Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization

Ke Tang [1], Xiaodong Li [2], P. N. Suganthan [3], Zhenyu Yang [1], and Thomas Weise [1]

1 Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
`ketang@ustc.edu.cn`, `http://staff.ustc.edu.cn/~ketang`

2 School of Computer Science and Information Technology, RMIT University, Australia
`xiaodong.li@rmit.edu.au`,`http://goanna.cs.rmit.edu.au/~xiaodong`

3 School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
`epnsugan@ntu.edu.sg`, `http://www.ntu.edu.sg/home/epnsugan`

January 8, 2010

1 Introduction

In the past decades, different kinds of metaheuristic optimization algorithms [1, 2] have been developed; Simulated Annealing (SA) [3, 4], Evolutionary Algorithms (EAs) [5–7], Differential Evolution (DE) [8, 9], Particle Swarm Optimization (PSO) [10, 11], Ant Colony Optimization (ACO) [12, 13], and Estimation of Distribution Algorithms (EDAs) [14, 15] are just a few of them. These algorithms have shown excellent search abilities but often lose their efficacy when applied to large and complex problems, e.g., problem instances with high dimensions, such as those with more than one hundred decision variables.

Many optimization methods suffer from the "curse of dimensionality" [16, 17], which implies that their performance deteriorates quickly as the dimensionality of the search space increases. The reasons for this phenomenon appear to be two-fold. First, the solution space of a problem often increases exponentially with the problem dimension [16, 17] and more efficient search strategies are required to explore all promising regions within a given time budget. Second, also the characteristics of a problem may change with the scale. Rosenbrock's function [18] (see also Section 2.6), for instance, is unimodal for two dimension but becomes multimodal for higher ones [19]. Because of such a worsening of the features of an optimization problem resulting from an increase in scale, a previously successful search strategy may no longer be capable of finding the optimal solution.

Historically, scaling EAs to large-scale problems has attracted much interest, including both theoretical and practical studies. The earliest practical approach might be parallelizing an existing EA [20–22]. Later, cooperative coevolution appeared as another promising method [23, 24]. However, existing works on this topic are often limited to test problems used in individual studies and a systematic evaluation platform is still not available in literature for comparing the scalability of different EAs. This report aims to contribute to solving this problem. In particular, we provide a suite of benchmark functions for large-scale numerical optimization.

Although the difficulty of a problem generally increases with its dimensionality, it is natural that some high-dimensional problems are easier than others. For example, if the decision variables involved in a problem are independent of each other, the problem can be easily solved by decomposing it into a number of sub-problems, each of which involving only one decision variable while treating all others as constants. This way, even a line search or greedy method can solve the problem efficiently [25]. This class of problem is known as separable problems, and has been formally defined in [26] as follows:

Definition 1 *A function $f(\mathbf{x})$ is separable iff*

$$\arg\min_{(x_1,\cdots,x_n)} f(x_1,\cdots,x_n) = \left(\arg\min_{x_1} f(x_1,\cdots),\cdots,\arg\min_{x_n} f(\cdots,x_n)\right) \tag{1}$$

In other words, a function of n variables is separable if it can be rewritten as a sum of n functions of just one variable [27, 28]. If a function $f(\mathbf{x})$ is separable, its parameters x_i are called independent. Functions which are not separable are called nonseparable. Such functions can be defined as:

Definition 2 *A nonseparable function $f(\mathbf{x})$ is called m-nonseparable function if at most m of its parameters x_i are not independent. A nonseparable function $f(\mathbf{x})$ is called fully-nonseparable* [1] *function if any two of its parameters x_i are not independent.*

The definitions of separability provide us a measure of the difficulty of different problems based on which a spectrum of benchmark problems can be designed. It is maybe interesting to notice that *nonseparability* here has a similar meaning as the term *epistasis* more common in biology and in the area of discrete optimization [29–32].

In general, separable problems are considered to be easiest, while the fully-nonseparable ones usually are the most difficult problems. In between these two extreme cases, there are various kinds of partially separable problems [33–35]. Matter of fact, real-world optimization problems will most likely consist of different groups of parameters with strong dependencies within but little interaction between the groups. This issue must be reflected in benchmark problems in order to ensure that the optimization algorithms suggested by researcher based on their performance when applied to test problems are as same as efficient in practical scenarios. With this in mind, we designed our test suite in such a way that four types of high-dimensional problems are included:

1. Separable functions;
2. Partially-separable functions, in which a small number of variables are dependent while all the remaining ones are independent;
3. Partially-separable functions that consist of multiple independent subcomponents, each of which is m-nonseparable; and
4. Fully-nonseparable functions.

