Applied Soft Computing 37 (2015) 774–786

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A variable reduction strategy for evolutionary algorithms handling equality constraints

Guohua Wu [a], Witold Pedrycz [b,c], P.N. Suganthan [d], Rammohan Mallipeddi [e,*]

[a] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, Hunan, China
[b] Department of Electrical & Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada
[c] Warsaw School of Information Technology, Newelska 6, Warsaw, Poland
[d] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore, Singapore
[e] School of Electronics Engineering, Kyungpook National University, 1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, South Korea

ARTICLE INFO

Article history:
Received 2 March 2014
Received in revised form 26 May 2015
Accepted 7 September 2015
Available online 15 September 2015

Keywords:
Evolutionary computation
Constrained optimization
Equality constraint reduction
Variable reduction

ABSTRACT

Efficient constraint handling techniques are of great significance when Evolutionary Algorithms (EAs) are applied to constrained optimization problems (COPs). Generally, when use EAs to deal with COPs, equality constraints are much harder to satisfy, compared with inequality constraints. In this study, we propose a strategy named equality constraint and variable reduction strategy (ECVRS) to reduce equality constraints as well as variables of COPs. Since equality constraints are always expressed by equations, ECVRS makes use of the variable relationships implied in such equality constraint equations. The essence of ECVRS is it makes some variables of a COP considered be represented and calculated by some other variables, thereby shrinking the search space and leading to efficiency improvement for EAs. Meanwhile, ECVRS eliminates the involved equality constraints that providing variable relationships, thus improves the feasibility of obtained solutions. ECVRS is tested on many benchmark problems. Computational results and comparative studies verify the effectiveness of the proposed ECVRS.

1. Introduction

Optimization plays an important role in areas of engineering, management, commercial, etc. as many problems in the real world are essentially optimization tasks. Particularly, many of these real life applications like VLSI design, structural optimization, and location problems involve difficult numerical constrained optimization problems (COPs) [1]. Mathematical programming (MP) techniques are generally the first option to solve optimization problems, which can provide, under some specific conditions to be accomplished by the problem, convergence to global optimal solution [2]. However, due to the fact that these problems may have complex constraints and objective functions characterized as multi-modality, non-continuity, non-differentiability and high dimensionality, which lead to complicated search landscapes, traditional MP methods often fail to solve such problems as they may easily get trapped in local optima.

In response to this challenge, many Evolutionary Algorithms (EAs), such as Genetic Algorithms (GAs) [3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5] and Particle Swarm Optimization (PSO) [6,7], have emerged [8]. Because of their competitive performance in solving complex optimization problems, these nature-inspired algorithms attract extensive attention in recent years. In general, EAs like DE and PSO were initially proposed to solve numerical unconstrained optimization problems, thereby they lack explicit constraint handling techniques. When solving COPs, candidate solutions may be infeasible because of the violation of constraints [9]. The search space becomes more complicated (e.g. being separated into multiple discontinuous areas) because of the impact of constraints. It is therefore of great significance to integrate effective constraint handling techniques into EAs when using them to solve COPs.

The main task of a constraint handling technique is to deal with infeasible solutions to drive the search efforts to feasible solution region while maintain the diversity of the overall population. A variety of constraint handling techniques have been proposed [10–12], such as penalty function [13–15], separation of objective function and constraints [16,17], feasibility rules [18–22], stochastic ranking [23–25], ε-constrained method [26–28], multi-objective concepts [29–31] and ensemble of constraint handling techniques [9,32,33].

Constraints of optimization problems can be categorized into inequality constraints and equality constraints. Empirical

* Corresponding author. Tel.: +82 1035751276.
E-mail address: mallipeddi.ram@gmail.com (R. Mallipeddi).

http://dx.doi.org/10.1016/j.asoc.2015.09.007
1568-4946/

evidences show that equality constraints are more difficult to satisfy than inequality constraints [2,34,35]. In addition, it was found that the rate of feasible solutions in the search space of a COP becomes low if there are more equality constraints required to satisfy [36,37]. Previous constraint handling methods mainly focus on the effective and comprehensive evaluation of feasible and infeasible solutions. In major previous studies, an equality constraint is usually transformed into an inequality constraint with the use of a small threshold value. As a result, the real feasibility degree of an obtained solution is highly related to the selected threshold value. In some high-precision real-life optimization domain, the feasibility of the solution can be a concern. To address this issue, we need to reduce the threshold value. Nevertheless, the efficiency of EAs degrades significantly with the decrease of the threshold value. It thus remains a challenge to generate solutions with high quality and feasibility simultaneously for equality COPs.

In this study, we make use of variable relationships implied in equality constraints in order to represent and determine some variables by others. Based on this idea, we propose an equality constraint and variable reduction strategy (ECVRS) to improve the efficiency of EAs in solving COPs. ECVRS can eliminate equality constraints so as to improve the feasibility of obtained solutions. Meanwhile, it is able to reduce variables, thus shrinks the solution space of COPs. As the proposed ECVRS only need to modify the original COPs, theoretically it can be integrated with any EA.

A related concept in conventional optimization is called "variable elimination" [38]. Variable elimination is usually applied to constraint satisfaction problems (CSPs) to support the search and pruning process [39]. In reality, since variable elimination is time and space exponential in a graph called induced width, it is not suitable to many class of problems [40,41]. In addition, variable elimination theory is specifically oriented to discrete optimization area [42,43]. ECVRS is different from variable elimination in following aspects. First, ECVRS is aimed to find variable relationships in COPs before EAs are used to solving the COPs, thus it is not dynamically used in the solution search process as variable elimination. Consequently, implementations of variable reduction and variable elimination are different. Second, it is possible to reduce variables involved in nonlinear equality constraints. As shown in experiments, we have successfully applied ECVRS to many benchmark COPs with nonlinear equality constraints. Third, ECVRS presented here is specifically oriented to continuous COPs rather than discrete ones.

The rest of this paper is structured as follows: Section 2 briefly introduces the formation of COPs and reviews related works. Section 3 introduces the equality constraint and variable reduction strategy (ECVRS). Section 4 reports the experimental results and comparative studies. Section 5 concludes this paper identifying future research directions.

2. Related works

2.1. Constrained optimization problem

Without loss of generality, a COP can be defined as:

$$\text{Minimize}: \quad f(X) \tag{1}$$

$$\text{Subject to}: \quad g_i(X) \le 0, \quad i = 1, \ldots, p \tag{2}$$

$$h_j(X) = 0, \quad j = 1, \ldots, m \tag{3}$$

$$l_k \le x_k \le u_k, \quad k = 1, \ldots, n \tag{4}$$

where $X=(x_1, x_2, \ldots, x_n)$ is the vector of solutions with n variables; p is the number of inequality constraint; m is the number of equality constraint; The values of l_k and u_k are the lower and upper bounds of x_k, respectively. When solving COPs, equality constraints are usually transformed into inequality constraint as [10,23]: $|h_j(X)| - \varepsilon \le 0$, where ε is the tolerance allowed (very small value).

There are many complex real world COPs, such as optimal reactive dispatch problem [32], economic load dispatch [44] and reactive power and voltage control [44]. In academic community, a special session and a competition on real-parameter optimization were held in IEEE Congress of Evolutionary Computation 2006 (CEC 2006) and 2010 respectively [36,37]. In addition, some real world COPs are defined and summarized in the competition on testing EAs on real world optimization problems in CEC 2011 [45].

2.2. Constraint handling techniques incorporated into EAs

EAs were generally developed for unconstrained optimization problems (with unconstraint search space), without explicit mechanisms to deal with constraints [46]. Constraints in a COP may cause the search to keep away the focus on optimization to just seeking a feasible (i.e. valid) solution [10]. Therefore, the integration of effective constraint handling techniques into EAs is necessary when using them to solve COPs. Comprehensive survey of constraint handling techniques are given in [10–12].

The straightforward and earliest idea is to translate a COP into an unconstrained optimization problem with the aid of penalty functions [9]. This method punishes the solution violating constraints by adding a penalty value to its fitness value, to direct infeasible individuals to move to feasible area. It can be described in the following form [10]:

$$\phi(X) = f(X) + p(X) \tag{5}$$

where $\phi(X)$ is the new optimized objective function expanded with penalty function, and $p(X)$ denotes the penalty function that can be formulated as follows:

$$p(X) = \sum_{i=1}^{p} r_i \cdot \max(0, g_i(X))^a + \sum_{j=1}^{m} c_j \cdot |h_j(X)|^b \tag{6}$$

where r_i, c_j, a, b are all parameters used to control the scale and weight of each constraint. Since equality constraints are often transformed into inequality constraints with the aid of a threshold value ε, (6) can also be represented as below.

$$p(X) = \sum_{i=1}^{p} r_i \cdot \max(0, g_i(X))^a + \sum_{j=1}^{m} c_j \cdot \max(0, (h_j(X) - \varepsilon))^b \tag{7}$$

The static penalty function is popular at the beginning due to its simplicity, viz., the parameters in (7) are fixed during the evolutionary process of EAs [47]. However, the amount of penalties added to the objective function is required to properly controlled, as for different problems or different evolutionary stages of one problem, the penalty amount should be different to make EAs more efficient. To overcome this shortcoming, adaptive penalty function (APF) was proposed [13–15].

Another popular constraint handling technique is called feasibility rules (FR) [48]. The advantage of FR is its simplicity to be incorporated into different EAs, meanwhile, unlike APF, FR does not introduce any new parameter. FR can be descried as [49]: (1) When comparing two feasible solutions, the one with the best objective function is chosen. (2) When comparing a feasible and an infeasible solution, the feasible one is chosen. (3) When comparing two infeasible solutions, the one with the lowest sum of constraint violation is chosen.

Stochastic ranking (SR) is also an effective constraint handling technique which was initially developed by Runarsson and Yao [23,24]. SR aims to overcome the inherent shortcomings of PF because of the improper use of penalty factors. SR introduces a probability factor p_f to determine whether the objective function value or the constraint violation value will be used to rank individuals. If a random value (uniformly distributed from 0 to 1) is smaller than p_f, SR will rank solutions only in terms of their objective values; otherwise, SR will determine the ranks of individuals based on constraint violation values [23].

