82

TABLE VIII
COMPARISON OF METHODS ON FREQUENCY OF CONVERGENCE IN THE RANGES OF COST

	Range of Cost											
Method	127.0 - 126.5	126.5 - 126.0	126.0 - 125.5	125.5 - 125.0	125.0 - 124.5	124.5 - 124.0	124.0 - 123.5	123.5 - 123.0	123.0 - 122.5	122.5 - 122.0	122.0 - 121.5	121.5 - 121.0
CEP	10	4	-	16	22	42	4	2	-	-	-	-
FEP	6	-	4	2	10	20	26	24	6	-	-	-
MFEP	-	-	-	-	-	14	26	50	10	-	-	-
IFEP	-	-	2	-	4	4	18	50	22	-	-	-
MPSO	-	-	-	-	-	-	-	-	53	47	-	-
PSO-RDL	-	-	-	-	-	-	-	6	8	36	49	1

TABLE IX
OUTPUTS AND CORRESPONDING COSTS OF THE BEST SOLUTION OBTAINED BY PSO-RDL

Unit	P_{min}	P_{max}	Output	Cost
1	36	114	112.2886	949.880767
2	36	114	111.0704	929.604348
3	60	120	97.49443	1192.38418
4	80	190	179.7531	2143.97098
5	47	97	88.89745	724.712068
6	68	140	140	1596.46432
7	110	300	300	3216.42404
8	135	300	284.7229	2782.07788
9	135	300	284.777	2801.46883
10	130	300	130	2502.065
11	94	375	94.00612	1893.44177
12	94	375	94.03925	1909.04089
13	125	500	214.77	3792.32437
14	125	500	394.2823	6414.93466
15	125	500	304.5313	5171.47843
16	125	500	394.2847	6436.72027
17	220	500	489.2827	5296.78245
18	220	500	489.3102	5289.42926
19	242	550	511.2908	5541.17665
20	242	550	511.2941	5541.22862
21	254	550	523.2818	5071.33798
22	254	550	523.398	5073.69255
23	254	550	523.3437	5058.51899
24	254	550	523.3715	5059.07705
25	254	550	523.2815	5275.13221
26	254	550	523.28	5275.10232
27	10	150	10.00005	1140.52506
28	10	150	10.00442	1140.62574
29	10	150	10.01797	1140.93732
30	47	97	92.60281	785.447407
31	60	190	190	1643.99125
32	60	190	190	1643.99125
33	60	190	190	1643.99125
34	90	200	200	2101.01703
35	90	200	200	2043.72703
36	90	200	200	2043.72703
37	25	110	110	1220.16612
38	25	110	110	1220.16612
39	25	110	110	1220.16612
40	242	550	511.3228	5541.87129
Total Generation & Total Cost			10500	121468.82

From the experimental results, it can be observed that PSO-RDL performs quite well on the two ED problems. In particular, for the 40-unit system, we improve the known best solution to 121 468.82. According to Table VIII, PSO-RDL can be considered outperforming MPSO [48] on the 40-unit system. Based on the results of this real-world application, we can know that for constrained optimization problems, PSO-RDL can perform well and deliver good solutions.

VI. SUMMARY AND CONCLUSION

In this paper, we have studied the PSO and the linkage problem. After conducting a survey on the hybridization of particle swarm optimizers and GAs, we introduced the linkage concept, which is an important topic in GAs, to the particle swarm optimizer. In order to address the linkage problem in real-parameter optimization problems, we developed the dynamic linkage discovery technique. Furthermore, to make good use of the obtained information, we designed a recombination operator. By combining these mechanisms, we proposed a new evolutionary algorithm, which is called PSO-RDL, and conducted experiments on test functions. Finally, we applied PSO-RDL on the ED problem, which is an essential problem in power systems, and successfully obtained the currently best known result for the 40-unit system.

The work on PSO-RDL gives us two observations. First, in the literature, it is rarely discussed on the building blocks in real-parameter optimization problems. This paper may shed some light on the existence of building blocks in real-parameter optimization problems. Second, if building blocks do exist, why are these building blocks not detected by the linkage detection techniques previously proposed in the literature? According to the information obtained in this paper, perhaps in real-parameter optimization problems, at least on some of them, the configuration of building blocks dynamically changes along with the search stage. Thus, those traditional static linkage detection techniques may fail to accomplish the task of detecting linkage.

