Reactive energy scheduling using bi-objective programming with modified particle swarm optimization

Cheng-Chien Kuo*

Department of Electrical Engineering, Saint John's University, Sec. 4, 499 Tam King Road, Tamsui, Taipei 25135, Taiwan

ABSTRACT

Interactive Bi-objective with Valuable Trade-off programming, together with a modified particle swarm optimization for the daily scheduling of switched capacitors is presented. The two main contradictory concerns of line loss reduction and minimum number of switching operations are considered for realistic request. Both the operating and load constraints for distribution feeders are formulated for practical operation. The proposed approach can provide a set of flexible and valuable trade-off solutions as dictated by decision makers of electric utilities. Quantitative measures can also be provided to aid the decision-making process. To demonstrate the effectiveness and feasibility of the proposed approach, comparative studies were systematically conducted on an actual feeder. The experiment showed encouraging results suggesting that the proposed approach was capable of efficiently determining better quality solutions.

1. Introduction

Capacitors are widely employed in distribution systems for reactive energy compensation to achieve line loss reduction and voltage regulation. The purpose of scheduling for switched capacitors is to determine the daily schedule ('ON' or 'OFF' status) of each switched capacitor, such that the expected constraints and objectives are satisfied. This kind of problem belongs to the class of daily energy operation problems. Owing to the development of distribution automation, capacitors are switched ON and OFF by remote control rather than by manual operation. It would be beneficial for feeder operators to have more flexible capacitor dispatching strategies, so that greater loss reduction and higher power quality could be achieved in daily system operations. Therefore, a fixed-schedule dispatching strategy used nowadays does not allow for this flexibility.

During capacitor switching, both transient over-voltage and high frequency noise are produced and, if large enough, can damage sensitive power electronic devices or even the electrical equipment [1–3]. Although a short transient period could well be tolerated by electronic equipment design, it might cause the activation of protective devices and result in the load being temporarily disconnected from the power circuit. Moreover, over-frequent operation of the switched capacitors decreases the lifetime of the switching equipment. Therefore, the number of switching operations is important to both power utilities and their customers as it represents the power quality both supplied and received. Generally, reactive energy scheduling is formulated as a single objective problem, with daily line loss of feeders employed as the objective function and the number of switching operations treated as the binding constraint. However, since the number of switching operations is important to the operation of the distribution system, it is beneficial to tackle it as an objective function instead of just as a constraint.

The mixed integer programming [4] and dynamic programming (DP) [5–7] were used to be the most popular approach for this kind of combinatorial optimization problems. However, they are strongly restricted by solution space and easily to fall into local minimum under huge feasible combination. In the past decade, evolutionary computational techniques, such as genetic algorithms (GA) [8–10] and simulated annealing (SA) [11–14] have been widely used to solve reactive energy scheduling problems. These algorithms are in the form of probabilistic heuristics with global search properties. Though GA methods have been employed successfully to solve complex optimization problems, recent research has identified deficiencies in GA performance. This degradation in efficiency is apparent in applications with highly epistatic objective functions (i.e., where the parameters being optimized are highly correlated) [the crossover and mutation operations cannot ensure

* Tel.: +886 920881490; fax: +886 228013973.
E-mail address: cckuo@mail.sju.edu.tw
improved fitness of offspring because chromosomes in the population have similar structures and their average fitness is high toward the end of the evolutionary process [15]. Moreover, the premature convergence of GA degrades its performance and reduces its search capability, which leads to a higher probability of obtaining a local optimum [16].

Recently, a global optimization technique called particle swarm optimization (PSO) [17–19] has been used to solve real-world problems and has aroused researchers’ interest due to its flexibility and efficiency. Limitations regarding the form of the fitness function employed and the continuity of variables used for the classical greedy search technique can be completely eliminated. The PSO, first introduced by Kennedy and Eberhart [20], is one of the modern heuristic algorithms. It was developed through the simulation of a simplified social system and has been found to be robust in solving continuous nonlinear optimization problems [21,22]. In general, the PSO method is usually faster than the SA because it has parallel search techniques. However, similar to GA, the main adversity of PSO is premature convergence, which happens when the particle and group best solutions are trapped into local minima during the search process. Localization occurs because particles have the tendency to fly to local, or near local, optima; therefore, particles will concentrate in a small region and the global exploration ability will be weakened. On the other hand, the most significant characteristic of SA is the probabilistic jumping property called the metropolis process, which can be controlled by adjusting the temperature. It has been theoretically proven that the SA technique converges asymptotically to the global optimum solution with probability ONE [23,24].