ε-Constrained method is a constraint handling technique proposed by Takahama and Sakai [50] recently. This technique works according to following rules. (1) If the constraint violation values of two solutions are equal, they will be ranked in terms of their objective values. (2) If both constraint violation values of two solutions are smaller than the value of ε, they will be ranked by their objective values. (3) Otherwise, they will be ranked in accordance with their constraint violation values. ε-Constrained method was further improved in conjunction with a gradient-based mutation strategy [51].

According to the no free lunch theorem [52], it is impossible for a single constraint handling technique to outperform all other techniques on every problem. Therefore, Mallipeddi and Suganthan [9] proposed an ensemble of four constraint handling techniques (ECHT) mentioned above. In ECHT, each single constraint handling technique has its own population. In addition, different sub-populations can communicate with and learn from each other by sharing all of their offspring. ECHT has been adopted in evolutionary programming (EP) [53] and Differential Evolution (DE) [9,33].

2.3. Problem domain knowledge based strategies adopted in EAs

Currently, in addition to the design of more efficient generic EAs, the adoption of problem domain knowledge is also very common when EAs are applied to concrete optimization problems. As for discrete optimization, take ant colony optimization (ACO) as an example, the domain knowledge is used for solution representation, neighborhood construction and search strategy design [54,55]. With regard to continuous optimization, gradient knowledge of optimization problems can be used to identify more effective search directions, which can enhance the local search capability of EAs [56] or push infeasible solutions to feasible area [28]. Particularly, in [8], the authors employed derivative knowledge of optimization functions to obtain quantitative relationships of variables, such that a variable reduction strategy (VRS) was proposed and integrated into PSO variants. Experimental tests shown that VRS can simplify the original optimization problems and improve the efficiency of PSO variants.

In this study, we exploit the quantitative variable relationships implied in equality constraints of COPs with an attempt to reduce equality constraints as well as variables.

3. Equality constraint and variable reduction strategy

3.1. Impacts of equality constraints

Equality constraints are harder to satisfy because they make the feasible solution space become very small. Statistical information about the 24 benchmark COPs used in CEC 2006 [36] is given in Table 1, from which we can observe that the estimated feasible region ratio associated with each COP having equality constraints is pretty low (almost close to zero often). Intuitively, this reveals that equality constraints will make it harder for EAs to find feasible solutions. Therefore, just as mentioned before, when solving COPs, equality constraints are usually transformed into inequality constraint as $|h_j(X)| - \varepsilon \le 0$, where ε is a threshold value.

The threshold value ε should be small. For example, in CEC 2006 and CEC 2010, the value of ε was set to $1e-4$ [36,37]. Especially in some engineering areas, the feasibility degree of the final solution is crucial. On these occasions, the value of ε should be reduced to improve the precision of equality constraint satisfaction. However, it is shown that the threshold value $\varepsilon \le 1e-7$ already presents difficulties, and that for threshold value $\varepsilon = 1e-15$, the optimum is hardly found at all for COPs [14]. Therefore, new effective equality constraint handling techniques are of great significance.

Table 1
Statistic details of the 24 CEC 2006 benchmark problems. *n* is the number of decision variables, ρ is the estimated ratio between the feasible region and the search space, *LI* is the number of linear inequality constraints, *NI* the number of nonlinear inequality constraints, *LE* is the number of linear equality constraints and *NE* is the number of nonlinear equality constraints. *a* is the number of active constraints on the optima.

Prob.	*n*	Type of function	ρ	*LI*	*NI*	*LE*	*NE*	*a*
g01	13	Quadratic	0.0111%	9	0	0	0	6
g02	20	Nonlinear	99.9971%	0	2	0	0	1
g03	**10**	**Polynomial**	**0.0000%**	**0**	**0**	**0**	**1**	**1**
g04	5	Quadratic	52.1230%	0	6	0	0	2
g05	**4**	**Cubic**	**0.0000%**	**2**	**0**	**0**	**3**	**3**
g06	2	Cubic	0.0066%	0	2	0	0	2
g07	10	Quadratic	0.0003%	3	5	0	0	6
g08	2	Nonlinear	0.8560%	0	2	0	0	0
g09	7	Polynomial	0.5121%	0	4	0	0	2
g10	8	Linear	0.0010%	3	3	0	0	6
g11	**2**	**Quadratic**	**0.0000%**	**0**	**0**	**0**	**1**	**1**
g12	3	Quadratic	4.7713%	0	1	0	0	0
g13	**5**	**Nonlinear**	**0.0000%**	**0**	**0**	**0**	**3**	**3**
g14	**10**	**Nonlinear**	**0.0000%**	**0**	**0**	**3**	**0**	**3**
g15	**3**	**Quadratic**	**0.0000%**	**0**	**0**	**1**	**1**	**2**
g16	5	Nonlinear	0.0204%	4	34	0	0	4
g17	**6**	**Nonlinear**	**0.0000%**	**0**	**0**	**0**	**4**	**4**
g18	9	Quadratic	0.0000%	0	13	0	0	6
g19	15	Nonlinear	33.4761%	0	5	0	0	0
g20	**24**	**Linear**	**0.0000%**	**0**	**6**	**2**	**12**	**16**
g21	**7**	**Linear**	**0.0000%**	**0**	**1**	**0**	**5**	**6**
g22	**22**	**Linear**	**0.0000%**	**0**	**1**	**8**	**11**	**19**
g23	**9**	**Linear**	**0.0000%**	**0**	**2**	**3**	**1**	**6**
g24	2	Linear	79.6556%	0	2	0	0	2

The items indicated in BOLD are the problems with equality constraints.

3.2. Equality constraint and variable reduction strategy

The hardest thing for EAs solving COPs is how to handle equality constraints efficiently [57]. In [58], the authors proposed a local search technique with special emphasis on equality constraints. This technique is to reach a point on the equality constraint from the current position of an individual solution, and then explore on the constraint landscape. This approach may face difficulties when there are many variables and especially many of them are nonlinear, as when it explores the constraint landscape it needs to use numerical search methods to calculate the values of nonlinear variables. Another approach called geometrical crossover for exploring the edge of feasibility area was presented in [59]. Actually, geometrical crossover also makes use of the variable relationships implied in constraint equations and then design sophisticated cross mechanisms to transform one feasible solution to another feasible solution. This method is suitable to linear equality constraints and may fail in nonlinear constraints. To design an effective geometrical crossover mechanism is problem dependent and need special skills.

As shown in (3), an equality constraint is presented in the form of an equation, which actually provides a relationship among some variables in the COP. Through such relationship, if one variable can be represented by some other variables, then this variable can be reduced. This is because in the solution search process, its value can be calculated by the relationship between it and other variables and the values of other variables. As a result, the related equality constraint can always be satisfied by all solutions. On this occasion, equality constraints are no longer barriers to EAs in solving COPs.

On the contrary, they may provide problem domain knowledge that is beneficial to the complexity reduction of COPs. Let us consider a COP described in (1)–(4).

Assume that Ω denotes the set of variables included in COP, $\Omega = \{x_k | k = 1, 2, \ldots, n\}$; Ω_j denotes the collection of variables involved in equality constraint $h_j(X) = 0$ $(1 \le j \le m)$. Obviously, we have $\Omega_j \subset \Omega$. From equality constraint $h_j(X) = 0$ $(1 \le j \le m)$, if we can obtain a relationship as

$$x_k = R_{k,j}(\{x_l | l \in \Omega_j, l \neq k\}) \tag{8}$$

then during the solution search process of EAs, x_k can be actually calculated by relationship $R_{k,j}$ and the values of variables in $\{x_l | l \in \Omega_j, l \neq k\}$. As a result, variable x_k can be reduced. Moreover, equality constraint $h_j(X)$ can be reduced as well, since the information about the constraint is completely captured in (8) to calculate the value of x_k. In addition, according to (8), the bound constraint associated with reduced variable x_k is translated into

$$l_k \le R_{k,j}(\{x_l | l \in \Omega_j, l \neq k\}) \le u_k \tag{9}$$

Let us introduce some essential concepts.

Core variable(s): The variable(s) used to represent other variables in terms of the variable relationships in equality constraints.

Reduced variable(s): The variable(s) expressed and calculated by core variables.

Eliminated equality constraint(s): The equality constraint(s) eliminated along with the reduction of variables due to full satisfaction by all solutions.

Therefore, with the aid of these concepts, we can define the equality constraint and variable reduction strategy (ECVRS) as: making advantage of the equality constraint equations to find a set of core variables with minimum cardinality, such that maximum number of equality constraints and variables are reduced.

Assume that $N = \{1, 2, \ldots, n\}$ denotes the set of indices of overall variables in the original COP, C_1 denotes the set of indices of reduced variables and C_2 is the collection of indices of core variables, hence we have $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$. In addition, let $M = \{1, 2, \ldots, m\}$ denote the collection of indices of equality constraints, M_1 is the collection of indices of reduced equality constraints, and M_2 is the collection of indices of remaining equality constraints. Similarly, we have $M_1 \cup M_2 = M$ and $M_1 \cap M_2 = \emptyset$. After the processing of ECVRS, the original COP given in (1)–(4) can be modified as below.

$$\text{Minimize}: \quad f(X) \tag{10}$$

$$\text{Subject to}: \quad g_i(X) \le 0, \quad i = 1, \ldots, p \tag{11}$$

$$h_j(X_k | k \in C) = 0, \quad j \in M_2 \tag{12}$$

$$l_k \le R_{k,j}(\{x_l | l \in \Omega_j, l \neq k\}) \le u_k, \quad k \in C_1, \quad j \in M_1 \tag{13}$$

$$l_k \le x_k \le u_k, \quad k \in C_2 \tag{14}$$

Noticeably, the task of ECVRS is to maximize the cardinality of C_1 and M_1. If the M_2 is $\emptyset$, then the equality constraints are completely eliminated.