To produce functions which have different degrees of separability, we can first randomly divide the objective variables into several **groups**, each of which contains a number of variables. Then, for each group of variables, we can decide to either keep them independent or to make them interact with each other by using some coordinate rotation techniques [36]. Finally, a fitness function will be applied to each group of variables. For this purpose, the following six functions will be used as the basic functions:

1. The Sphere Function
2. The Rotated Elliptic Function
3. Schwefel's Problem 1.2
4. Rosenbrock's Function
5. The Rotated Rastrigin's Function
6. The Rotated Ackley's Function

All these basic functions are nonseparable except for the simple sphere function, which is often used for demonstration only. We choose these basic functions because they are the most classical examples of well-known benchmark suites [37–39] in the area of continuous optimization. Since some of these functions were separable in their original form, we applied Salomon's random coordinate rotation technique [36] to make them nonseparable. To control the separability of naturally nonseparable functions such as Schwefel's Problem 1.2 and Rosenbrock's Function, we use the sphere function to provide the separable part.

Although state-of-the-art EAs have shown satisfying performance on low-dimensional instances of these functions with, for example, 30 decision variables, the reported results for approaches that were able to handle the high-dimensional cases (e.g. consisting of 1000 or more decision variables) are still scarce. It can thus be considered to be very important to provide a benchmark suite of functions with variable dimension in order to promote the competition between researchers and, as a consequence, boost the performance of EAs in high-dimensional problems.

[1] We use "nonseparable" to indicate "fully-nonseparable" in this report if without any further explanation

Before introducing the test suite in detail, we conclude this section by summing up some key points. As listed below, this test suite consists of 20 benchmark functions. All functions are given for the special case of dimension $D = 1000$. The parameter m is used to control the number of variables in each group and hence, defining the degree of separability. We set $m = 50$ in this test suite, but the users can control this parameter conveniently for their own purposes. The test suite is an improved version of the test suite released for the CEC'2008 special session and competition on large-scale global optimization [39], which included only seven functions which were either separable or fully-nonseparable. By incorporating the partially-separable functions, the current test suite provides an improved platform for investigating the behavior of algorithms on high-dimensional problems in different scenarios.

The MATLAB and Java codes [2] of the test suite are available at

`http://nical.ustc.edu.cn/cec10ss.php`

Section 2 introduces the basic functions. The mathematical formulas and properties of these functions are described in Section 3. Finally, evaluation criteria are given in Section 4.

1. Separable Functions (3)
   (a) F_1: Shifted Elliptic Function
   (b) F_2: Shifted Rastrigin's Function
   (c) F_3: Shifted Ackley's Function
2. Single-group m-nonseparable Functions (5)
   (a) F_4: Single-group Shifted and m-rotated Elliptic Function
   (b) F_5: Single-group Shifted and m-rotated Rastrigin's Function
   (c) F_6: Single-group Shifted and m-rotated Ackley's Function
   (d) F_7: Single-group Shifted m-dimensional Schwefel's Problem 1.2
   (e) F_8: Single-group Shifted m-dimensional Rosenbrock's Function
3. $\frac{D}{2m}$-group m-nonseparable Functions (5)
   (a) F_9: $\frac{D}{2m}$-group Shifted and m-rotated Elliptic Function
   (b) F_{10}: $\frac{D}{2m}$-group Shifted and m-rotated Rastrigin's Function
   (c) F_{11}: $\frac{D}{2m}$-group Shifted and m-rotated Ackley's Function
   (d) F_{12}: $\frac{D}{2m}$-group Shifted m-dimensional Schwefel's Problem 1.2
   (e) F_{13}: $\frac{D}{2m}$-group Shifted m-dimensional Rosenbrock's Function
4. $\frac{D}{m}$-group m-nonseparable Functions (5)
   (a) F_{14}: $\frac{D}{m}$-group Shifted and m-rotated Elliptic Function
   (b) F_{15}: $\frac{D}{m}$-group Shifted and m-rotated Rastrigin's Function
   (c) F_{16}: $\frac{D}{m}$-group Shifted and m-rotated Ackley's Function
   (d) F_{17}: $\frac{D}{m}$-group Shifted m-dimensional Schwefel's Problem 1.2
   (e) F_{18}: $\frac{D}{m}$-group Shifted m-dimensional Rosenbrock's Function
5. Nonseparable Functions (2)
   (a) F_{19}: Shifted Schwefel's Problem 1.2
   (b) F_{20}: Shifted Rosenbrock's Function

[2] An algorithm may obtain different results (e.g., fitness values) with the Matlab and Java codes. This is due to the precision threshold of the double precision floating-point format. However, with the evaluation criteria given in this report, such difference will not influence the comparison between algorithms.

2 Basic Functions

2.1 The Sphere Function

The Sphere function is defined as follows:

$$F_{\textit{sphere}}(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 \tag{2}$$

where D is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). The Sphere function is very simple and is mainly used for demonstration. In this test suite this function serves as separable part when using a naturally nonseparable function to form some partially nonseparable functions.

2.2 The Rotated Elliptic Function

The original Elliptic Function is separable, and is defined as follows:

$$F_{\textit{elliptic}}(\mathbf{x}) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} x_i^2 \tag{3}$$

where D is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). The number 10^6 is called condition number, which is used to transform a Sphere function to an Elliptic function [38]. To make this function be nonseparable, an orthogonal matrix will be used to rotate the coordinates. The rotated Elliptic function is defined as follows:

$$F_{\textit{rot\_elliptic}}(\mathbf{x}) = F_{\textit{elliptic}}(\mathbf{z}),\ \mathbf{z} = \mathbf{x} * \mathbf{M} \tag{4}$$

where D is the dimension, $\mathbf{M}$ is a $D \times D$ orthogonal matrix, and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix).