Therefore, the SA–PSO method, a hybrid of the advantages from both SA and PSO, provides a searching tool for high-quality solutions. Also, the interactive bi-objective with valuable trade-off (IBVT) programming is proposed to help system operators determine the near optimal capacitor dispatch for the next 24 h of daily operation. Two important objectives are considered: one being the daily loss of a feeder that represents the economic condition and the other the number of switching operations that associates with the electric quality condition for a system. These two objectives are somewhat non-commensurable and usually conflict with each other; hence, any improvement of one objective can be reached only by the reduction of the other. The IBVT method is a powerful tool which can provide a valuable trade-off solution for reactive energy scheduling by following the intention of decision makers (DMs). The feasibility of the proposed approach was demonstrated on a real feeder and compared with the PSO and GA methods.

2. Problem formulation

For ease of exposition, consider a distribution feeder with twelve sections, as shown in Fig. 1. The real and reactive loads for each section of the next day are obtained by short term load forecasting. In Fig. 1, there are 18 switchable capacitors $C_i (i = 1, ... , 18)$ with capacities $Q_i$ ($i = 1, ... , 18$) on the feeder. Each capacitor can be independently operated by the system operators or computers at the control centers via communication links. The daily reactive energy scheduling problem of switched capacitors is to determine the ON/OFF schedule of each capacitor within a 24-h period, such that the total loss of the feeder and the number of switching operations can both be minimized under various load and operating constraints.

2.1. Objective functions

The objective functions considered in this study are:

(a) Economic objective function $E_1(S)$:

The economic objective function employed is the daily line loss of the system as stated below:

$$ E_1(S) = P_{\text{loss}}(S) = \sum_{h=1}^{24} \sum_{i=0}^{N_c} \left( P_i^h \right)^2 + \left( Q_i^h \right)^2 $$

Note that capacitors are grouped into banks of standard discrete capacities. Therefore, capacitor sizes are represented as discrete variables to meet the real situation.

(b) Quality objective function $Q_1(S)$:

This objective function is concerned with the number of switching operations for switched capacitors during a day. It represents the quality of the electricity, as mentioned before.
3. The IBVT programming

Complex real-world decision-making problems are multifarious and their multiple objectives are usually non-commensurable and often in conflict. The ultimate goal in multi-objective optimization is to seek the most preferred solution from the set of pareto optimal solutions. Thus, in this paper, our goal was to develop an efficient interactive solution methodology for generating preferred solutions.

Some literature translates the bi-objective characteristic into a single objective optimization model and, generally, in previous research, two methods have been commonly applied. In the “single objective with constraints” method, the most important item is selected as the main objective function, which is expressed by single objective programming. The other optimization objectives are treated with constraints. Although this method reduces solving difficulty, it does not provide complete programming for problems; thus, obtained solutions may not conform to the principle of optimal benefit, especially when objective functions are in conflict with each other. To optimize the main objective function generally leads to the other objective function values being very close to the set constrained value. In other words, obtained solutions can only meet the basic requirement of the objective function that is treated as a constraint. For example, suppose that one optimization problem seeks solution \( x \), which can minimize the value of objective functions \( O_Q(x) \) and \( O_E(x) \) that are conflicting with each other. To apply the single objective with constraints approach, the problem is expressed as follows, where \( O_Q(x) \) is selected as the main objective:

\[
\text{Minimize } O_Q(x) \text{ such that } O_E(x) \leq C
\]