Compared to the original one, the modified COP in (10)–(14) exhibits some unique characteristics. First, some equality constraints can be reduced thereby reducing the difficulty brought by equality constraints. Second, some variables could be reduced thereby shrinking the search space and possibly reducing the number of function calls required by EAs. Third, some bound constraints related to reduced variables will be changed into new inequality constraints (as in (13)), which are easier to cope with than equality ones.

To show how ECVRS works, we provide an illustrative example. Let us consider the following COP:

$$\begin{aligned} &\min x_1^2 + x_2^2 \\ &x_1 + x_2 = 2 \\ &0 \le x_1 \le 5, \quad 0 \le x_2 \le 5 \end{aligned} \tag{15}$$

According to ECVRS, we can obtain a relationship $x_2 = 2 - x_1$ from the equality constraint. The COP presented in (15) can be transformed into

$$\begin{aligned} &\min 2x_1^2 - 4x_1 + 4 \\ &0 \le x_1 \le 2 \end{aligned} \tag{16}$$

As shown in Fig. 1a, the solution space of COP (15) is two-dimensional, and an equality constraint has to be considered during the solution search process. In comparison, as shown in Fig. 1b, the solution space of COP (16) processed by ECVRS becomes one-dimensional with only one bound inequality constraint. Noticeably, the complexity of the original COP is reduced.

3.3. Some empirical guidance for applying ECVRS

Since equality constraints can be nonlinear and very complex, it is a big challenge to design a unified method to reduce all types of equality constraints and variables. More solid theory is still in need to efficiently deal with general nonlinear equality constraints.

Here, let us first safely give some application conclusions. (1) If an equality constraint of a COP includes a variable less than or equal to second-order, this variable and the corresponding equality constraint can be reduced. This is because through the equality constraint equation, we can use other variables to represent that variable less than or equal to second-order. (2) If a certain number of equality constraints of a COP are linear, all these linear equality constraints and the same number of variables can be reduced. (3) In the case of nonlinear equality constraints, whether we can obtain explicit variable relationships and then reduce equality constraints and variables is problem dependent. Nonlinear operators like $\sin x$, $\cos x$, $\ln x$ and e^x could increase the complexity of equality constraints, especially when they co-exist. However, if a variable is operated only by such one operator and separately included in the considered COP, this variable can be reduced.

Practicing engineers usually solve one class of problems with minor variations. Hence, even deriving constraint reduction strategies manually can be beneficial during the repeated runs. From the perspective of real world application, before we use EAs to solve a COP, it would be valuable to check whether ECVRS is applicable. In fact, we have found that ECVRS can be applied to many optimization problems in power systems, such as optimal reactive dispatch problem [32], economic load dispatch [44] and optimal power flow problem [60].

3.4. A formal and automatic approach to deal with linear equality constraints

It is meaningful to develop a practical procedure that can be programmed in a computer to reduce equality constraints and variables automatically. However, as mentioned before, nonlinear and complex equality constraints are difficult to deal with. To provide theoretical and unified approaches for addressing general equality constraints is still an open problem and deserves further investigations. Here we consider a special kind of COP with linear equality constraints and give a formal method to realize the automatic equality constraint and variable reduction.

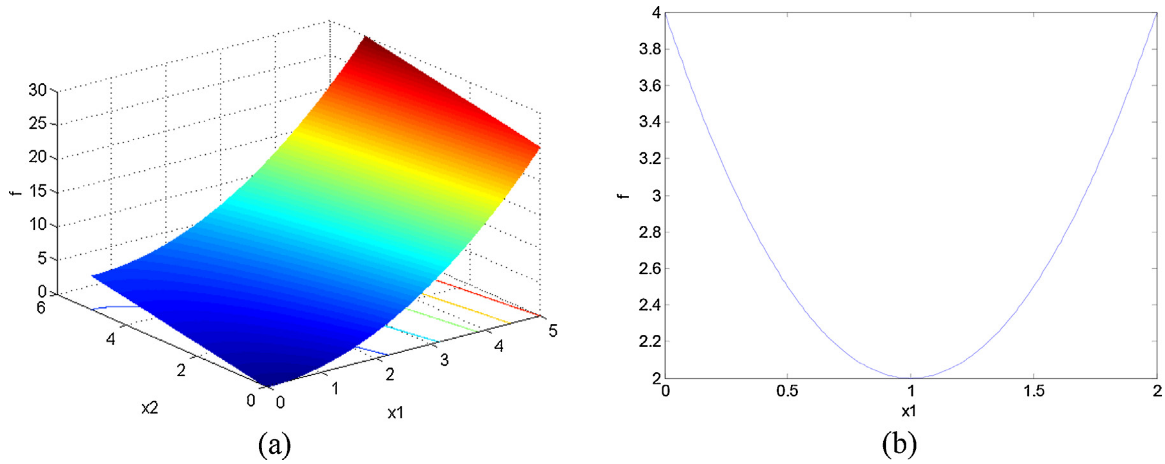

Fig. 1. Illustration of the solution space of the example COP with or without ECVRS: (a) solution space of the original COP and (b) solution space of the COP after ECVRS.

The matrix form of the system of linear equations describing linear equality constraints of a COP is presented as below.

$$AX = b \tag{17}$$

Suppose that the COP has m linear equality constraints and n variables, then A is an $m \times n$ matrix. If the linear system (17) is determined or overdetermined, there is either no solution or a single solution for the system, respectively. These two types of situations are not practical and general in COPs. As a result, we consider the occasion that linear system (17) is underdetermined, namely $m < n$.

Let us expand formula (17) and obtain

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ &\vdots \qquad\quad \vdots \qquad\quad \vdots \\ &a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{aligned} \tag{18}$$

As there are m linear equality constraints, m variables are possibly reduced. Without loss of generality, the reduced variables are x_1, $x_2, \ldots, x_m$ and the core variables are $x_{m+1}, x_{m+2}, \ldots, x_n$. To make the reduced variables be expressed by the core variables, we transform formula (18) into

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + a_{1m}x_m = b_1 - (a_{1,m+1}x_{m+1} + a_{1,m+2}x_{m+2} + \cdots + a_{1,n}x_n) \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m = b_2 - (a_{2,m+1}x_{m+1} + a_{2,m+2}x_{m+2} + \cdots + a_{2,n}x_n) \\ &\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ &a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m = b_m - (a_{m,m+1}x_{m+1} + a_{m,m+2}x_{m+2} + \cdots + a_{m,n}x_n) \end{aligned} \tag{19}$$

Let

$$A' = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{pmatrix}, \quad X' = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \quad \text{and} \quad b' = \begin{pmatrix} b_1 - (a_{1,m+1}x_{m+1} + a_{1,m+2}x_{m+2} + \cdots + a_{1,n}x_n) \\ b_2 - (a_{2,m+1}x_{m+1} + a_{2,m+2}x_{m+2} + \cdots + a_{2,n}x_n) \\ \vdots \\ b_m - (a_{m,m+1}x_{m+1} + a_{m,m+2}x_{m+2} + \cdots + a_{m,n}x_n) \end{pmatrix}$$

Then we have

$$A'X' = b' \tag{20}$$

The linear system (19) is supposed to be determined. As if it is underdetermined, there are one or more equations are redundant because they are the combinations of others. The redundant equations are invalid and can be removed. As a result, the rank of A' is m and A' is invertible. Hence, the linear system (20) can be solved through the following form:

$$X' = (A')^{-1}b' \tag{21}$$

Noticeably, values of reduced variables $x_1, x_2, \ldots, x_m$ can be computed by variable relationships given in (19) and values of core variables $x_{m+1}, x_{m+2}, \ldots, x_n$. As a result, when we employ an EA to solve the COP, the EA only needs to search the values of variables $x_{m+1}, x_{m+2}, \ldots, x_n$. The linear equality constraint elimination and variable reduction process is equivalent to the solution of a determined linear system (20).

The time complexity for computing $(A')^{-1}$, b' and $(A')^{-1}b'$ are $o(m^3)$, $(n-m)m$ and $2m^2$, respectively. Since $(A')^{-1}$ is computed only once during the entire problem solving process, its time complexity can be neglected compared to the overall time required for solving the problem. At each iteration of solution search process, the time complexity of ECVRS is $(n-m)m + 2m^2 = nm + m^2$. Traditionally, (i.e. without using ECVRS,) at each iteration, to compute the equality constraint violation values according to (18), the time complexity is $2nm$. Because $m < n$, there exists

$$nm + m^2 < 2nm \tag{22}$$

As a result, ECVRS is able to reduce the computation time required at every search iteration. In fact, as ECVRS can reduce variables, eliminate equality constraints and shrink solution space, it may also greatly reduce the search iterations, thereby accelerating the problem solving process significantly.

Table 2
The impact of ECVRS on the number of variables and equality constraints of the benchmark COPs.

Problem		Original COP	COP after ECVRS
g03	Variables	10	**9**
	Equality const.	1	**0**
g05	Variables	4	**1**
	Equality const.	3	**0**
g11	Variables	2	**1**
	Equality const.	1	**0**
g13	Variables	5	**2**
	Equality const.	3	**0**
g14	Variables	10	**7**
	Equality const.	3	**0**
g15	Variables	3	**2**
	Equality const.	1	**0**
g17	Variables	6	**2**
	Equality const.	4	**0**
g21	Variables	7	**2**
	Equality const.	5	**0**
g22	Variables	22	**3**
	Equality const.	19	**0**
g23	Variables	9	**5**
	Equality const.	4	**0**

The items indicated in BOLD are the problems with equality constraints.