2.3 The Rotated Rastrigin's Function

The original Rastrigin's function is separable, and is defined as follows:

$$F_{\textit{rastrigin}}(\mathbf{x}) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right] \tag{5}$$

where D is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). Similarly, to make it nonseparable, an orthogonal matrix is also used for coordinate rotation. The rotated Rastrgin's function is defined as follows:

$$F_{\textit{rot\_rastrigin}}(\mathbf{x}) = F_{\textit{rastrigin}}(\mathbf{z}),\ \mathbf{z} = \mathbf{x} * \mathbf{M} \tag{6}$$

where D is the dimension, $\mathbf{M}$ is a $D \times D$ orthogonal matrix, and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). Rastrigin's function is a classical multimodal problem. It is difficult since the number of local optima grows exponentially with the increase of dimensionality.

2.4 The Rotated Ackley's Function

The original Ackley's function is separable, and is defined as follows:

$$F_{\textit{ackley}}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e \tag{7}$$

where D is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix). To make it nonseparable, an orthogonal matrix is again used for coordinate rotation. The rotated Ackley's function is defined as follows:

$$F_{\textit{rot\_ackley}}(\mathbf{x}) = F_{\textit{ackley}}(\mathbf{z}),\ \mathbf{z} = \mathbf{x} * \mathbf{M} \tag{8}$$

where D is the dimension, $\mathbf{M}$ is a $D \times D$ orthogonal matrix, and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix).

2.5 Schwefel's Problem 1.2

Schwefel's Problem 1.2 is a naturally nonseparable function, which is defined as follows:

$$F_{\textit{schwefel}}(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i \right)^2 \tag{9}$$

where D is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix).

2.6 Rosenbrock's Function

Rosenbrock's function is also naturally nonseparable and is defined as follows:

$$F_{\textit{rosenbrock}}(\mathbf{x}) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right] \tag{10}$$

where $D \geq 2$ is the dimension and $\mathbf{x} = (x_1, x_2, \cdots, x_D)$ is a D-dimensional row vector (i.e., a $1 \times D$ matrix).

3 Definitions of the Benchmark Functions

3.1 Separable Functions

3.1.1 F_1: Shifted Elliptic Function

$$F_1(\mathbf{x}) = F_{\textit{elliptic}}(\mathbf{z}) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} z_i^2 \tag{11}$$

Dimension: $D = 1000$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector

Properties:

1. Unimodal
2. Shifted
3. Separable
4. Scalable
5. $\mathbf{x} \in [-100, 100]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_1(\mathbf{x}^\star) = 0$

3.1.2 F_2: Shifted Rastrigin's Function

$$F_2(\mathbf{x}) = F_{\textit{rastrigin}}(\mathbf{z}) = \sum_{i=1}^{D} \left[z_i^2 - 10\cos(2\pi z_i) + 10\right] \tag{12}$$

Dimension: $D = 1000$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector

Properties:

1. Multimodal
2. Shifted
3. Separable
4. Scalable
5. $\mathbf{x} \in [-5, 5]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_2(\mathbf{x}^\star) = 0$

3.1.3 F_3: Shifted Ackley's Function

$$F_3(\mathbf{x}) = F_{\mathit{ackley}}(\mathbf{z}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + 20 + e \tag{13}$$

Dimension: $D = 1000$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector

Properties:

1. Multimodal
2. Shifted
3. Separable
4. Scalable
5. $\mathbf{x} \in [-32, 32]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_3(\mathbf{x}^\star) = 0$

3.2 Single-group m-nonseparable Functions

3.2.1 F_4: Single-group Shifted and m-rotated Elliptic Function

$$F_4(\mathbf{x}) = F_{\textit{rot\_elliptic}}\left[\mathbf{z}(P_1 : P_m)\right] * 10^6 + F_{\textit{elliptic}}\left[\mathbf{z}(P_{m+1} : P_D)\right] \tag{14}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: a random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. Single-group m-rotated
4. Single-group m-nonseparable
5. $\mathbf{x} \in [-100, 100]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_4(\mathbf{x}^\star) = 0$

3.2.2 F_5: Single-group Shifted and m-rotated Rastrigin's Function

$$F_5(\mathbf{x}) = F_{\textit{rot\_rastrigin}}\left[\mathbf{z}(P_1 : P_m)\right] * 10^6 + F_{\textit{rastrigin}}\left[\mathbf{z}(P_{m+1} : P_D)\right] \tag{15}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: a random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. Single-group m-rotated
4. Single-group m-nonseparable
5. $\mathbf{x} \in [-5, 5]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_5(\mathbf{x}^\star) = 0$

3.2.3 F_6: Single-group Shifted and m-rotated Ackley's Function

$$F_6(\mathbf{x}) = F_{\textit{rot\_ackley}}\left[\mathbf{z}(P_1 : P_m)\right] * 10^6 + F_{\textit{ackley}}\left[\mathbf{z}(P_{m+1} : P_D)\right] \tag{16}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: a random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. Single-group m-rotated
4. Single-group m-nonseparable
5. $\mathbf{x} \in [-32, 32]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_6(\mathbf{x}^\star) = 0$

