Research Article

Complexity Reduction in the Use of Evolutionary Algorithms to Function Optimization: A Variable Reduction Strategy

Guohua Wu,^{1,2} Witold Pedrycz,^{2,3} Haifeng Li,^{4,5} Dishan Qiu,¹ Manhao Ma,¹ and Jin Liu¹

¹ Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, 47 Yanzheng Street, Changsha, Hunan 410073, China

² Department of Electrical & Computer Engineering, University of Alberta, Edmonton, AB, Canada T6R 2V4

³ Warsaw School of Information Technology, Newelska 6, Warsaw, Poland

⁴ School of Civil Engineering and Architecture, Central South University, Changsha, Hunan 410004, China

⁵ School of Geosciences and Info-Physics, Central South University, Changsha, Hunan, China

Correspondence should be addressed to Haifeng Li; lihaifeng@csu.edu.cn

Received 13 August 2013; Accepted 8 September 2013

Academic Editors: P. Bala and P. K. Egbert

Copyright © 2013 Guohua Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Discovering and utilizing problem domain knowledge is a promising direction towards improving the efficiency of evolutionary algorithms (EAs) when solving optimization problems. We propose a knowledge-based variable reduction strategy (VRS) that can be integrated into EAs to solve unconstrained and first-order derivative optimization functions more efficiently. VRS originates from the knowledge that, in an unconstrained and first-order derivative optimization function, the optimal solution locates in a local extreme point at which the partial derivative over each variable equals zero. Through this collective of partial derivative equations, some quantitative relations among different variables can be obtained. These variable relations have to be satisfied in the optimal solution. With the use of such relations, VRS could reduce the number of variables and shrink the solution space when using EAs to deal with the optimization function, thus improving the optimizing speed and quality. When we apply VRS to optimization problems, we just need to modify the calculation approach of the objective function. Therefore, practically, it can be integrated with any EA. In this study, VRS is combined with particle swarm optimization variants and tested on several benchmark optimization functions and a real-world optimization problem. Computational results and comparative study demonstrate the effectiveness of VRS.

1. Introduction

Optimization, including continuous optimization and discrete optimization, plays an important role in scientific research, management, industry, and so forth, given the fact that many problems in the real world are essentially optimization tasks. Evolutionary algorithms (EAs), such as genetic algorithms (GAs), ant colony optimization (ACO), and particle swarm optimization (PSO), have shown competitive performance when solving complex and large-scale optimization problems. To improve the efficiency of EAs, two aspects are important and deserve investigation. The first one is the search capability of the used EA itself, including both the exploitation and exploration capabilities. The other one is how to effectively integrate

domain knowledge about the optimization problem into EAs [1].

Previously, more attention was paid to the design of generic EA variants for higher search capability. Take PSO as an example. In the last decades, many enhanced PSO versions were developed, such as comprehensive learning PSO [2], mimetic fitness Euclidean-distance PSO [3], orthogonal learning PSO [4], and PSO with local search [5]. According to no free lunch theory [6], no algorithm will be effective for all optimization problems. It is therefore hard to design an efficient EA that is suitable to all kinds of optimization problems. However, if we can make use of some valuable domain knowledge implied in optimization problems, we may improve the efficiency of EAs by reducing the complexity of the optimization problems.

In the area of discrete optimization, the problem domain knowledge has started to attract researchers' attention. For instance, the incorporation of knowledge-based strategies into the heuristics of swarm optimization has demonstrated to be effective [7]. Note that the problem domain knowledge in discrete optimization (e.g., the scheduling problem [8, 9] and spatial geoinformation services composition problem [10, 11]) is dependent on concrete problems considered and the knowledge extraction and discovery process is relatively subjective.