4. Experimental tests

4.1. Experiment setting

The benchmark COPs g3, g5, g11, g13, g14, g15, g17, g21, g22 and g23 presented in CEC 2006 [36] were selected to test the effectiveness of ECVRS, as they contain equality constraints. An ensemble of constraint handling techniques (ECHT) was proposed to solve COPs to overcome the shortcoming that a single constraint handling technique may be suited to some COPs while fails in others [9]. ECHT was combined with differential evolution (ECHT-DE) and evolutionary programming (ECHT-EP) respectively, both of which show state-of-the-art performance. In our experiments, ECHT-DE and ECHT-EP are taken as solvers (search engine) for the benchmark COPs. In addition, superiority of feasible solutions (SF) constraint handling technique integrated with DE (SF-DE) and ε-constraint (EC) handling technique integrated with EP (EC-EP) are also selected as comparative algorithms. ECVRS is either incorporated into each comparative EA (i.e. ECHT-DE, SF-DE, ECHT-EP and EC-EP) or not so as to test whether it improves the efficiency of EAs.

4.2. Experimental results

The detailed experimental results of all benchmark COPs are listed in Table 3. The execution process of ECVRS on each benchmark COP is given in the Appendix of this paper. Table 2 displays the impacts of ECVRS on the variables and equality constraints of benchmark COPs. It can be observed that all equality constraints and the corresponding number of variables of each benchmark COP are reduced. For instance, for COP g22, there are 19 equality constraints and 19 variables being reduced. It should be noted that, except COPs g11 and g14, all other benchmark COPs have nonlinear equality constraints, which indicates that ECVRS can deal with nonlinear equality constraints under some conditions. As a result, ECVRS may have great potential in solving the real world COPs.

According to the results given in Table 3, we can make some observations. First, for problem g22, we have obtained the newly best result ever known. According to [36], the previously best-known result of problem g22 was 236.4310. In contrast, with ECVRS, all EAs can obtain a solution with the objective function value being equal to 236.3703 and the constraint violation value being equal to zero. It should be noted that it seems that EAs without ECVRS can produce solutions with smaller objective function values for COPs g3, g13 and g14. For example, the result of COP g13 obtained by ECHT-DE without ECVRS is 5.3942e−02, while that obtained by ECHT-DE with ECVRS is 5.3949e−02. The fact is that ECHT-DE without ECVRS obtains better objective function values at the cost of bigger constraint violations. For instance, ECHT-DE without ECVRS obtains the solution for COP g13 with constraint violation value being 3.9935e−04, while the violation value of the solution generated by ECHT-DE with ECVRS is equal to zero. Strictly speaking, these solutions obtained by EAs without ECVRS are infeasible.

Second, for COPs g13, g14, g21, g22 and g23, EAs with ECVRS obtained significantly better results than those without ECVRS. Especially, for COPs g22 and g23, it is difficult for EAs without ECVRS to find satisfactory results. This is because there are many equality constraints in these two COPs, thereby making it hard for EAs to cope with them. However, with the aid of ECVRS, equality constrains of the COPs are all eliminated, thus the performance of EAs is enhanced considerably. It is worth noting that in some COPs (i.e. g5, g17, g21, g23), there are more variables in constraints than in the corresponding objective functions. Although modifications conducted by ECVRS on constraints of such COPs may have no direct impact on the objective function values, ECVRS improves the performance of EAs and speeds up the solution search process in solving these problems. The reason is that in constrained optimization solutions violating constraints are discarded irrespective of their fitness values. This is always the case irrespective of which variables are in the objective or constraint equations. In other words, high-quality solutions with respect to objective function values first must be feasible solutions, whereas equality constraints hamper EAs in finding feasible solutions. ECVRS makes the constraints be satisfied more easily, thereby enabling EAs to expend less computational resources for finding solutions with superior objective function values.

Third, in conjunction with ECVRS, EAs can always find completely feasible solutions. That is to say, ECVRS enables EAs to obtain high-quality and high-feasibility solutions for COPs. Moreover, it can be found that the performance improvement extent brought by ECVRS depends on the amount of reduced equality constraints and variables. For example, the improvements for COPs g13, g14, and g15 are relatively small, as relatively less variables and equality constraints are reduced. In contrast, the improvements for COPs g21, g22 and g23 are significant, because more variables and equality constraints are reduced. Especially, an EA cannot find satisfactory solutions for COP g22 unless it is combined with ECVRS.

4.3. Convergence analysis

To test the impact of ECVRS on the convergence performance, we first define that a satisfactory solution is a feasible solution (with regard to EAs without ECVRS, we allow a 1e−4 relaxation for equality constraints) and 99% close to the best known. If an EA finds a satisfactory solution in a run, this run is viewed as successful. Table 4 provides the mean function evaluations (FEs) required by each EA to obtain a satisfactory solution, and the number of successful runs (Suc) among 25 runs. From Table 4, we can discover that with ECVRS, all EAs always generate satisfactory solutions for all COPs. In addition, the convergence of all EAs is significantly improved by ECVRS, which enables each EA to find satisfactory solutions for COPs at the cost of much less FEs.

The convergence processes of each EA on all benchmark COPs are displayed in Fig. 2, which shows that compared to EAs without ECVRS, EAs with ECVRS always present much faster convergence for every COP. Particularly, for COP g22, EAs without ECVRS cannot converge to a satisfactory solution effectively. In contrast, EAs with ECVRS can find a high-quality solution very quickly.

Table 3
Computational results of each EA with or without ECVRS.

Problems		ECHT-DE without ECVRS	ECHT-DE with ECVRS	ECHT-EP without ECVRS	ECHT-EP with ECVRS	SF-DE without ECVRS	SF -DE with ECVRS	EC-EP without ECVRS	EC-EP with ECVRS
g03	Best	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**
	Median	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**
	Mean	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**
	Worst	−1.0005	**−1.0000**	−1.0004	**−1.0000**	−1.0005	**−1.0000**	−1.0005	**−1.0000**
	Std	2.3930e−10	**2.1612e−16**	1.6026e−05	**8.7721e−14**	3.3013e−16	**2.8816e−16**	1.4728e−06	**2.8658e−16**
	Violation	3.6892e−04	**0.0**	2.6684e−04	**0.0**	2.6631e−04	**0.0**	1.0000e−04	**0.0**
g05	Best	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**
	Median	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**
	Mean	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**
	Worst	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1265e+03	**5.1265e+03**	5.1266e+03	**5.1265e+03**
	Std	2.0865e−12	**9.33125e−13**	2.0212e−07	**9.3312e−13**	1.9236e−12	**9.3312e−13**	3.2214e−02	**9.3312e−13**
	Violation	3.0543e−04	**0.0**	4.6534e−04	**0.0**	8.0598e−04	**0.0**	6.0558e−04	**0.0**
g11	Best	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**
	Median	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**
	Mean	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**
	Worst	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**	7.4990e−01	**7.5000e−01**
	Std	1.1390e−16	**0.0**	1.1390e−16	**0.0**	1.1390e−16	**0.0**	2.1630e−09	**0.0**
	Violation	1.0989e−04	**0.0**	1.0000e−05	**0.0**	1.0539e−04	**0.0**	9.9999e−05	**0.0**
g13	Best	5.3942e−02	**5.3949e−02**	5.3942e−02	**5.3949e−02**	5.3942e−02	**5.3949e−02**	5.4137e−02	**5.3949e−02**
	Median	5.3942e−02	**5.3949e−02**	5.3942e−02	**5.3949e−02**	4.3883e−01	**5.3949e−02**	5.4285e−02	**5.3949e−02**
	Mean	1.3124e−01	**5.3949e−02**	5.3942e−02	**5.3949e−02**	3.5288e−01	**5.3949e−02**	5.4375e−02	**5.3949e−02**
	Worst	4.4373e−01	**5.3949e−02**	5.3942e−02	**5.3949e−02**	4.6384e−01	**5.3949e−02**	5.6346e−02	**5.3949e−02**
	Std	1.5841e−01	**6.3675e−18**	5.3942e−02	**1.5597e−17**	1.4745e−01	**7.9594e−18**	6.3439e−03	**1.2834e−17**
	Violation	3.9935e−04	**0.0**	5.0444e−09	**0.0**	1.0000e−04	**0.0**	2.7237e−02	**0.0**
g14	Best	−4.7765e+01	**−4.7761e+01**	−4.7761e+01	**−4.7761e+01**	−4.7765e+01	**−4.7761e+01**	−4.7765e+01	**−4.7761e+01**
	Median	−4.7765e+01	**−4.7761e+01**	−4.7715e+01	**−4.7761e+01**	−4.7765e+01	**−4.7761e+01**	−4.5053e+01	**−4.7721e+01**
	Mean	−4.7765e+01	**−4.7761e+01**	−4.7703e+01	**−4.7761e+01**	−4.7765e+01	**−4.7761e+01**	−4.5142e+01	**−4.7706e+01**
	Worst	−4.7765e+01	**−4.7761e+01**	−4.7405e+01	**−4.7761e+01**	−4.7765e+01	**−4.7761e+01**	−4.3762e+01	**−4.7361e+01**
	Std	2.1625e−05	**1.6703e−14**	7.8687e−02	**2.9722e−12**	1.8728e−14	**3.7355e−14**	7.3651e−01	**2.1542e−02**
	Violation	2.9212e−04	**0.0**	2.9999e−04	**0.0**	3.0000e−004	**0.0**	2.9999e−04	**0.0**
g15	Best	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6233e+02	**9.6172e+02**
	Median	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6242e+02	**9.6172e+02**
	Mean	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6248e+02	**9.6172e+02**
	Worst	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6172e+02	**9.6172e+02**	9.6265e+02	**9.6172e+02**
	Std	5.8320e−13	**1.1664e−13**	6.1830e−13	**1.1664e−13**	5.8320e−13	**1.1664e−13**	7.3719e+01	**1.1664e−13**
	Violation	1.9995e−04	**0.0**	1.9999e−04	**0.0**	2.0000e−04	**0.0**	2.1737e−01	**0.0**
g17	Best	8.8535e+03	**8.8535e+03**	8.8535e+03	**8.8535e+03**	8.8535e+03	**8.8535e+03**	9.1573e+03	**8.8535e+03**
	Median	8.8535e+03	**8.8535e+03**	8.8535e+03	**8.8535e+03**	8.8538e+03	**8.8535e+03**	9.1791e+03	**8.8535e+03**
	Mean	8.8535e+03	**8.8535e+03**	8.8535e+03	**8.8535e+03**	8.8757e+03	**8.8535e+03**	9.1791e+03	**8.8535e+03**
	Worst	8.8535e+03	**8.8535e+03**	8.8535e+03	**8.8535e+03**	8.9439e+03	**8.8535e+03**	9.2005e+03	**8.8535e+03**
	Std	3.7324e−12	**1.8662e−12**	2.0301e−08	**1.8662e−12**	3.8524e+01	**1.8662e−12**	1.2346e+01	**1.8662e−12**
	Violation	3.2953e−04	**0.0**	2.6943e−04	**0.0**	1.7744e−04	**0.0**	3.1295e−04	**0.0**
g21	Best	1.9372e+02	**1.9379e+02**	1.9372e+02	**1.9379e+02**	1.9372e+02	**1.9379e+02**	1.9872e+02	**1.9379e+02**
	Median	1.9372e+02	**1.9379e+02**	1.9374e+02	**1.9379e+02**	1.9372e+02	**1.9379e+02**	2.4014e+02	**1.9379e+02**
	Mean	1.9984e+02	**1.9379e+02**	1.9498e+02	**1.9379e+02**	2.0682e+02	**1.9379e+02**	2.3474e+02	**1.9379e+02**
	Worst	3.1604e+02	**1.9379e+02**	2.0661e+02	**1.9379e+02**	3.2470e+02	**1.9379e+02**	2.7589e+02	**1.9379e+02**
	Std	2.7351e+01	**2.8632e−12**	3.8129e+00	**3.8765e−12**	4.0314e+01	**6.8507e−10**	2.6621e+01	**3.8625e−12**
	Violation	4.0195e−04	**0.0**	4.8585e−04	**0.0**	9.9999e−05	**0.0**	3.0432e−03	**0.0**
g22	Best	1.8857e+03	**2.3637e+02**	3.9184e+02	**2.3637e+02**	3.9643e+03	**2.3637e+02**	2.2545e+03	**2.3637e+02**
	Median	9.8406e+03	**2.3637e+02**	6.8360e+02	**2.3637e+02**	1.5369e+04	**2.3637e+02**	8.3654e+03	**2.3637e+02**
	Mean	1.0158e+04	**2.3637e+02**	7.7786e+02	**2.3637e+02**	1.3812e+04	**2.3637e+02**	1.2854e+04	**2.3637e+02**
	Worst	1.7641e+04	**2.3637e+02**	1.4844e+03	**2.3637e+02**	1.9205e+04	**2.3637e+02**	1.6328e+04	**2.3637e+02**
	Std	4.2890e+03	**1.4580e−13**	3.0970e+02	**2.2875e−13**	5.0860e+03	**7.3769e−14**	3.2582e+03	**1.9875e−13**
	Violation	4.1562e+03	**0.0**	2.7186e+03	**0.0**	1.3192e+04	**0.0**	4.156e+03	**0.0**
g23	Best	−3.9072e+02	**−4.0000e+02**	−3.4556e+02	**−4.0000e+02**	−3.9158e+02	**−4.0000e+02**	−3.8625e+02	**−4.0000e+02**
	Median	−3.7416e+02	**−4.0000e+02**	−3.1065e+02	**−4.0000e+02**	−3.8465e+02	**−4.0000e+02**	−3.5865e+02	**−4.0000e+02**
	Mean	−3.6413e+02	**−4.0000e+02**	−3.0952e+02	**−4.0000e+02**	−2.4367e+02	**−4.0000e+02**	−3.4864e+02	**−4.0000e+02**
	Worst	−2.3426e+02	**−4.0000e+02**	−2.5807e+02	**−4.0000e+02**	−1.0004e+02	**−4.0000e+02**	−2.7235e+02	**−4.0000e+02**
	Std	3.4129e+01	**1.1664e−13**	2.5417e+01	**4.8217e−09**	1.9487e+01	**0.0**	2.3654e+01	**1.7496e−13**
	Violation	3.5951e−04	**0.0**	1.7373e−04	**0.0**	2.5635e−02	**0.0**	8.8827e−02	**0.0**