In this paper, we have introduced recombination with dynamic linkage discovery to PSO and considered integration as a promising research direction. By combining the strengths of different optimization models, we have created PSO-RDL with intriguing features and properties. We will continue to work on understanding and analyzing the properties and characteristics of real-number optimization problems in order to design better evolutionary algorithms.

ACKNOWLEDGMENT

The authors would like to thank the National Center for High-performance Computing for computer time and facilities.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.
[2] J. Kennedy and R. C. Eberhart, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine and Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.
[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[5] D. E. Goldberg, *The Design of Innovation: Lessons From and for Competent Genetic Algorithms*. Norwell, MA: Kluwer Academic, 2002.
[6] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first result," *Complex Syst.*, vol. 3, no. 5, pp. 493–530, 1989.
[7] D. E. Goldberg, K. Deb, and B. Korb, "Messy genetic algorithms revisited: Studies in mixed size and scale," *Complex Syst.*, vol. 4, no. 4, pp. 415–444, 1990.
[8] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proc. 7th Int. Conf. EP (EP VII)*, 1998, pp. 611–616.
[9] P. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," in *Proc. 7th Int. Conf. EP (EP VII)*, 1998, pp. 601–610.
[10] M. Løvbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," in *Proc. Genetic Evol. Comput. Conf.*, 2001, pp. 469–476.
[11] M. Settles and T. Soule, "Breeding swarms: A GA/PSO hybrid," in *Proc. Genetic Evol. Comput. Conf.*, 2005, pp. 161–168.
[12] D. Devicharan and C. K. Mohan, "Particle swarm optimization with adaptive linkage learning," in *Proc. IEEE Congr. Evol. Comput.*, 2004, pp. 530–535.
[13] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325–1332, Aug. 1993.
[14] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming technique for economic load dispatch," *IEEE Trans. Evol. Comput.*, vol. 7, no. 1, pp. 83–94, Feb. 2003.
[15] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proc. 7th Int. Conf. Evol. Program.*, 1998, pp. 591–600.
[16] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 1951–1957.
[17] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 1931–1938.
[18] R. C. Eberhart and X. Hu, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2002, pp. 1677–1681.
[19] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle optimizer," in *Proc. IEEE Int. Swarm Intell. Symp.*, 2005, pp. 124–129.
[20] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Particle swarm optimization algorithm with novel learning strategies," in *Proc. Int. Conf. Syst., Man Cybern.*, 2004, pp. 3659–3664.
[21] J. J. Liang, P. N. Suganthan, A. K. Qin, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
[22] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna," in *Proc. IEEE Antennas and Propag. Soc. Int. Symp. and URSI Nat. Radio Sci. Meeting*, San Antonio, TX, 2002, pp. 314–317.
[23] 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 *Proc. IEEE Congr. Evol. Comput.*, 2003, pp. 2393–2399.
[24] A. Singh, D. E. Goldberg, and Y.-P. Chen, "Modified linkage learning genetic algorithm for difficult non-stationary problems," in *Proc. Genetic Evol. Comput. Conf.*, 2002, p. 699.
[25] M. Tezuka, M. Munetomo, and K. Akama, "Linkage identification by nonlinearity check for real-coded genetic algorithms," in *Proc. Genetic Evol. Comput. Conf.*, 2004, pp. 222–233.
[26] G. R. Harik, "Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms," Illinois Genetic Algorithms Lab., Univ. Illinois at Urbana-Champaign, Urbana, IL, Tech. Rep. 97 005, 1997.
[27] K. Deb and D. E. Goldberg, "Analyzing deception in trap functions," in *Foundations of Genetic Algorithms 2*. San Mateo, CA: Morgan Kaufmann, 1994, pp. 93–108.
[28] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 522–528.
[29] *Particle Swarm Central*. [Online]. Available: http://www.particleswarm.info
[30] G. Syswerda, "Simulated crossover in genetic algorithms," in *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 239–255.
[31] A. E. Eiben, P.-E. Raué, and Z. Ruttkay, "Genetic algorithms with multi-parent recombination," in *Proc. PPSN III*, 1994, pp. 78–87.
[32] J. Smith and T. C. Fogarty, "An adaptive poly-parental recombination strategy," in *Proc. AISB-95 Workshop Evol. Comput.*, 1995, pp. 48–61.
[33] Y. Shi and R. Eberhart, "A modified particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, 1998, pp. 69–73.
[34] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.
[35] N. K. Bambha, S. S. Bhattacharyya, J. Teich, and E. Zitzler, "Systematic integration of parameterized local search into evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 137–154, Apr. 2004.
[36] B. Li and W. Jiang, "A novel stochastic optimization algorithm," *IEEE Trans. Syst., Man Cybern.*, vol. 30, no. 1, pp. 193–198, Feb. 2000.
[37] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, Univ. California at San Diego, La Jolla, 1994.
[38] 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," Nanyang Technol. Univ., Singapore, May 2005. Tech. Rep. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/indexfiles/CEC-05/Tech-Report-May-30-05.pdf
[39] G. B. Sheble and K. Brittig, "Refined genetic algorithm—Economic-dispatch example," *IEEE Trans. Power Syst.*, vol. 10, no. 1, pp. 117–124, Feb. 1995.
[40] P. H. Chen and H. C. Chang, "Large-scale economic-dispatch by genetic algorithm," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1919–1926, Nov. 1995.
[41] S. Baskar, P. Subbaraj, and M. V. C. Rao, "Hybrid real coded genetic algorithm solution to economic dispatch problem," *Comput. Electr. Eng.*, vol. 29, no. 3, pp. 407–419, May 2003.
[42] T. Yalcinoz, H. Altun, and M. Uzam, "Economic dispatch solution using a genetic algorithm based on arithmetic crossover," in *Proc. IEEE Power Tech Conf.*, Sep. 10–13, 2001, vol. 2. DOI.10.1109/PTC.2001.964734.
[43] T. Jayabarathia, K. Jayaprakasha, D. N. Jeyakumarb, and T. Raghunathan, "Evolutionary programming techniques for different kinds of economic dispatch problems," *Electr. Power Syst. Res.*, vol. 73, no. 2, pp. 169–176, 2005.
[44] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 112–117, Feb. 1996.
[45] K. P. Wong and J. Yuryevich, "Evolutionary-programming-based algorithm for environmentally-constrained economic dispatch," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 301–306, May 1998.