In comparison, in the area of continuous optimization, such as function optimization, the problem domain knowledge is seldom considered. One may consider the integration of PSO with the gradient-based search technique as an instance that the problem domain knowledge is combined into EAs [12], as the gradient-based search technique utilizes the gradient information implied in the optimization problem. This gradient-based search technique is used to guide the search direction of EAs. It is believed that there should be some relations among different variables on the optima of an optimization problem. Particularly, in [1], the problem domain knowledge of variable symmetry was formulated. Based on this knowledge, an inner variable learning (IVL) strategy was proposed and incorporated into PSO, thus a new PSO variant named PSO-IVL was developed. PSO-IVL demonstrates the effectiveness and potential of the integration of problems domain knowledge into EAs. However, it is not a generic algorithm because it is only suitable to optimization problems with symmetric variables.

Therefore, we may wonder whether there exist more general methods for finding relations among different variables of an optimization problem; then we can utilize the knowledge of such relations to simplify the original optimization problems and improve the efficiency of EAs. As we know, the variable relation reflects the variable dependence, which can shrink the solution space of original optimization problems.

Driven by this motivation, we investigate a method to discover underlying variable relations existing in an unconstrained and first-order derivative optimization function. We find that the discovered variable relations can be used effectively to reduce the number of variables included in the optimization function when applying PSO (other EAs should be suitable as well) to that function optimization problem. Consequently a variable reduction strategy (VRS) is developed and integrated into the PSO variants. Experimental tests on some benchmark optimization functions and a real-world optimization problem demonstrate that VRS can reduce the complexity of the optimization functions and help PSO to find high-quality solutions more efficiently.

2. Variable Reduction Strategy

Assume that $f(x_1, x_2, \dots, x_D)$ has a first-order derivative. For the corresponding unconstrained optimization problem

$\min f(x_1, x_2, \dots, x_D)$, the optimal solution arises from the relationships

$$\begin{aligned} \frac{\partial f(x_1, x_2, \dots, x_D)}{\partial x_1} &= 0, \\ \frac{\partial f(x_1, x_2, \dots, x_D)}{\partial x_2} &= 0, \\ &\vdots \\ \frac{\partial f(x_1, x_2, \dots, x_D)}{\partial x_D} &= 0. \end{aligned} \quad (1)$$

It may sometimes be difficult to solve the equations above to obtain exact values of their variables. There are two reasons for this. The first one is that the equations may be nonlinear and complex which makes it difficult to get the completely analytic solution. The second one is that there might be many extreme points for a multimodal optimization function; that is to say, solutions of the equations related to such an optimization function are not unique. However, some quantitative and explicit relations among variables could be determined from (1). We do not have to find out all the relations among variables, since if we can obtain just some variable relations from (1), we can reduce the number of variables and shrink the solution space, thus, decreasing the complexity of the original optimization function.

For example, if from (1) we can form a relation described as

$$x_i = g(\{x_j \mid i = 1, \dots, m, j \neq i\}), \quad 1 \leq i \leq D, 1 \leq m \leq D, \quad (2)$$

we say x_i can be expressed by $\{x_j \mid i = 1, \dots, m, j \neq i\}$. This variable relation has to be satisfied in the optimal solution. Under this condition, in the course of using EA (e.g., PSO) to solve the optimization function, the value of x_i can be calculated directly from (2) and the values of variables in $\{x_j \mid i = 1, \dots, m, j \neq i\}$. As a result, in the problem solving process, variable x_i can be reduced.

To give an intuitive illustration, let us now consider optimization problem $\min f = x_1^2 + x_1 \cdot \sin x_2$ as an example. The solution space of this optimization problem is illustrated in Figure 1(a). Let the derivative of the optimization function be equal to zero; we can obtain the following:

$$2x_1 + \sin x_2 = 0, \quad (3)$$

$$x_1 \cdot \cos x_2 = 0. \quad (4)$$

From (3), we get the relation $x_1 = -0.5 \sin x_2$. With this relation, variable x_1 can be reduced and the original optimization problem is changed into $\min f = -0.25(\sin x_2)^2$. The solution space of the optimization problem after variable reduction is changed accordingly as displayed in Figure 1(b). As a result, the original two-variable optimization problem is transformed into a one-variable optimization problem after variable reduction. In addition, the solution space is shrunk from two dimensions to one dimension. Therefore,

FIGURE 1: Illustration of an example of variable reduction strategy. (a) Solution space of the original problem; (b) solution space of the problem after variable reduction.

with variable reduction, the complexity of this optimization problem is reduced noticeably.