The items indicated in BOLD are the problems with equality constraints.

4.4. Discussion on experimental results

Reasons for why ECVRS takes effect can be explained as below. When solving a COP, an EA actually parallel searches different dimensions of the solution space of the considered COP in a stochastic manner. This **parallel-in-dimension** and **stochastic** search fashion makes it difficult for EAs to locate a special point which exactly meets variable relationships implied in equality constraints. To illustrate this, let us consider a COP with two variables, say x_1 and x_2. Its solution space is a two-dimensional plane. If it has a single inequality constraint $x_1 + x_2 < 1$, its feasible solution region is still a plane. In contrast, if the inequality constraint is replaced by an equality one, say $x_1 + x_2 = 1$, then its feasible solution region is limited to a line. Obviously, it is much harder for conventional EAs to generate offspring strictly lying on a multi-variable line than in an unconstrained search space. In response to this challenge, ECVRS exploits the variable relationships captured in the equations expressing equality constraints and in order to represent and calculate some variables by using other variables. Through this manner, the involved equality constraint providing variable relationships is always satisfied during the solution search process, in other words, it is eliminated. As a result, ECVRS directs the search efforts of EAs to more feasible area, thus improves the efficiency of EAs and the feasibility of obtained solutions. This is clearly revealed by the data in Table 3 that ECVRS helps EAs to obtain significantly better results for COPs especially those with complex equality constraints (e.g. COP g21, g22 and g23), and that the constraint violation values associated with solutions obtained by EAs with ECVRS are much smaller (refer to rows "violation" of each COP).

In addition, some variables can be reduced along with the equality constraint elimination process, as they can be calculated and expressed by other variables and corresponding variable relationships. As a result, the solution space is shrunk, which speeds up the problem solving process. Table 4 and Fig. 2 together show that

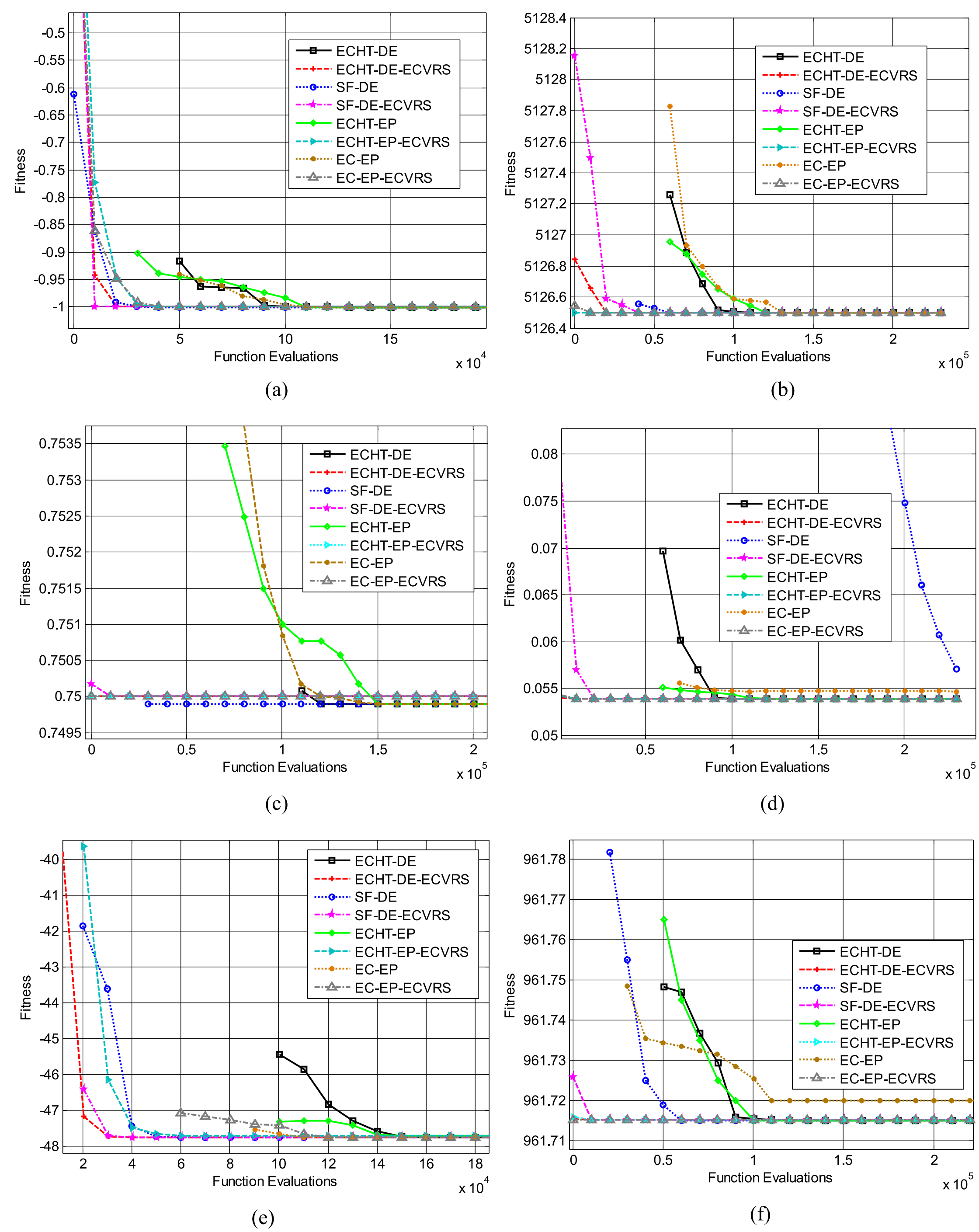

Fig. 2. Illustration of the convergence process of each EA on the benchmark COPs: (a) g3; (b) g5; (c) g11; (d) g13; (e) g14; (f) g15; (h) g17; (i) g21; (j) g22; (k) g23.