Because \( O_Q(x) \) and \( O_E(x) \) conflict with each other, to ensure the minimal value for objective function \( O_Q(x) \), the obtained solutions may make the value of constraint \( O_E(x) \) very close to constraint value \( C \), in which case the optimization of \( O_Q(x) \) may not process properly. Moreover, it is also very difficult to choose an ideal value for \( C \). Suppose that Fig. 2 shows the functions of \( O_Q(x) \) and \( O_E(x) \) corresponding to \( x \). As shown, constraint value \( C_1 \) is very close to \( C_2 \). The selection of the constraint value of \( O_E(x) \) between \( C_1 \) and \( C_2 \) exerts insignificant influence on \( O_E(x) \), but considerable effect on \( O_Q(x) \); if \( C_1 \) is selected as the constraint value of \( O_E(x) \), \( O_Q(x) \) can only be obtained for \( O_Q(x) \) even in the best situation. On the other hand, if the constraint value of \( O_E(x) \) is \( C_2 \), the optimal value \( O_E(x) \) can be obtained for \( O_Q(x) \). It is obvious that there is large difference between \( O_1 \) and \( O_2 \); thus, constraint value \( C_2 \) should be much better than \( C_1 \). However, for ordinary optimization problems, a curve
similar to Fig. 2 cannot be obtained in advance, thus it is difficult to select an adequate constraint value C.

The other commonly used method for solving bi-objective programming problems is called the "weighting method". It multiplies all objective functions by weighted value and then adds up all the functions, so as to transform bi-objective programming problems into single objective programming problems, as expressed below:

\[ T = W_1 \times O_1(x) + W_2 \times O_2(x) \] (7)

Whether \( O_1(x) \) or \( O_2(x) \) is selected, they are regarded as objective functions and are important to users. However, because the units of objective function \( O_1(x) \) or \( O_2(x) \) may be different, and there is no direct corresponding transformation relationship, weighted value cannot be used merely to represent the importance of different objective functions. Hence, it is difficult to determine the weighted value (\( W_1, W_2 \)) intentionally.

Based on the above discussion, this paper proposes the IBVT approach which uses a simple interactive method to satisfy users' preferences and obtains the most valuable trade-off solution for the bi-objective function. In the solution process, users do not need to input a weighted value, but only need to make a choice of favorite objective function and the bi-objective problem can be solved smoothly. The main advantage of this method is in providing a larger programming space in the modeling process which is not limited to a single objective programming model. The problems of bi-objective programming can be easily solved so as to provide users with more favorable solutions and a more convenient operating environment.

In this paper, mathematical deduction is applied to explain the theoretical meaning of the most valuable trade-off solution in bi-objective optimization and to construct a complete solution process for the suggested IBVT method.

3.1. Mathematical deduction and theoretical explanation

Assume the bi-objective function to be solved is shown as (6). The pareto optimal front of this bi-objective function (6) is shown as the thick solid line in Fig. 3. \( S_p \) is the single objective optimal solution when considering only objective function \( E(S) \); thus, the corresponding \( E_{ideal} \) should be the optimal value of \( E(S) \). Because the effect of \( Q(S) \) is not considered, the corresponding \( Q_{nonideal} \) should be the nonideal value of \( Q(S) \). On the other hand, \( S_q \) is the single objective optimal solution when considering only the \( Q(S) \) objective function; thus, the corresponding \( Q_{ideal} \) is the optimal value of \( Q(S) \) and the corresponding \( E_{nonideal} \) should be the nonideal value of \( E(S) \). The solution that makes \( (E(S), Q(S)) = (E_{ideal}, Q_{ideal}) \) is the unattainable ideal goal. Therefore, the proposed approach attempts to seek solution \( S \), which is not only closer to the unattainable ideal goal \((E_{ideal}, Q_{ideal}) \) but also has the most valuable trade-off among pareto optimal solutions in accordance with users' requirements.

The slope of the straight line connecting \( S_p \) and \( S_q \), shown in Fig. 3, is

\[ m = -\frac{Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}} \]

Here, \( m \) represents the average trade-off ratio between these two objective functions.