It is clear that if more variable relations can be found, then more variables can be reduced. Let us introduce several essential definitions.

- (i) Core variable: the variable is used to represent other variables.
- (ii) Reduced variable: the reduced variable can be represented by core variables.
- (iii) Optimization variable core: the collection of all core variables present in an optimization function. We denote it by $C = \{x_i \mid i = 1, \dots, n\}$, $1 \leq n \leq D$.

Obviously, the fewer core variables and the more reduced variables we obtain, the more the complexity of the original optimization function will be eliminated. Therefore, the task of the variable reduction strategy (VRS) is to obtain an optimization variable core with the minimum cardinality. To obtain a general theory and method for finding minimum core variables is still an open problem, since the equations described in (1) may be too complex. However, at least, we can safely conclude that if a variable in a derivative optimization problem is less than or equal to third-order, and this variable can be reduced.

In general, the performance of EAs degrades noticeably with the dimensionality increase of an optimization problem. VRS can help alleviate the problem.

3. Experimental Study on Benchmark Optimization Problems

3.1. Experimental Setting. VRS is integrated into the basic version of PSO [13] to obtain a new PSO variant called PSO-VRS. We apply PSO-VRS to several benchmark optimization functions to test the effectiveness of VRS. The Rosenbrock

function [2], variably dimensioned function [14], Wood function [14], and Ackley function [2] are selected as the benchmark optimization functions. Each function was optimized by some state-of-the-art PSO variants as well as PSO-VRS. We present the details of the variable reduction procedure for each function, provide the computational results of each algorithm, and demonstrate the evolutionary process of PSO-VRS. The PSO variants used in the comparative study are listed below:

- (i) PSO with inertia weight (PSO-w) [15];
- (ii) PSO with constriction factor (PSO-cf) [16];
- (iii) UPSO [17];
- (iv) fully informed particle swarm (FIPS) [18];
- (v) FDR-PSO [19];
- (vi) CPSO-H [20];
- (vii) CLPSO [2];
- (viii) PSO-VRS.

It should be noted that the parameter settings of PSO-w, PSO-cf, UPSO, FDR, FIPS, CPSO-H, and CLPSO are the same as those in [2]. Related parameters of PSO-VRS are set up as follows: the inertia weight $w = 0.9 - (0.5 \cdot g / \max \text{Gen})$, the maximum function evaluations $\max \text{FEs} = 200000$, acceleration coefficients $c_1 = c_2 = 2.0$, and number of particles $ps = 20$.

3.2. Variable Reduction Process of Test Optimization Functions

3.2.1. Rosenbrock Function. This function is formulated as $f(\mathbf{x}) = \sum_{i=1}^D (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$, which is multimodal, is nonseparable, and exhibits a very narrow valley moving

from local optimum to global optimum [21]. Let us rewrite the function as follows:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right) \\ &= 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + 100(x_2^2 - x_3)^2 \\ &\quad + (x_2 - 1)^2 + 100(x_3^2 - x_4)^2 + (x_3 - 1)^2 \\ &\quad + \cdots + 100(x_{n-1}^2 - x_n)^2 + (x_{n-1} - 1)^2. \end{aligned} \quad (5)$$