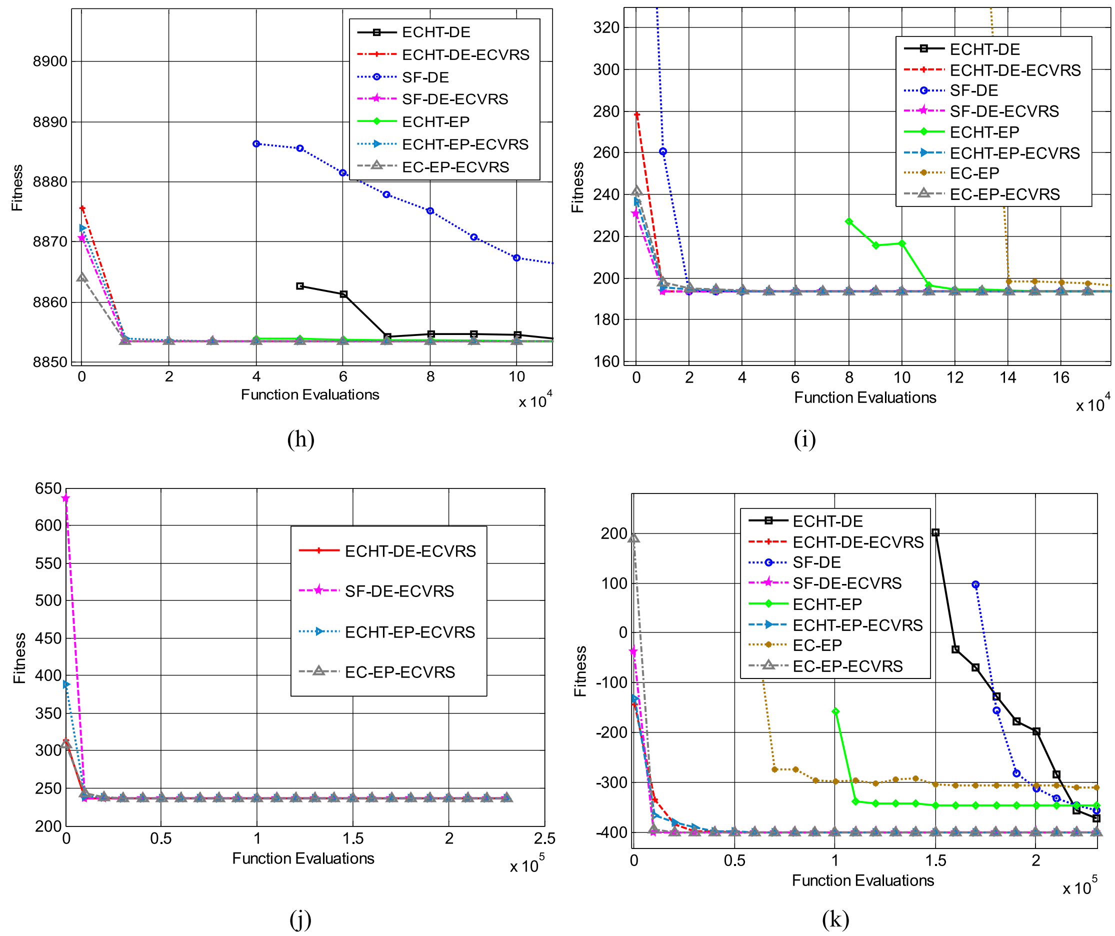

Fig. 2. (*Continued*).

Table 4
Number of function objective evaluation required by each EA with or without ECVRS to reach the near optimal objective function values, FEs is the mean function evaluations required to reach the predefined satisfactory function value, Suc denotes the number of times (out of 25 runs) each EA being able to solve the problem to the required precision.

Problems	ECHT-DE without ECVRS		ECHT-DE with ECVRS		ECHT-EP without ECVRS		ECHT-EP with ECVRS		SF-DE without ECVRS		SF-DE with ECVRS		EC-EP without ECVRS		EC-EP with ECVRS	
	FEs	Suc	FEs	Suc	FEs	Suc	FEs	Suc	FEs	Suc	FEs	Suc	FEs	Suc	FEs	Suc
g03	80,160	25	16,630	25	11,8280	25	18,540	25	22,570	25	6955	25	10,250	25	8555	25
g05	109,760	25	400	25	119,610	25	400	25	27,290	25	200	25	12,290	25	400	25
g11	36,810	25	400	25	120,200	25	400	25	13,090	25	200	25	121,410	25	400	25
g13	89,300	18	840	25	126,600	25	860	25	140,550	5	620	25	56,000	15	1040	25
g14	139,590	25	23,490	25	86,720	25	39,250	25	46,440	25	23,650	25	141,470	25	82,385	25
g15	103,460	25	400	25	120,200	25	400	25	26,750	25	200	25	–	0	400	25
g17	108,340	25	400	25	115,640	25	400	25	40,860	25	200	25	–	0	400	25
g21	107,500	22	6160	25	148,560	22	8280	25	25,870	20	18,800	25	13,556	12	9580	25
g22	–	0	660	25	–	0	1860	25	–	0	455	25	–	0	4526	25
g23	–	0	28,060	25	–	0	37,200	25	–	0	650	25	–	0	24,630	25

ECVRS greatly accelerates EAs with the fact that ECVRS helps EAs to converge to a satisfactory solution of every benchmark COP with much less FEs.

5. Conclusions

Equality constraints are very hard to satisfy when EAs are utilized to solve COPs. In this study, we propose an equality constraint and variable reduction strategy (ECVRS) based on the variable relationships in the equations describing equality constraints. ECVRS can reduce the equality constraints as well as variables of COPs, thereby shrink the solution space, improve the feasibility of obtained solutions and improve the efficiency of used EAs. Experiments on many benchmark COPs illustrate the effectiveness of ECVRS. When applying ECVRS, we only need to modify the considered COP itself. Therefore, it can be integrated with any EA to solve equality constraint COPs. As equality constraints can be nonlinear and complex, how to deal with general equality constraints in line with the variable reduction principle is an open problem. To facilitate to the application of ECVRS, we present some application conclusions. Particularly, we provide a general and formal ECVRS approach to tackle linear constraints.

As shown in Section 4.2, ECVRS is expected to have significant potential in real world COPs. In addition, although only continuous

COPs are discussed in this study, it is easy to apply ECVRS to discrete COPs (e.g. complex scheduling problems and task assignment problems), only if these problems come with equality constraints.

Some future research can be carried out in the following three directions: (1) Investigate general and theoretical approaches to deal with complex and nonlinear equality constraints. (2) Design effective methods to realize ECVRS automatically by computer software. (3) Apply ECVRS to many real world COPs. In fact, we have found that ECVRS can be applied to many optimization problems in power systems, such as optimal reactive dispatch problem [32] and economic load dispatch [44].

Acknowledgments

This work was supported by the Singapore National Research Foundation (NRF) under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, and Cambridge Advanced Research Centre in Energy Efficiency in Singapore (CARES), C4T project.

This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 61563016, 41571397 and 51178193.

Appendix.

Below are the processes of using ECVRS to reduce the complexity of benchmark constrained optimization problems.

g3:

$$\begin{aligned} &\min f(X) = -(\sqrt{n})^n \textstyle\prod_{i=1}^n x_i \\ &\text{s.t.} \quad h_1(X) = \textstyle\sum_{i=1}^n x_i^2 - 1 = 0 \\ &\qquad 0 \le x_i \le 1 \quad i = 1, \ldots, n, \quad n = 10 \end{aligned} \tag{17}$$

According to the equality constraint of the problem given in (17) we can obtain the relation $x_1 = \sqrt{1 - \sum_{i=2}^n x_i^2}$. As a result, variable x_1 and the equality constraint can be reduced. Besides, the ordinary inequality constraint will be added: $0 \le \sqrt{1 - \sum_{i=2}^n x_i^2} \le 1$.

g5:

$$\begin{aligned} &\min f(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + \left(\frac{0.000002}{3}\right) x_2^3 \\ &\text{s.t.} \quad g_1(x) = -x_4 + x_3 - 0.55 \le 0 \\ &\quad g_2(x) = -x_3 + x_4 - 0.55 \le 0 \\ &\quad h_1(x) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ &\quad h_2(x) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ &\quad h_3(x) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0 \end{aligned} \tag{18}$$

where $0 \le x_1,\ x_2 \le 1200,\ -0.55 \le x_3,\ x_4 \le 0.55$. From the equality constraints above, we can obtain the following variable relations.

$$\begin{aligned} &x_3 = x_4 - \arcsin\left(\frac{-1924.8}{1000} - \sin(x_4 - 0.25)\right) - 0.25 \\ &x_1 = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 \\ &x_2 = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 \end{aligned} \tag{19}$$

For problem g5, all equality constraints can be reduced and variable x_3, x_1, x_2 are reduced. Therefore, x_4 is the only core variable. Note that when calculating the objective function after ECVRS, x_3 should be computed prior to the calculation of x_1 and x_2.

g11:

$$\begin{aligned} &\min f(x) = x_1^2 + (x_2 - 1)^2 \\ &\text{s.t.} \quad h(x) = x_2 - x_1^2 = 0 \end{aligned} \tag{20}$$

where $-1 \le x_1, x_2 \le 1$. From the equality constraints above, we can obtain the following variable relations.

$$x_2 = x_1^2 \tag{21}$$

Therefore, for problem g11, the equality constraint and variable x_2 can be eliminated. x_4 is the only core variable.

g13:

$$\begin{aligned} &\min f(x) = e^{x_1x_2x_3x_4x_5} \\ &\text{s.t.} \quad h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ &\quad h_2(x) = x_2x_3 - 5x_4x_5 = 0 \\ &\quad h_3(x) = x_1^3 + x_2^3 + 1 = 0 \end{aligned} \tag{22}$$

where $-2.3 \le x_i \le 2.3$ $(i = 1, 2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3, 4, 5)$. From the equality constrains, we can obtain the following relations.