3.2.4 F_7: Single-group Shifted m-dimensional Schwefel's Problem 1.2

$$F_7(\mathbf{x}) = F_{\textit{schwefel}}\left[\mathbf{z}(P_1 : P_m)\right] * 10^6 + F_{\textit{sphere}}\left[\mathbf{z}(P_{m+1} : P_D)\right] \tag{17}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. Single-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_7(\mathbf{x}^\star) = 0$

3.2.5 F_8: Single-group Shifted m-dimensional Rosenbrock's Function

$$F_8(\mathbf{x}) = F_{\textit{rosenbrock}}\left[\mathbf{z}(P_1 : P_m)\right] * 10^6 + F_{\textit{sphere}}\left[\mathbf{z}(P_{m+1} : P_D)\right] \tag{18}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. Single-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star(P_1 : P_m) = \mathbf{o}(P_1 : P_m) + 1$, $\mathbf{x}^\star(P_{m+1} : P_D) = \mathbf{o}(P_{m+1} : P_D)$, $F_8(\mathbf{x}^\star) = 0$

3.3 $\frac{D}{2m}$-group m-nonseparable Functions

3.3.1 F_9: $\frac{D}{2m}$-group Shifted and m-rotated Elliptic Function

$$F_9(\mathbf{x}) = \sum_{k=1}^{\frac{D}{2m}} F_{\textit{rot\_elliptic}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] + F_{\textit{elliptic}}\Big[\mathbf{z}(P_{\frac{D}{2}+1} : P_D)\Big] \tag{19}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: a random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. $\frac{D}{2m}$-group m-rotated
4. $\frac{D}{2m}$-group m-nonseparable
5. $\mathbf{x} \in [-100, 100]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_9(\mathbf{x}^\star) = 0$

3.3.2 F_{10}: $\frac{D}{2m}$-group Shifted and m-rotated Rastrigin's Function

$$F_{10}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{2m}} F_{\textit{rot\_rastrigin}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] + F_{\textit{rastrigin}}\Big[\mathbf{z}(P_{\frac{D}{2}+1} : P_D)\Big] \tag{20}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{2m}$-group m-rotated
4. $\frac{D}{2m}$-group m-nonseparable
5. $\mathbf{x} \in [-5, 5]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{10}(\mathbf{x}^\star) = 0$

3.3.3 F_{11}: $\frac{D}{2m}$-group Shifted and m-rotated Ackley's Function

$$F_{11}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{2m}} F_{\textit{rot\_ackley}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] + F_{\textit{ackley}}\Big[\mathbf{z}(P_{\frac{D}{2}+1} : P_D)\Big] \tag{21}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{2m}$-group m-rotated
4. $\frac{D}{2m}$-group m-nonseparable
5. $\mathbf{x} \in [-32, 32]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{11}(\mathbf{x}^\star) = 0$

3.3.4 F_{12}: $\frac{D}{2m}$-group Shifted m-dimensional Schwefel's Problem 1.2

$$F_{12}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{2m}} F_{\textit{schwefel}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] + F_{\textit{sphere}}\Big[\mathbf{z}(P_{\frac{D}{2}+1} : P_D)\Big] \tag{22}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. $\frac{D}{2m}$-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{12}(\mathbf{x}^\star) = 0$

3.3.5 F_{13}: $\frac{D}{2m}$-group Shifted m-dimensional Rosenbrock's Function

$$F_{13}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{2m}} F_{\textit{rosenbrock}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] + F_{\textit{sphere}}\Big[\mathbf{z}(P_{\frac{D}{2}+1} : P_D)\Big] \tag{23}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{2m}$-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star(P_1 : P_{D/2}) = \mathbf{o}(P_1 : P_{D/2}) + 1$, $\mathbf{x}^\star(P_{D/2+1} : P_D) = \mathbf{o}(P_{D/2+1} : P_D)$, $F_{13}(\mathbf{x}^\star) = 0$

3.4 $\frac{D}{m}$-group m-nonseparable Functions

3.4.1 F_{14}: $\frac{D}{m}$-group Shifted and m-rotated Elliptic Function

$$F_{14}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{m}} F_{\mathit{rot\_elliptic}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] \tag{24}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. $\frac{D}{m}$-group m-rotated
4. $\frac{D}{m}$-group m-nonseparable
5. $\mathbf{x} \in [-100, 100]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{14}(\mathbf{x}^\star) = 0$

3.4.2 F_{15}: $\frac{D}{m}$-group Shifted and m-rotated Rastrigin's Function

$$F_{15}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{m}} F_{\mathit{rot\_rastrigin}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] \tag{25}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{m}$-group m-rotated
4. $\frac{D}{m}$-group m-nonseparable
5. $\mathbf{x} \in [-5, 5]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{15}(\mathbf{x}^\star) = 0$