Then setting the partial derivatives to zero, we have

$$\frac{df}{dx_1} = 200(x_1^2 - x_2) \cdot 2x_1 + 2(x_1 - 1) = 0, \quad (6)$$

$$\begin{aligned} \frac{df}{dx_i} &= -200(x_{i-1}^2 - x_i) + 200(x_i^2 - x_{i+1}) \cdot 2x_i \\ &\quad + 2(x_i - 1) = 0, \quad 2 < i < n, \end{aligned} \quad (7)$$

$$\frac{df}{dx_n} = -200(x_{n-1}^2 - x_n) = 0. \quad (8)$$

From (6), we have

$$x_2 = x_1^2 + \frac{x_1 - 1}{200x_1}. \quad (9)$$

Subsequently from (7), we have

$$x_i = x_{i-1}^2 + \frac{(x_{i-1} - 1) - 100(x_{i-2}^2 - x_{i-1})}{200x_{i-1}}, \quad 2 < i < n. \quad (10)$$

Furthermore,

$$x_n = x_{n-1}^2. \quad (11)$$

We can observe from the above expressions (9)–(11) that any other variables in the Rosenbrock function can be calculated with the use of x_1 , such that the objective function can be calculated only with the aid of the value of x_1 and (9)–(11). Therefore x_1 is the one and only one core variable of this optimization function. As a result, the original multivariable optimization problem is actually transformed into a one-variable optimization problem. The Rosenbrock function with 10 variables was optimized by each PSO variant. The search range of each variable is taken from -3 to 3 .

3.2.2. Variably Dimensioned Function. This function is described as $f(x) = \sum_{i=1}^n (x_i - 1)^2 + (\sum_{i=1}^n i(x_i - 1))^2 + (\sum_{i=1}^n i(x_i - 1))^4$. With regard to this function, we obtain

$$\begin{aligned} \frac{df}{dx_i} &= 2(x_i - 1) + 2i \left(\sum_{i=1}^n i(x_i - 1) \right) \\ &\quad + 4i \left(\sum_{i=1}^n i(x_i - 1) \right)^3 = 0, \quad 1 \leq i \leq n. \end{aligned} \quad (12)$$

According to (12), for any two variables x_i and x_k we have

$$\begin{aligned} 2(x_i - 1) + 2i \left(\sum_{i=1}^n i(x_i - 1) \right) &= -4i \left(\sum_{i=1}^n i(x_i - 1) \right)^3, \\ 2(x_k - 1) + 2k \left(\sum_{i=1}^n i(x_i - 1) \right) &= -4k \left(\sum_{i=1}^n i(x_i - 1) \right)^3. \end{aligned} \quad (13)$$

From (13), we get

$$k(x_i - 1) = i(x_k - 1). \quad (14)$$

In the sequel, from (14), we can obtain $x_i = (ix_k - i + k)/k$.

Let $k = 1$, then we have $x_i = ix_1 - i + 1$. As a result all other variables can be represented by x_1 , which is the core variable of this optimization function.

3.2.3. Wood Function. This function comes in the form

$$\begin{aligned} f(x) &= \left(10(x_2 - x_1^2) \right)^2 + (1 - x_1)^2 + \left((90)^{1/2} (x_4 - x_3^2) \right)^2 \\ &\quad + (1 - x_3)^2 + \left((10)^{1/2} (x_2 + x_4 - 2) \right)^2 \\ &\quad + \left((10)^{-1/2} (x_2 - x_4) \right)^2. \end{aligned} \quad (15)$$

We determine the related derivative as shown below:

$$\begin{aligned} \frac{df}{dx_1} &= 200(x_2 - x_1^2)(-2x_1) - 2(1 - x_1) = 0, \\ \frac{df}{dx_2} &= 200(x_2 - x_1^2) + 20.2x_2 + 19.8x_4 - 40 = 0, \\ \frac{df}{dx_3} &= 180(x_4 - x_3^2)(-2x_3) - 2(1 - x_3) = 0, \\ \frac{df}{dx_4} &= 180(x_4 - x_3^2) + 19.8x_2 + 20.2x_4 - 40 = 0. \end{aligned} \quad (16)$$