[46] Z. L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, Aug. 2003.

[47] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for economic dispatch with valve-point effect," *Electr. Power Syst. Res.*, vol. 71, no. 1, pp. 51–59, 2004.

[48] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42, Feb. 2005.

[49] T. A. A. Victoire and A. E. Jeyakumar, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 2121–2122, 2004.

[50] J. W. Allen and F. W. Bruce, *Power Generation, Operation, and Control*. New York: Wiley, 1984.

[51] W. M. Lin, F. S. Cheng, and M. T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 108–112, Feb. 2002.

[52] Y.-M. Park, J. R. Won, and J. B. Park, "New approach to economic load dispatch based on improved evolutionary programming," *Eng. Intell. Syst. Electr. Eng. Commun.*, vol. 6, no. 2, pp. 103–110, Jun. 1998.

[53] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 1777–1784.

Ying-Ping Chen (S'04–M'05) received the B.S. and M.S. degrees in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1995 and 1997, respectively, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 2004.

Since 2004, he has been an Assistant Professor with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. His research interests in the field of genetic and evolutionary computation include linkage learning techniques, adaptive sampling methodologies, explanatory theories, and real-world applications.

Prof. Chen is a member of the Association for Computing Machinery.

Wen-Chih Peng (M'04) was born in Hsinchu, Taiwan, R.O.C., in 1973. He received the B.S. and M.S. degrees from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 2001.

He is currently an Assistant Professor with the Department of Computer Science, National Chiao Tung University. He was previously mainly involved in projects related to mobile computing, data broadcasting, and network data management. His current research interests include mobile computing, network data management, and data mining.

Prof. Peng serves as Program Committee (PC) member in several prestigious conferences, such as the IEEE International Conference on Data Engineering, Pacific–Asia Knowledge Discovering and Mining, and Mobile Data Management.

Ming-Chung Jian received the B.Eng. degree from the National Taiwan Normal University, Taipei, Taiwan, R.O.C., and the M.Eng. degree from National Chiao Tung University, Hsinchu, Taiwan.

He is currently an Engineer with Airoha Technology Corporation, Hsinchu. His main research interests include genetic linkage learning, particle swarm optimization, and evolutionary electronics.