For point \( P_b \) as shown in Fig. 3, the corresponding values of \( E(P_b) \) and \( Q(P_b) \) are \( E_1 \) and \( Q_1 \), respectively. With the bi-objective optimization concept, there would be no advantage between \( P_1 \) and \( P_2 \), since \( P_2 \) has \( E_{advance} \) with the deterioration of \( E_{expense} \) where the ratio of advance and expense is the same as with the average trade-off ratio m. Therefore, \( P_2 \) is located in a line that has the same slope as that which forms \( S_p \) and \( S_q \), so that:

\[ \frac{Q_1 - Q_2}{E_2 - E_1} = \frac{Q_{advance}}{E_{expense}} = m = \frac{Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}} \] (8)

\( P_1 \) and \( P_2 \) determine a line that has slope \(-m\) and intercept \( b \) in \( Q(S) \) axis. All of the points in this line will have similar performance for bi-objective programming based on mathematical explanation. The line equation can be formulated as below:

\[ Q_1 = -mE_1 + b = \frac{Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}}E_1 + b \] (9)

\[ b = Q_1 + \frac{Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}}E_1 \] (10)

Solution \( S \) caused a smaller \( b \) in (10) and represents a better result because it is closer to the unattainable ideal goal. Therefore, to find a solution \( S \) that can have \( b \) will be a most valuable trade-off solution of bi-objective programming, as shown in Fig. 4. The valuable trade-off solution for bi-objective programming is then formulated as below:

\[ \min b = Q(S) + \frac{Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}}E(S) \] (11)

However, if DMs still have a preference for the dedicated objective, it only needs straightforward consideration of the percentage of favorites objective with a simple weight \( w \). Suppose DMs think that \( Q \) is more important than \( E \) for the final result, for example \( w = 80\% \); this means that 80\% expense of \( E \) is worth 20\% improvement of \( Q \) as shown below:

\[ m' = \frac{w \times Q_{advance}}{(1 - w) \times E_{expense}} = \frac{w}{1 - w} \]

\[ m = \frac{w \times Q_{nonideal} - Q_{ideal}}{E_{nonideal} - E_{ideal}} \] (12)
A similar formulation process as (8)–(10) then changes the bi-objective programming, as below, to take DM’s willingness into consideration easily:

$$\min b = Q(\bar{S}) + \frac{w}{1-w} Q_{\text{nonideal}} - Q_{\text{ideal}} E(\bar{S})$$  \hspace{1cm} (13)

### 3.2. Solution procedure of IBVT

According to the principle and discussion above, this section describes the solution process of the IBVT method. Its three steps are divided and shown as follows.

(a) Step 1:

Solve single objective functions (14) and (15) below to get \( S_E, E_{\text{ideal}}, Q_{\text{nonideal}} \) and \( S, Q_{\text{ideal}}, E_{\text{ideal}}, Q_{\text{nonideal}} \) respectively.

$$\min E(\bar{S}) \text{ subject to all constraints}$$  \hspace{1cm} (14)

$$\min Q(\bar{S}) \text{ subject to all constraints}$$  \hspace{1cm} (15)

(b) Step 2:

Solve objective function (16), as below, to get the valuable trade-off solution that should satisfy all constraints. Suppose that \( S \) is solved, as shown in Fig. 5.

$$\begin{align*}
\min b(\bar{S}) &= Q(\bar{S}) + \frac{w}{1-w} Q_{\text{nonideal}} - Q_{\text{ideal}} E(\bar{S}) \\
&\text{subject to all constraints}
\end{align*}$$  \hspace{1cm} (16)

\( S \) divides the shaded part in Fig. 5 into areas I, II, III and IV. Area III can be called the “unreachable solution space” since no solutions could be obtained in this area. However, the solutions in area I are even worse than \( S \) so there is no need to consider this area. The pareto optimal front in area II possesses one characteristic in which \( E(\text{Area II}) \) is superior to \( E_i \) but \( Q(\text{Area II}) \) is worse than \( Q_i \). Similar situations also occur in area IV. This characteristic can help users searching for preferred solutions when considering different policies to be processed in the next step.