From (16), one has

$$x_2 = \frac{x_1 - 1}{200x_1} + x_1^2, \quad (17)$$

$$x_4 = \frac{1000(x_2 - x_1^2) + 101x_2 - 200}{-99}, \quad (18)$$

$$x_4 = \frac{x_3 - 1}{180x_3} + x_3^2, \quad (19)$$

$$x_3 = \sqrt{\frac{1001x_4 + 99x_2 - 200}{900}}. \quad (20)$$

We can see from (17)–(20) that x_2 can be calculated from x_1 ; x_4 can be calculated with the use of both x_2 and x_1 ; and x_3 can be calculated based on x_4 and x_2 . In fact, all x_4 , x_3 , and x_2 can be computed from x_1 , which is therefore the only core

TABLE 1: Optimization results obtained for the test functions; the best results are shown in boldface.

Algorithm	Mean	Std	FEs	Mean	Std	FEs
Rosenbrock function			Variably dimensioned function			
PSO-w	2.182e + 000	2.642e - 002	30 000	1.576e + 003	4.610e + 002	200 000
PSO-cf	3.833e - 001	0.250e + 000	30 000	5.618e + 000	1.385e + 001	200 000
UPSO	1.055e + 000	1.542e + 000	30 000	1.164e + 003	4.763e + 002	200 000
FDR	6.480e - 001	6.846e - 001	30 000	3.188e - 001	5.820e - 001	200 000
FIPS	1.186e + 000	1.080e - 001	30 000	4.479e + 003	1.317e + 003	200 000
CPSO-H	1.044e + 000	1.265e + 000	30 000	1.365e + 004	3.277e + 003	200 000
CLPSO	1.262e + 000	9.658e - 001	30 000	2.889e + 003	6.235e + 002	200 000
PSO-VRS	0.0	0.0	11 565	0.0	0.0	12 684
Algorithm	Wood function			Ackley function		
PSO-w	3.347e - 006	3.383e - 006	200 000	3.942e - 014	1.123e + 000	200 000
PSO-cf	9.784e - 006	2.225e - 007	200 000	1.123e + 000	8.655e - 001	200 000
UPSO	1.459e - 009	5.144e - 009	200 000	1.225e - 015	3.162e - 015	200 000
FDR	1.576e - 009	1.589e - 009	200 000	2.844e - 014	4.107e - 015	200 000
FIPS	5.868e - 005	7.834e - 005	200 000	4.812e - 007	9.172e - 008	200 000
CPSO-H	5.861e - 001	2.616e - 001	200 000	4.931e - 014	1.104e - 014	200 000
CLPSO	1.570e - 002	3.330e - 002	200 000	0.0	0.0	180 864
PSO-VRS	0.0	0.0	13 078	0.0	0.0	11 254

variable of this optimization function. Note that, in the search process of PSO-VRS, the value of $(1001x_4 + 99x_2 - 200)/900$ in (20) may be smaller than zero. Under this condition, the function fitness will be penalized and equal to 1000, to drive the solution to the feasible area.

3.2.4. Ackley Function. This function reads as $f_5(x) = -20 \cdot \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) + 20 - \exp((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + e$.

Regarding this function, the partial derivatives are

$$\begin{aligned}
 \frac{df}{dx_j} &= -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) \\
 &\quad \times (-0.2) \frac{1}{2} \frac{1}{\sqrt{(1/n) \sum_{i=1}^n x_i^2}} \frac{1}{n} 2x_j \\
 &\quad - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \frac{1}{n} 2\pi \sin(2\pi x_j) = 0, \\
 \frac{df}{dx_k} &= -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) \\
 &\quad \times (-0.2) \frac{1}{2} \frac{1}{\sqrt{(1/n) \sum_{i=1}^n x_i^2}} \frac{1}{n} 2x_k \\
 &\quad - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \frac{1}{n} 2\pi \sin(2\pi x_k) = 0.
 \end{aligned} \tag{21}$$

From (21), we obtain the following relationship:

$$\frac{x_j}{x_k} = \frac{\sin(2\pi x_j)}{\sin(2\pi x_k)}. \tag{22}$$

Such that we have $x_j = x_k$. Therefore, variable x_1 can be the only core variable.