3.4.3 F_{16}: $\frac{D}{m}$-group Shifted and m-rotated Ackley's Function

$$F_{16}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{m}} F_{\textit{rot\_ackley}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] \tag{26}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{m}$-group m-rotated
4. $\frac{D}{m}$-group m-nonseparable
5. $\mathbf{x} \in [-32, 32]^D$
6. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{16}(\mathbf{x}^\star) = 0$

3.4.4 F_{17}: $\frac{D}{m}$-group Shifted m-dimensional Schwefel's Problem 1.2

$$F_{17}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{m}} F_{\textit{schwefel}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] \tag{27}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Unimodal
2. Shifted
3. $\frac{D}{m}$-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{17}(\mathbf{x}^\star) = 0$

3.4.5 F_{18}: $\frac{D}{m}$-group Shifted m-dimensional Rosenbrock's Function

$$F_{18}(\mathbf{x}) = \sum_{k=1}^{\frac{D}{m}} F_{\textit{rosenbrock}}\Big[\mathbf{z}(P_{(k-1)*m+1} : P_{k*m})\Big] \tag{28}$$

Dimension: $D = 1000$
Group size: $m = 50$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector
$\mathbf{P}$: random permutation of $\{1, 2, \cdots, D\}$

Properties:

1. Multimodal
2. Shifted
3. $\frac{D}{m}$-group m-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o} + 1$, $F_{18}(\mathbf{x}^\star) = 0$

3.5 Nonseparable Functions

3.5.1 F_{19}: Shifted Schwefel's Problem 1.2

$$F_{19}(\mathbf{x}) = F_{\textit{schwefel}}(\mathbf{z}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i \right)^2 \tag{29}$$

Dimension: $D = 1000$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector

Properties:

1. Unimodal
2. Shifted
3. Fully-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o}$, $F_{19}(\mathbf{x}^\star) = 0$

3.5.2 F_{20}: Shifted Rosenbrock's Function

$$F_{20}(\mathbf{x}) = F_{\textit{rosenbrock}}(\mathbf{z}) = \sum_{i=1}^{D-1} \left[100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right] \tag{30}$$

Dimension: $D = 1000$
$\mathbf{x} = (x_1, x_2, \cdots, x_D)$: the candidate solution – a D-dimensional row vector
$\mathbf{o} = (o_1, o_2, \cdots, o_D)$: the (shifted) global optimum
$\mathbf{z} = \mathbf{x} - \mathbf{o}$, $\mathbf{z} = (z_1, z_2, \cdots, z_D)$: the shifted candidate solution – a D-dimensional row vector

Properties:

1. Multimodal
2. Shifted
3. Fully-nonseparable
4. $\mathbf{x} \in [-100, 100]^D$
5. Global optimum: $\mathbf{x}^\star = \mathbf{o} + 1$, $F_{20}(\mathbf{x}^\star) = 0$

4 Experimental Protocol

4.1 General Settings

1. **Problems:** 20 minimization problems
2. **Dimension:** $D = 1000$
3. **Runs/problem:** 25 (Please do *not* run multiple sets of 25 runs to pick the best set)
4. **Max_FEs:** the maximum number of function evaluations is $3.0 * 10^6$, i.e., 3e6
5. **Initialization:** Uniform random initialization within the search space
6. **Global optimum:** All problems have the global optimum within the given bounds, so there is no need to perform search outside of the given bounds for these problems. The optimum function values are 0 for all the problems.
7. **Termination:** Terminate when reaching Max_FEs.

Table 1 presents the time required for 10000 function evaluations (FEs) using the Matlab and Java versions of the test suite. The Java version was tested in a single thread on an Intel(R) Core(TM)2 Duo CPU T7500 processor with 2.20GHz in Eclipse Platform 3.4 using Java(TM) SE (build 1.6.0_16, 1 GiB maximum heap memory) for Microsoft Windows 6.0 (Vista). The Matlab version was tested in a single thread on an Intel(R) Core(TM)2 Quad CPU Q6600 with 2.40GHz in Matlab R2009a for Linux. The whole experiment with 3,000,000 FEs is thereby expected to take about 205 hours with the Matlab version and 104 hours with the Java version on a computer with similar configurations.

Table 1: Runtime of 10,000 FEs (in milliseconds) on the benchmark functions for $D = 1000$, $m = 50$.

Implementation	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
Matlab	369	566	643	646	678	754	635	535	579	886
Java	100	3461	3642	396	3621	3757	135	143	2263	5559

Implementation	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
Matlab	1086	1638	7291	7012	1115	1184	2763	9507	51893	8664
Java	6004	140	190	4324	7596	8173	144	237	84	119

4.2 Data To Be Recorded and Evaluation Criteria

Solution quality for each function when the FEs counter reaches:

- $FEs1 = 1.2e5$
- $FEs2 = 6.0e5$
- $FEs3 = 3.0e6$

The 1st(best), 13th(median) and 25th(worst) function values, mean and standard deviation of the 25 runs should be recorded.

4.3 Example of Representing the Results

Participants are requested to present their results in a tabular form, following the example given by Table 2. Competition entries will be mainly ranked based on the median results achieved when $FEs = 1.2e+5$, $6.0e+5$ and $3.0e+6$. In addition, please also provide convergence curves of your algorithm on the following eight selected functions: F_2, F_5, F_8, F_{10}, F_{13}, F_{15}, F_{18} and F_{20}. For each function, a single convergence curve should be plotted using the average results over all 25 runs.