$$\begin{aligned} &x_1 = (-x_2{}^3 - 1)^{1/3} \\ &x_3 = \frac{5x_4x_5}{x_2} \\ &x_4 = \sqrt{\frac{((-x_2^2 - x_1^2 - x_5^2 + 10)x_2^2)}{(25x_5^2 + x_2^2)}} \end{aligned} \tag{23}$$

As a result, equality constraints in g13 all completely eliminated. In addition, variables x_1, x_3 and x_4 can be reduced. The core variables of this optimization problem are x_2 and x_5.

g14:

$$\begin{aligned} &\min f(X) = \textstyle\sum_{i=1}^{10} x_i \left(c_i + \ln \dfrac{x_i}{\sum_{j=1}^{10} x_j}\right) \\ &\text{s.t.} \quad h_1(X) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0 \\ &\quad h_2(X) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0 \\ &\quad h_3(X) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0 \\ &\quad 0 < x_i \le 10, \quad i = 1, 2, \ldots, 10 \end{aligned} \tag{24}$$

From the related equality constraints in, we can get the variable relation as follows:

$$\begin{aligned} &x_3 = 1 - x_7 - x_8 - 2x_9 - x_{10} \\ &x_4 = 1 - 2x_5 - x_6 - x_7 \\ &x_1 = 2 - 2x_2 - 2x_3 - x_6 - x_{10} \end{aligned} \tag{25}$$

Therefore, with these relations, x_3, x_4 and x_1 can be reduced and the whole three equality constraints can be reduced.

g15:

$$\begin{aligned} &\min f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\ &\text{s.t.} \quad h_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ &\quad h_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0 \end{aligned} \tag{26}$$

where the bounds are $0 \le x_i \le 10$ $(i = 1, 2, 3)$. Also from the equations associated with the equality constraints, we can obtain the following relations.

$$\begin{aligned} &x_2 = 3.2 \pm \sqrt{\frac{5.12x_1}{7} - \frac{12.36x_1^2}{49} + 2.44} + \frac{16x_1}{35} \\ &x_3 = \frac{(56 - 8x_1 - 14x_2)}{7} \end{aligned} \tag{27}$$

The equality constraints of g15 can be eliminated totally accompanied by the reduction of variable x_2 and x_3. Note that in (27), x_2 may have two values at the same time. So when we calculate the objective function, the value producing better function objective will be assigned to x_2.

g17:

$$\min f(x) = f_1(x_1) + f_2(x_2)$$
$$\text{where} \quad f_1(x_1) = \begin{cases} 30x_1 & 0 \le x_1 < 300 \\ 31x_1 & 300 \le x_1 < 400 \end{cases}$$
$$f_2(x_2) = \begin{cases} 28x_2 & 0 \le x_2 < 100 \\ 29x_2 & 100 \le x_2 < 200 \\ 30x_2 & 200 \le x_1 < 1000 \end{cases}$$
$$\begin{aligned} \text{s.t.} \quad & h_1(x) = -x_1 + 300 - \left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 - x_6) \\ & \quad + \left(\frac{0.90789x_3^2}{131.078}\right)\cos(1.47588) \\ & h_2(x) = -x_2 - \left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 + x_6) \\ & \quad + \left(\frac{0.90789x_4^2}{131.078}\right)\cos(1.47588) \\ & h_3(x) = -x_5 - \left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 + x_6) \\ & \quad + \left(\frac{0.90789x_4^2}{131.078}\right) \\ & h_4(x) = 200 - \left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 - x_6) \\ & \quad + \left(\frac{0.90789x_3^2}{131.078}\right)\cos(1.47588) \end{aligned} \tag{28}$$

where the bounds are $0 \le x_1 \le 400$, $0 \le x_2 \le 1000$, $340 \le x_3 \le 420$, $340 \le x_4 \le 420$, $-1000 \le x_5 \le 1000$ and $0 \le x_6 \le 0.5236$. From the equality constraints, we can get the relations as below.

$$\begin{aligned} & x_4 = 131.078\frac{((0.90789x_3^2/131.078)\cos(1.47588) + 200)}{(x_3\cos(1.48477 - x_6))} \\ & x_1 = 300 - \left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 - x_6) + \left(\frac{0.90789x_3^2}{131.078}\right)\cos(1.47588) \\ & x_2 = -\left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 + x_6) + \left(\frac{0.90789x_4^2}{131.078}\right)\cos(1.47588) \\ & x_5 = -\left(\frac{x_3x_4}{131.078}\right)\cos(1.48477 + x_6) + \left(\frac{0.90789x_4^2}{131.078)}\right)\cos(1.47588) \end{aligned} \tag{29}$$

From (29), we can see that equality constraints of problem g17 are all eliminated. The reduced variables are x_1, x_2, x_4 and x_5. The related core variables are x_3 and x_6.

g21:

$$\begin{aligned} & \min f(x) = x_1 \\ & \text{s.t.} \quad g_1(x) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \le 0 \\ & \quad h_1(x) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0 \\ & \quad h_2(x) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0 \\ & \quad h_3(x) = -x_5 + \ln(-x_4 + 900) = 0 \\ & \quad h_4(x) = -x_6 + \ln(x_4 + 300) = 0 \\ & \quad h_5(x) = -x_7 + \ln(-2x_4 + 700) = 0 \end{aligned} \tag{30}$$

where the bounds are $0 \le x_1 \le 1000$, $0 \le x_2$, $x_3 \le 40$, $100 \le x_4 \le 300$, $6.3 \le x_5 \le 6.7$, $5.9 \le x_5 \le 6.4$ and $4.5 \le x_7 \le 6.25$. The variable relations obtained from the equality constraints are as below.

$$\begin{aligned} & x_5 = \ln(-x_4 + 900) = 0 \\ & x_6 = \ln(x_4 + 300) = 0 \\ & x_7 = \ln(-2x_4 + 700) = 0 \\ & x_3 = \frac{(7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6)}{(300 - x_4)} = 0 \\ & x_2 = \frac{(155.365x_4 + 2500x_7 - 25x_4x_7 - 15536.5)}{(x_4 - 100)} = 0 \end{aligned} \tag{31}$$

Relation shown in (31) indicates that all the equality constraints are eliminated with variables x_2, x_3, x_5, x_6 and x_7 being reduced. The core variables are x_1 and x_4 accordingly.

g22:

$$\begin{aligned} & \min f(x) = x_1 \\ & \text{s.t.} \quad g_1(x) = -x_1 + x_2^{0.6} + x_3^{0.6} + x_4^{0.6} \le 0 \\ & \quad h_1(x) = x_5 - 100000x_8 + 1 \times 10^7 = 0 \\ & \quad h_2(x) = x_6 + 100000x_8 - 100000x_9 = 0 \\ & \quad h_3(x) = x_7 + 100000x_9 - 5 \times 10^7 = 0 \\ & \quad h_4(x) = x_5 + 100000x_{10} - 3.3 \times 10^7 = 0 \\ & \quad h_5(x) = x_6 + 100000x_{11} - 4.4 \times 10^7 = 0 \\ & \quad h_6(x) = x_7 + 100000x_{12} - 6.6 \times 10^7 = 0 \\ & \quad h_7(x) = x_5 - 120x_2x_{13} = 0 \\ & \quad h_8(x) = x_6 - 120x_3x_{14} = 0 \\ & \quad h_9(x) = x_7 - 120x_4x_{15} = 0 \\ & \quad h_{10}(x) = x_8 - x_{11} + x_{16} = 0 \\ & \quad h_{11}(x) = x_9 - x_{12} + x_{17} = 0 \\ & \quad h_{12}(x) = -x_{18} + \ln(x_{10} - 100) = 0 \\ & \quad h_{13}(x) = -x_{19} + \ln(-x_8 + 300) = 0 \\ & \quad h_{14}(x) = -x_{20} + \ln(x_{16}) = 0 \\ & \quad h_{15}(x) = -x_{21} + \ln(-x_9 + 400) = 0 \\ & \quad h_{16}(x) = -x_{22} + \ln(x_{17}) = 0 \\ & \quad h_{17}(x) = -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0 \\ & \quad h_{18}(x) = x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0 \\ & \quad h_{19}(x) = x_9 - x_{12} - 4.60517x_{15} + x_{15}x_{22} + 100 = 0 \end{aligned} \tag{32}$$

where $0 \le x_1 \le 20{,}000$, $0 \le x_2, x_3, x_4 \le 1 \times 10^6$, $0 \le x_5, x_6, x_7 \le 4 \times 10^7$, $100 \le x_8 \le 299.99$, $100 \le x_9 \le 399.99$, $100.01 \le x_{10} \le 300$, $100 \le x_{11} \le 400$, $100 \le x_{12} \le 600$, $0 \le x_{13}$, x_{14}, $x_{15} \le 500$, $0.01 \le x_{17} \le 400$ and $-4.7 \le x_{18}$, x_{19}, x_{20}, x_{21}, $x_{22} \le 6.25$. According to the equations involved in the equality constraints of problem g22, we can obtain

$$
\begin{aligned}
&x_5 = 100000x_8 + 1 \times 10^7 \\
&x_6 = -100000x_8 + 100000x_9 \\
&x_7 = -100000x_9 + 5 \times 10^7 \\
&x_{10} = -\frac{(x_5 - 3.3 \times 10^7)}{100000} \\
&x_{11} = -\frac{(x_6 - 4.4 \times 10^7)}{100000} \\
&x_{12} = -\frac{(x_7 - 6.6 \times 10^7)}{100000} \\
&x_{16} = -x_8 + x_{11} \\
&x_{17} = -x_9 + x_{12} \\
&x_{18} = \ln(x_{10} - 100) \\
&x_{19} = \ln(-x_8 + 300) \\
&x_{20} = \ln(x_{16}) \\
&x_{21} = \ln(-x_9 + 400) \\
&x_{22} = \ln(x_{17}) \\
&x_{13} = \frac{(x_8 + x_{10} - 400)}{(x_{18} - x_{19})} \\
&x_{14} = \frac{(x_8 - x_9 - x_{11} + 400)}{(x_{21} - x_{20})} \\
&x_{15} = \frac{(x_9 - x_{12} + 100)}{(4.60517 - x_{22})} \\
&x_2 = \frac{x_5}{120x_{13}} \\
&x_3 = \frac{x_6}{80x_{14}} \\
&x_4 = \frac{x_7}{40x_{15}}
\end{aligned}
\tag{33}
$$