3.3. Computational Results and Comparative Study. To solve each test optimization function, each algorithm was run 30 times. The corresponding computational results are listed in Table 1, where Mean is the mean fitness value of 30 runs; Std is the related standard deviation of the results; FEs is the mean function evaluations to obtain the results. The evolutionary process of the best fitness value of each function obtained by PSO-VRS is displayed in Figure 2.

From the results about the Rosenbrock function in Table 1, we can find that it is generally difficult for other PSO variants to find an optimal or near-optimal solution of the Rosenbrock function. However, with the aid of VRS, PSO-VRS can use the basic PSO to efficiently find the optimal solution averagely within about 12000 fitness evaluations. Moreover, Figure 2(a) demonstrates that VRS enables PSO to converge to high-quality solutions quickly.

Results of the variably dimensioned function listed in Table 1 reveal that it is hard for most of the comparative PSO variants without the integration of VRS to generate a solution of high quality. In contrast, PSO-VRS always produces the optimal solution in relatively small number of function evaluations (averagely about 12684). Figure 2(b) demonstrates that PSO-VRS converges to the optimal solution quickly. The variably dimensioned function is highly related, which brings a great deal of difficulty to typical PSO variants to

FIGURE 2: The evolutionary process of PSO-VRS on test functions, (a) Rosenbrock function, (b) variably dimensioned function, (c) Wood function, and (d) Ackley function.

form satisfactory solutions. On the other hand, VRS utilizes the underlying variable relations and translates the original problem into a one-variable optimization task. This indicates that VRS significantly reduces the complexity of the variably dimensioned function.

We can see from Table 1 that, compared with other comparative PSO variants, PSO-VRS generates the best result (also the optimal) of the Wood function averagely within no more than 13100 function evaluations. Though the variable relations described in (17)–(20) seem more complex, they well support PSO to find the optimal solution. Figure 2(c) underlines that PSO-VRS exhibits high convergence.

It also can be observed from Table 1 that both CLPSO and PSO-VRS could always find the best results, compared to other PSO variants. The superiority of PSO-VRS to CLPSO is that PSO-VRS can obtain the optimal solution at the cost of much less function evaluations. The high convergence capability of PSO-VRS when solving the Ackley function is shown in Figure 2(d).

4. Experimental Study on a Real-World Optimization Problem

Frequency-modulated (FM) sound wave synthesis has an important role in several modern music systems and to optimize the parameter of an FM synthesizer is a six-dimensional optimization problem where the vector to be optimized is represented by $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$ [22, 23]. This problem is a highly complex multimodal one having strong epistasis with minimum value $f(X^*) = 0$ [24]. This problem was frequently solved by EAs or taken as a benchmark real-world optimization problem to test the performance of new EA variants [25, 26]. This optimization problem is formulated as follows [22]:

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta) + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta)),$$

TABLE 2: Optimization results obtained for the frequency-modulated synthesizer optimization problem; the results obtained by EAs with VRS are shown in boldface.