Table 2: Experimental Results

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
$FEs = 1.2e5$	Best	0.00e+00	x.xxe+xx	x.xxe+xx	x.xxe+xx	x.xxe+xx	x.xxe+xx	x.xxe+xx
	Median							
	Worst							
	Mean							
	Std							
$FEs = 6.0e5$	Best							
	Median							
	Worst							
	Mean							
	Std							
$FEs = 3.0e6$	Best							
	Median							
	Worst							
	Mean							
	Std							

		F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
$FEs = 1.2e5$	Best							
	Median							
	Worst							
	Mean							
	Std							
$FEs = 6.0e5$	Best							
	Median							
	Worst							
	Mean							
	Std							
$FEs = 3.0e6$	Best							
	Median							
	Worst							
	Mean							
	Std							

		F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}	
$FEs = 1.2e5$	Best							
	Median							
	Worst							
	Mean							
	Std							
$FEs = 6.0e5$	Best							
	Median							
	Worst							
	Mean							
	Std							
$FEs = 3.0e6$	Best							
	Median							
	Worst							
	Mean							
	Std							

5 Beyond the CEC'2010 Special Session and Competition

This section briefly describes some thoughts that were relevant to the design of the test suite and the further usage of the test suite beyond the scope of the special session and competition at CEC'2010. Lying in the heart of the design of this test suite are two considerations:

First, the test problems must be scalable to allow researchers to carry out investigations in case of even more decision variables (e.g., 10000). In particular, scaling up the problems to even higher dimensions should not lead to overwhelming computational overhead.

Second, the test suite should cover a set of cases with different degrees of separability. This is to simulate real-world problems, in which decision variables are seldom independent on each other while dependency can often be observed though in different forms and to different extent. Existing examples can be identified from many application domains [40] such as image processing [41, 42], chemical engineering and biomedical modeling [43], engineering design optimization [44], and network optimization [45]. In the area of Genetic Programming, the size of the evolved programs or trees is usually added to the raw functional objective in order to compute the fitness [46], which could be considered to be an example for separability as well.

With the new benchmark function suite defined in this report, we continue the series of numerical optimization competitions at the CEC and contribute to bridging the gap between practitioners and algorithm developers by

1. providing a set of scalable benchmark functions suitable for examining large-scale optimization techniques and
2. defining partially separable problems which will allow us to examine optimization algorithms from a new perspective from which we assume that it comes closer to real-world situations.

For creating the m-nonseparable functions mentioned in previous sections, two options were employed: First, an inherently separable function was combined with rotated version of itself [36] and second, an inherently nonseparable function was combined with a separable one. The rotation method has the advantages that it is, without doubt, very elegant, that it can be universally applied, and that it has been used in some of the past competitions [38]. Moreover, researchers can "tune" the degree of separability of the function simply by changing the rotation matrix. Its drawback is that it requires matrix operations which scale badly and slow down the evaluation of the objective functions. In fact, using the rotation method for 1000-dimensional nonseparable functions is already very time consuming and we had to exclude it from the nonseparable function category in order to guarantee an interested participant to finish his experiments before the deadline. The combination of a nonseparable function with a separable one, as done in Section 3.2.4 with Schwefels problem 1.2 and the sphere function, is computationally more efficient. However, since the partially-separable functions generated by this approach include components of a mathematical form different from the original nonseparable ones, it might be difficult to conclude that any difference of an algorithm's performance on partially separable and nonseparable functions is caused by the degrees of separability. Instead, the reason may also be this change of mathematical form.

Given the above discussions, we provide both variants for defining partially separable benchmark functions. By doing so, we aim at providing a suite of tests which will provide both, researchers and practitioners, with a more complete picture of the performance of optimization algorithms while ensuring backward comparability to previous test scenarios. For researchers who are interested in how well their algorithms scale with the number of decision variables while placing less importance on the separability issue, we would suggest starting with the inherently nonseparable functions. Further experimental study can be carried out by using very simple and sparse matrices for rotation. For example, one can set $z_i = x_i + x_{i+1}$ for $i = 1$ to $D-1$, and $z_D = x_1 + x_D$. This way, high-dimensional nonseparable functions can be obtained at relatively low computational costs. Yet, such an approach should be used with caution since the influence of such a specific rotation matrix on the problem still remains unclear.

On the other hand, researchers that are more interested in the performance of their algorithm on problems with different degrees of separability are suggested to adhere to the rotation method used in this test suite as long as the degree of separability of interest is of medium size.

References

[1] F. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science. Norwell, MA, USA / Dordrecht, Netherlands: Kluwer Academic Publishers / Springer Netherlands, 2003, vol. 57, Series Editor Frederick S. Hillier.

[2] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*, 2nd ed. Berlin/Heidelberg: Springer-Verlag, 2004.

[3] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science Magazine*, vol. 220, no. 4598, pp. 671–680, May 13, 1983.