From (33) we can see that although there are 19 equality constraints in problem g22, after using ECVRS, all the equality constraint can be eliminated. And there are only three core variables, i.e. x_1, x_8 and x_9.

g23:

$$
\begin{aligned}
&\min f(x) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7) \\
&\text{s.t.} \quad g_1(x) = x_9x_3 + 0.02x_6 - 0.025x_5 \le 0 \\
&\qquad g_2(x) = x_9x_4 + 0.02x_7 - 0.015x_8 \le 0 \\
&\qquad h_1(x) = x_1 + x_2 - x_3 - x_4 = 0 \\
&\qquad h_2(x) = 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0 \\
&\qquad h_3(x) = x_3 + x_6 - x_5 = 0 \\
&\qquad h_4(x) = x_4 + x_7 - x_8 = 0
\end{aligned}
\tag{34}
$$

where the bounds are $0 \le x_1, x_2, x_6 \le 300$, $0 \le x_3, x_5, x_7 \le 100$, $0 \le x_4, x_8 \le 200$ and $0.01 \le x_9 \le 0.03$. With the equality constraint and variable reduction strategy, the variable relations induced from the equality constraint of problem g23 are given below.

$$
\begin{aligned}
&x_3 = x_5 - x_6 \\
&x_4 = x_8 - x_7 \\
&x_1 = 0.5(100x_9 - 1)(x_3 + x_4) \\
&x_2 = 0.5(3 - 100x_9)(x_3 + x_4)
\end{aligned}
\tag{35}
$$

Then through the relations obtained above, we can eliminate all the equality constraints and variables x_1, x_2, x_3 and x_4. The core variables of g23 are x_5, x_6, x_7, x_8 and x_9.

References

[1] H. Lu, W. Chen, Self-adaptive velocity particle swarm optimization for solving constrained optimization problems, J. Glob. Optim. 41 (2008) 427–445.

[2] E. Mezura-Montes, J.I. Flores-Mendoza, Improved particle swarm optimization in constrained numerical search spaces, in: Nature-Inspired Algorithms For Optimisation, Springer, 2009, pp. 299–332.

[3] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, 1975.

[4] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B: Cybern. 26 (1996) 29–41.

[5] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.

[6] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.

[7] G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, H. Li, Superior solution guided particle swarm optimization combined with local search techniques, Expert Syst. Appl. 41 (2014) 7536–7548.

[8] G. Wu, W. Pedrycz, H. Li, D. Qiu, M. Ma, J. Liu, Complexity reduction in the use of evolutionary algorithms to function optimization: a variable reduction strategy, ScientificWorldJournal 2013 (2013) 1–8.

[9] R. Mallipeddi, P.N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (2010) 561–579.

[10] E. Mezura-Montes, C.A. Coello Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm Evol. Comput. 1 (2011) 173–194.

[11] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (2002) 1245–1287.

[12] E. Mezura-Montes, Constraint-Handling in Evolutionary Optimization, vol. 198, Springer-Verlag, 2009.

[13] R. Datta, K. Deb, An adaptive normalization based constrained handling methodology with hybrid bi-objective and penalty function approach, in: IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.

[14] M.M. Ali, W.X. Zhu, A penalty function-based differential evolution algorithm for constrained global optimization, Comput. Optim. Appl. (2012) 1–33.

[15] R. Farmani, J.A. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Trans. Evol. Comput. 7 (2003) 445–455.

[16] J. Liang, S. Zhigang, L. Zhihui, Coevolutionary comprehensive learning particle swarm optimizer, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.

[17] C.A.C. Coello, Treating constraints as objectives for single-objective evolutionary optimization, Eng. Optim. 32 (2000) 275–308.

[18] S.D. Handoko, C.K. Kwoh, Y.-S. Ong, Feasibility structure modeling: an effective chaperone for constrained memetic algorithms, IEEE Trans. Evol. Comput. 14 (2010) 740–758.

[19] E. Mezura-Montes, J. Velázquez-Reyes, C.A. Coello Coello, Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, 2005, pp. 225–232.

[20] Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Appl. Math. Comput. 186 (2007) 1407–1422.

[21] M.F. Tasgetiren, P. Suganthan, A multi-populated differential evolution algorithm for solving constrained optimization problem, in: IEEE Congress on Evolutionary Computation, 2006, pp. 33–40.

[22] B. Tessema, G.G. Yen, An adaptive penalty formulation for constrained evolutionary optimization, IEEE Trans. Syst. Man Cybern. A: Syst. Hum. 39 (2009) 565–578.

[23] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (2000) 284–294.

[24] T.P. Runarsson, X. Yao, Search biases in constrained evolutionary optimization, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 35 (2005) 233–243.

[25] L. Ali, S.L. Sabat, S.K. Udgata, Particle swarm optimisation with stochastic ranking for constrained numerical and engineering benchmark problems, Int. J. Bio-Inspir. Comput. 4 (2012) 155–166.

[26] T. Takahama, S. Sakai, Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations, IEEE Trans. Evol. Comput. 9 (2005) 437–451.

[27] T. Takahama, S. Sakai, Efficient constrained optimization by the ε constrained rank-based differential evolution, in: IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.

[28] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–9.

[29] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Trans. Evol. Comput. 16 (2012) 117–134.

[30] Y. Wang, Z. Cai, G. Guo, Y. Zhou, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems, IEEE Trans. Syst. Man Cybern. B: Cybern. 37 (2007) 560–575.
[31] Y. Wang, Z. Cai, Y. Zhou, W. Zeng, An adaptive tradeoff model for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 12 (2008) 80–92.
[32] R. Mallipeddi, S. Jeyadevi, P.N. Suganthan, S. Baskar, Efficient constraint handling for optimal reactive power dispatch problems, Swarm Evol. Comput. 5 (2012) 28–36.
[33] R. Mallipeddi, P.N. Suganthan, Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[34] E. Mezura-Montes, C. Coello Coello, Identifying on-line behavior and some sources of difficulty in two competitive approaches for constrained optimization, in: IEEE Congress on Evolutionary Computation, 2005, pp. 1477–1484.
[35] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites, in: IEEE Congress on Evolutionary Computation, 2006, pp. 1–8.
[36] J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C.C. Coello, et al., Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization, J. Appl. Mech. vol. 41 (2006).
[37] R. Mallipeddi, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization, Nanyang Technological University, Singapore, 2010.
[38] V. Weispfenning, Solving constraints by elimination methods, in: Automated Reasoning, Springer, 2004, pp. 336–341.
[39] H. Hong, Heuristic search and pruning in polynomial constraints satisfaction, Ann. Math. Artif. Intell. 19 (1997) 319–334.
[40] J. Larrosa, R. Dechter, Boosting search with variable elimination in constraint optimization and constraint satisfaction problems, Constraints 8 (2003) 303–326.
[41] J. Larrosa, E. Morancho, D. Niso, On the practical use of variable elimination in constraint optimization problems: 'still-life' as a case study, J. Artif. Intell. Res. 23 (2005) 421–440.
[42] V. Chandru, Variable elimination in linear constraints, Comput. J. 36 (1993) 463–472.
[43] R. Loos, V. Weispfenning, Applying linear quantifier elimination, Comput. J. 36 (1993) 450–462.
[44] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2008) 171–195.
[45] S. Das, P. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Jadavpur Univ., Nanyang Technol. Univ., Kolkata, India, 2010.
[46] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, vol. 1, Wiley, Chichester, 2005.
[47] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (1994) 242–253.
[48] A.I. Oyman, K. Deb, H.-G. Beyer, An alternative constraint handling method for evolution strategies, in: Proceedings of Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July, 1999, pp. 612–619.
[49] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2000) 311–338.
[50] T. Takahama, S. Sakai, N. Iwane, Constrained optimization by the ε constrained hybrid algorithm of particle swarm optimization and genetic algorithm, in: AI 2005: Advances in Artificial Intelligence, Springer, 2005, pp. 389–400.
[51] T. Takahama, S. Sakai, Solving difficult constrained optimization problems by the ε constrained differential evolution with gradient-based mutation, in: Constraint-Handling in Evolutionary Optimization, Springer, 2009, pp. 51–72.
[52] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[53] R. Mallipeddi, P. Suganthan, B. Qu, Diversity enhanced adaptive evolutionary programming for solving single objective constrained problems, in: IEEE Congress on Evolutionary Computation, 2009, pp. 2106–2113.
[54] G. Wu, J. Liu, M. Ma, D. Qiu, A two-phase scheduling method with the consideration of task clustering for earth observing satellites, Comput. Oper. Res. 40 (2013) 1884–1894.
[55] G. Wu, M. Ma, J. Zhu, D. Qiu, Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks, J. Syst. Eng. Electron. 23 (2012) 723–733.
[56] M.M. Noel, A new gradient based particle swarm optimization algorithm for accurate computation of global minimum, Appl. Soft Comput. 12 (2012) 353–359.
[57] Z. Michalewicz, C.Z. Janikow, Handling constraints in genetic algorithms, in: ICGA, 1991, pp. 151–157.
[58] A.S. Barkat Ullah, R. Sarker, C. Lokan, Handling equality constraints in evolutionary optimization, Eur. J. Oper. Res. 221 (2012) 480–490.
[59] M. Schoenauer, Z. Michalewicz, Evolutionary computation at the edge of feasibility, in: Parallel Problem Solving from Nature – PPSN IV, Springer, 1996, pp. 245–254.
[60] S. Duman, U. Güvenç, Y. Sönmez, N. Yörükeren, Optimal power flow using gravitational search algorithm, Energy Convers. Manag. 59 (2012) 86–95.