Algorithm	Best	Worst	Mean	Std
PSO-w	0.0	1.258023e + 001	3.752319e + 000	5.853543e + 000
PSO-w-VRS	0.0	1.254014e + 001	3.375620e + 000	5.893543e + 000
PSO-cf	2.673488e - 006	2.180154e + 001	1.200588e + 001	5.144827e + 000
PSO-cf-VRS	4.275136e - 007	1.217917e + 001	8.466443e + 000	4.848450e + 000
UPSO	5.839747e - 028	1.473172e + 001	6.608215e + 000	4.636487e + 000
UPSO-VRS	0.0	1.120719e + 001	6.425027e + 000	4.185174e + 000
FDR	0.0	2.016705e + 001	1.191916e + 001	6.193037e + 000
FDR-VRS	0.0	1.254068e + 001	1.055711e + 001	3.626518e + 000
FIPS	3.321497e - 003	8.500601e + 000	2.306348e + 000	2.513825e + 000
FIPS-VRS	9.388098e - 010	2.787766e + 000	1.127794e + 000	1.359315e + 000
CPSO-H	1.179023e + 001	2.553845e + 001	1.910668e + 001	4.381173e + 000
CPSO-H-VRS	7.783981e - 001	1.683571e + 001	1.274518e + 001	5.058687e + 000
CLPSO	1.354023e - 004	1.138270e + 001	1.666438e + 000	3.567321e + 000
CLPSO-VRS	2.309327e - 005	6.541305e + 000	3.734962e - 001	1.456388e + 000

$$\begin{aligned}
y_0(t) = & (1.0) \\
& \cdot \sin((5.0) \cdot t \cdot \theta - (1.5) \\
& \cdot \sin((4.8) \cdot t \cdot \theta + (2.0) \cdot \sin((4.9) \cdot t \cdot \theta))).
\end{aligned} \quad (23)$$

The objective function is

$$\min f(X) = \sum_{t=0}^{100} (y(t) - y_0(t))^2. \quad (24)$$

To use the variable reduction strategy, we let the derivative of the objective function on variable a_1 equal to zero and obtain

$$\sum_{t=0}^{100} 2 \cdot (a_1 \cdot \phi(t) - y_0(t)) \cdot \phi(t) = 0, \quad (25)$$

where $y(t) = a_1 \cdot \phi(t)$, $\phi(t) = \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta)))$.

From (25), we have

$$a_1 = \frac{\sum_{t=0}^{100} y_0(t) \cdot \phi(t)}{\sum_{t=0}^{100} \phi(t) \cdot \phi(t)}. \quad (26)$$

According to (26), variable a_1 can be calculated from other five variables. Therefore, a_1 is the reduced variable and collection $\{\omega_1, a_2, \omega_2, a_3, \omega_3\}$ is the corresponding optimization variable core. With VRS, the solution space is shrunk from six dimensions to five dimensions.

To evaluate the impact of VRS on this optimization problem, we use PSO-w, PSO-cf, UPSO, FDR, FIPS, CPSO-H, and CLPSO with and without the integration of VRS to solve the problem, respectively. Each algorithm is run for 30 times.

From Table 2, we can discover that, for every PSO variant, the results produced by the algorithm with the integration of VRS are better than that produced by the algorithm without

the integration of VRS. Particularly, this improvement is more significant when FIPS and CLPSO are taken as the solvers. These results demonstrate the potential application of VRS to real-world optimization problems. Moreover, in this optimization problem, only one variable is reduced, which indicates that when we apply VRS to optimization problems, we do not have to find all the quantitative variable relations and reduce major variables, even the reduction of small number of variables could be beneficial and improve the efficiency of EAs.

5. Conclusions

The utilization of the domain knowledge associated with the optimization problem can reduce the complexity of the original problem and facilitate the solution search of EAs. In this study, we investigate the underlying knowledge of quantitative variable relations that have to be satisfied in the optimal solutions of an unconstrained and first-order derivative optimization function. Based on these relations, we propose a variable reduction strategy (VRS). The essence of VRS is to find an optimization variable core with the minimum number of core variables. Computational results and comparative studies carried out for several test benchmark optimization functions and a real-life optimization problem demonstrate that VRS can significantly improve the efficiency of PSO variants. Currently, we cannot guarantee that VRS can be applied to any unconstrained optimization problems. However, it could be beneficial to check the variable relations and use VRS when using EAs to solve unconstrained optimization problems. VRS is expected to have large application potential in real-world optimization problems.