[4] A. Nolte and R. Schrader, "A Note on the Finite Time Behaviour of Simulated Annealing," *Mathematics of Operations Research (MOR)*, vol. 25, no. 3, pp. 476–484, Aug. 2000.

[5] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York, NY, USA: Oxford University Press, Inc., Jan. 1996.

[6] T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, ser. Computational Intelligence Library. New York, NY, USA: Oxford University Press, Inc. / Dirac House, Temple Back, Bristol, UK: Institute of Physics Publishing Ltd. (IOP) / Boca Raton, FL, USA: CRC Press, Inc., Jan. 1, 1997.

[7] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., ser. Genetic Algorithms and Evolutionary Computation. Boston, MA, USA: Springer US / Norwell, MA, USA: Kluwer Academic Publishers, 2002/2007, vol. 5.

[8] R. M. Storn and K. V. Price, "Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," International Computer Science Institute, Berkely, CA, USA, Tech. Rep. TR-95-012, 1995.

[9] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution – A Practical Approach to Global Optimization*, ser. Natural Computing Series. Basel, Switzerland: Birkhäuser Verlag / New York, NY, USA: Springer New York, 2005.

[10] R. C. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*. Piscataway, NJ, USA: IEEE Computer Society, Oct. 4–6, 1995, pp. 39–43.

[11] M. Clerc, *Particle Swarm Optimization*. London, UK: ISTE Publishing Company, Feb. 24, 2006.

[12] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[13] M. Dorigo and T. Stützle, *Ant Colony Optimization*, ser. Bradford Books. Stanford University, Camebridge, MA, USA: MIT Press, Jul. 1, 2004.

[14] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A Survey of Optimization by Building and Using Probabilistic Models," Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Computer Science, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, IlliGAL Report 99018, Sep. 1999.

[15] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms – A New Tool for Evolutionary Computation*, ser. Genetic Algorithms and Evolutionary Computation. Boston, MA, USA: Springer US / Norwell, MA, USA: Kluwer Academic Publishers, 2001, vol. 2.

[16] R. E. Bellman, *Dynamic Programming*, ser. Dover Books on Mathematics. Princeton, NJ, USA: Princeton University Press / Mineola, NY, USA: Dover Publications, 1957/2003.

[17] ——, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ, USA: Princeton University Press, 1961/1990.

[18] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, Mar. 1960.

[19] Y.-W. Shang and Y.-H. Qiu, "A Note on the Extended Rosenbrock Function," *Evolutionary Computation*, vol. 14, no. 1, pp. 119–126, Spring 2006.

[20] H. Mühlenbein, "Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization," in *Parallelism, Learning, Evolution: Workshop on Evolutionary Models and Strategies Neubiberg, Germany, 1989-03-10/11 Workshop on Parallel Processing: Logic, Organization, and Technology – Wildbad Kreuth, Germany, 1989-07-24 to 28 (WOPPLOT'89)*, ser. Lecture Notes in Computer Science (LNCS), Lecture Notes in Artificial Intelligence (LNAI, SL7), J. D. Becker, I. Eisele, and F. W. Mündemann, Eds., vol. 565/1991. Berlin, Germany: Springer-Verlag GmbH, 1991, pp. 398–406.

[21] E. Cantú-Paz, *Efficient and Accurate Parallel Genetic Algorithms*, ser. Genetic Algorithms and Evolutionary Computation. Boston, MA, USA: Springer US / Norwell, MA, USA: Kluwer Academic Publishers, Dec. 15, 2000, vol. 1.

[22] ——, "A Survey of Parallel Genetic Algorithms," Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Computer Science, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, IlliGAL Report 97003, May 1997.

[23] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling Up Fast Evolutionary Programming with Cooperative Co-Evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'01)*, vol. 2. Piscataway, NJ, USA: IEEE Computer Society, May 27–30, 2001, pp. 1101–1108.

[24] Z. Yang, K. Tang, and X. Yao, "Large Scale Evolutionary Optimization using Cooperative Coevolution," *Information Sciences*, vol. 178, no. 15, Aug. 1, 2008. [Online]. Available: http://nical.ustc.edu.cn/papers/yangtangyao%5fins.pdf

[25] Y. Davidor, "Epistasis Variance: A Viewpoint on GA-Hardness," in *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA)*, B. Spatz and G. Rawlins, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jul. 15–18, 1990, pp. 23–35.

[26] A. Auger, N. Hansen, N. Mauny, R. Ros, and M. Schoenauer, "Bio-Inspired Continuous Optimization: The Coming of Age," Piscataway, NJ, USA, Sep. 25–28, 2007, Invited Talk at CEC'2007.

[27] G. Hadley, *Nonlinear and Dynamics Programming*, ser. World Student. Reading, MA, USA: Addison-Wesley Professional, Dec. 1964.

[28] D. Ortiz-Boyer, C. Hervás-Martínez, and C. A. R. García, "CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features," *Journal of Artificial Intelligence Research (JAIR)*, vol. 24, pp. 1–48, Jul. 2005.

[29] P. C. Phillips, "The Language of Gene Interaction," *Genetics*, vol. 149, no. 3, pp. 1167–1171, Jul. 1998.