Some future researches can be carried in four directions. Although the variable reduction strategy is generic and effective, on some occasions, it might be very difficult to obtain the variable relations from the partial derivatives of an optimization function because of the complexity of

these derivatives. To construct a solid and comprehensive theory of variable reduction, we may investigate whether there are some generic and formal theories about finding the explicit variable relations through a group of equations. Furthermore, we also consider formulating the underlying variable relations by some approximate methods, such as neural network. The third direction of research is to further study the variable reduction strategy to apply it to constrained optimization problems. The fourth one is to test the efficiency and effectiveness of the variable reduction strategy with regard to more real-world optimization problems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC, 51178193 and 41001220). Guohua Wu is supported by the China Scholarship Council under Grant no. 201206110082. The authors thank Dr. Ponnuthurai Nagarathnam Suganthan for providing them with the source codes of the comparative algorithms and giving them insightful suggestion and comments.

References

- [1] G. Wu, W. Pedrycz, M. Ma, D. Qiu, and H. Li, "A particle swarm optimization variant with an inner variable learning strategy," *The Scientific World Journal*. In press.
- [2] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [3] J. J. Liang, B. Y. Qu, S. T. Ma, and P. N. Suganthan, "Memetic fitness Euclidean-distance particle swarm optimization for multi-modal optimization," in *Bio-Inspired Computing and Applications*, vol. 6840 of *Lecture Notes in Computer Science*, pp. 378–385, 2012.
- [4] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832–847, 2011.
- [5] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences*, vol. 197, pp. 131–143, 2012.
- [6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [7] L. N. Xing, Y. W. Chen, P. Wang, Q. S. Zhao, and J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," *Applied Soft Computing Journal*, vol. 10, no. 3, pp. 888–896, 2010.
- [8] G. Wu, M. Ma, J. Zhu, and D. Qiu, "Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks," *Journal of Systems Engineering and Electronics*, vol. 23, pp. 723–733, 2012.
- [9] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers & Operations Research*, vol. 40, no. 3, pp. 1884–1894, 2013.
- [10] H. Li and B. Wu, "Adaptive geo-information processing service evolution: reuse and local modification method," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 165–183, 2013.
- [11] H. Li, Q. Zhu, X. Yang, and L. Xu, "Geo-information processing service composition for concurrent tasks: a QoS-aware game theory approach," *Computers and Geosciences*, vol. 47, pp. 46–59, 2012.
- [12] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 353–359, 2012.
- [13] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.
- [14] S. K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.
- [15] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.
- [16] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [17] K. Parsopoulos and M. Vrahatis, "UPSO: a unified particle swarm optimization scheme," in *Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE '04)*, vol. 1 of *Lecture Series on Computer and Computational Sciences*, pp. 868–873, 2004.
- [18] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [19] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 174–181, April 2003.
- [20] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [21] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep., Nanyang Technological University, Singapore, 2005.
- [22] S. Das and P. Suganthan, "Problem definitions and evaluation criteria for CEC, 2011 competition on testing evolutionary algorithms on real world optimization problems," Tech. Rep., Jadavpur University, Kolkata, India, 2010.
- [23] A. Horner, J. Beauchamp, and L. Haken, "Machine tongues XVI: genetic algorithms and their application to FM matching synthesis," *Computer Music Journal*, vol. 17, no. 4, pp. 17–29, 1993.
- [24] H. Wang, S. Rahnamayan, H. Sun, and M. G. Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.
- [25] S. Ghosh, S. Das, S. Roy, S. K. Minhazul Islam, and P. N. Suganthan, "A Differential Covariance Matrix Adaptation Evolutionary Algorithm for real parameter optimization," *Information Sciences*, vol. 182, no. 1, pp. 199–219, 2012.
- [26] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.