[30] W. Bateson, *Mendel's Principles of Heredity*, ser. Kessinger Publishing's® Rare Reprints. Cambridge, UK: Cambridge University Press / Whitefish, MT, USA: Kessinger Publishing, 1909/1930.

[31] T. Weise, M. Zapf, R. Chiong, and A. J. Nebro Urbaneja, "Why Is Optimization Difficult?" in *Nature-Inspired Algorithms for Optimisation*, ser. Studies in Computational Intelligence, R. Chiong, Ed. Springer-Verlag, Apr. 30, 2009, vol. 193/2009, ch. 1, pp. 1–50.

[32] L. Altenberg, "NK Fitness Landscapes," in *Handbook of Evolutionary Computation*, ser. Computational Intelligence Library, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. Oxford University Press, Inc. / Institute of Physics Publishing Ltd. (IOP) / CRC Press, Inc., Jan. 1, 1997, ch. B2.7.2.

[33] A. Griewank and P. L. Toint, "Partitioned Variable Metric Updates for Large Structured Optimization Problems," *Numerische Mathematik*, vol. 39, no. 1, pp. 119–137, Feb. 1982.

[34] ——, "Local Convergence Analysis for Partitioned Quasi-Newton Updates," *Numerische Mathematik*, vol. 39, no. 3, pp. 429–448, Oct. 1982.

[35] B. Colson and P. L. Toint, "Optimizing Partially Separable Functions without Derivatives," *Optimization Methods and Software*, vol. 20, no. 4 & 5, pp. 493–508, Aug. 2005.

[36] R. Salomon, “Re-Evaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions. A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms,” *Biosystems*, vol. 39, no. 3, pp. 263–278, 1996.

[37] X. Yao, Y. Liu, and G. Lin, “Evolutionary Programming Made Faster,” *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[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 Technological University (NTU), Singapore, Tech. Rep. May-30-05, May 30, 2005. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/index%5ffiles/CEC-05/Tech-Report-May-30-05.pdf

[39] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, “Benchmark Functions for the CEC’2008 Special Session and Competition on Large Scale Global Optimization,” University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Héféi, Ānhuī, China, Tech. Rep., 2007. [Online]. Available: http://nical.ustc.edu.cn/cec08ss.php

[40] A. R. Conn, N. I. M. Gould, and P. L. Toint, “An Introduction to the Structure of Large Scale Nonlinear Optimization Problems and the LANCELOT Project,” in *Computing Methods in Applied Sciences and Engineering – Proceedings of the Ninth International Conference on Computing Methods in Applied Sciences and Engineering*, ser. Proceedings in Applied Mathematics Series, R. Glowinski and A. Lichnewsky, Eds., vol. 45. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), Jan. 29–Feb. 2, 1990, pp. 42–51.

[41] S. Ahn, J. A. Fessler, D. Blatt, and A. O. Hero, “Convergent Incremental Optimization Transfer Algorithms: Application to Tomography,” *IEEE Transactions on Medical Imaging*, vol. 25, no. 3, pp. 283–296, Mar. 2006.

[42] J. Koljonen, “Partially Separable Fitness Function and Smart Genetic Operators for Surface-Based Image Registration,” in *AI and Machine Consciousness – Proceedings of the 13th Finnish Artificial Intelligence Conference (STeP’08)*, ser. Publications of the Finnish Artificial Intelligence Society, T. Raiko, P. Haikonen, and J. Väyrynen, Eds., vol. 24. Helsinki, Finland: Nokia Research Center / Vantaa, Finland: Finnish Artificial Intelligence Society (FAIS), Aug. 20–22, 2008, pp. 7–14.

[43] J. J. Moré, “A Collection of Nonlinear Model Problems,” in *Computational Solution of Nonlinear Systems of Equations – Proceedings of the 1988 SIAM-AMS Summer Seminar on Computational Solution of Nonlinear Systems of Equations*, ser. Lectures in Applied Mathematics (LAM), E. L. Allgower and K. Georg, Eds., vol. 26. Providence, RI, USA: American Mathematical Society (AMS) Bookstore, Jul. 18–29, 1988, pp. 723–762.

[44] R. S. Krishnamachari and P. Y. Papalambros, “Hierarchical Decomposition Synthesis in Optimal Systems Design,” *Journal of Mechanical Design*, vol. 119, no. 4, pp. 448–457, Dec. 1997.

[45] A. J. Osiadacz and D. J. Bell, “Optimization Techniques for Large Networks: Gas and Water,” in *Simulation and Optimization of Large Systems – Based on the Proceedings of a Conference Organized by the Institute of Mathematics and its Applications on Simulation and Optimization of Large Systems, Held at the University of Reading, in September 1986*, ser. Institute of Mathematics and Its Applications Conference Series, A. J. Osiadacz, Ed. Oxford, UK: Clarendon Press (Oxford University Press), 1988, vol. 13, pp. 175–192.

[46] T. Weise, *Global Optimization Algorithms – Theory and Application.* Germany: it-weise.de (self-published), 2009. [Online]. Available: http://www.it-weise.de/