(c) Step 3:

To reach an interactive relation with users, if DMs are not satisfied with \( S \) in Step 2, one objective function may be sacrificed to improve another objective function as follows. If the intention is to further improve \( E(\bar{S}) \), \( E_i \) obtained in Step 2 is regarded as the new \( E_{\text{nonideal}} \) and \( Q \) is considered as the new \( Q_{\text{ideal}} \) as shown in (17). The corresponding solution space is reduced to area II and Step 2 is repeated to find the valuable trade-off solution within the preferred boundary, as shown in Fig. 6. On the other hand, if the intention is to further improve \( Q(\bar{S}) \), \( Q_i \) obtained in Step 2 is regarded as the new \( Q_{\text{nonideal}} \) and \( E_i \) is considered as the new \( E_{\text{ideal}} \).

$$\begin{align*}
\min b(\bar{S}) &= Q(\bar{S}) + \frac{w}{1-w} Q_{\text{nonideal}} - Q_{\text{ideal}} E(\bar{S}) \\
&\text{subject to all constraints}
\end{align*}$$  \hspace{1cm} (16)

$$\begin{align*}
\begin{cases}
E_{\text{nonideal}} &= E_i \\
Q_{\text{ideal}} &= Q_i
\end{cases}
\end{align*}$$  \hspace{1cm} (17)

As shown from the above discussion, solving direction conforms to users’ requirements. Users only need to input the objective function intended to be improved and the proposed approach can find a valuable trade-off solution meeting that requirement without considering any weighting value. As shown in Fig. 6, the whole space of feasible solution can automatically reduce its range according to the DMs’ requirements. Each time the search is done, the space gradually turns to the DM-appointed direction. As each search may reduce the space of feasible solutions, \( Dis_E \) and \( Dis_Q \) provided in the research are shown in (18), as reference for determining the next search. As shown in Fig. 6, \( Dis_E \) and \( Dis_Q \) in (18) represent the ranges of \( E \) and \( Q \), respectively, in space of feasible solution. If \( Dis_E \) and \( Dis_Q \) are too small, there will not be a significant difference of solution in the next search. Therefore, the next search may not be necessary. However, if \( Dis_E \) and \( Dis_Q \) remain large enough, there might be a different result in the next search that makes it worth doing.

$$\begin{align*}
\begin{cases}
\text{Dis}_E &= E_{\text{nonideal}} - E_{\text{ideal}} \\
\text{Dis}_Q &= Q_{\text{nonideal}} - Q_{\text{ideal}}
\end{cases}
\end{align*}$$  \hspace{1cm} (18)

The three steps as stated above constitute the IBVT method. It is obvious that, in the whole process, users only need to choose a favorite objective function. It is unnecessary to consider the selection of weighting value or unit difference of objective functions. Thus, the work for DMs is straightforward and solution efficiency can be effectively increased. The main advantage of IBVT is in providing a larger programming space in the modeling process, as it is not limited to single objective programming. The problems of bi-objective programming can be easily solved so as to provide users with more favorable solutions and a more convenient operating environment. Fig. 7 is the complete flowchart of the IBVT method for bi-objective programming.

### 4. SA–PSO method

The SA–PSO approach that combines the advantages of both SA and PSO methods is proposed and described in this section.
4.1. Brief review of standard PSO and SA

PSO search procedures are based on the swarm concept, which is a group of individuals that are able to optimize a certain fitness function. Each individual knows its best value (Pbest) and position at any given time. Moreover, each individual can refer to the best value (Gbest) and its position in the group. The modified velocity of each individual for the next movement can be calculated using its current velocity and distance from Pbest and Gbest, as shown below:

\[
\begin{align*}
\mathbf{v}_{i}^{k+1} &= \omega \mathbf{v}_{i}^{k} + c_1 \times \text{Rand} \left( \right) \times \left( \text{Pbest}^k - s_i^k \right) + c_2 \times \text{Rand} \left( \right) \times \left( \text{Gbest}^k - s_i^k \right) \\
\mathbf{v}_{i}^{k+1} &= \begin{cases} 
\mathbf{v}_{i}^{k} & |\mathbf{v}_{i}^{k+1}| < v_{\text{max}} \\
\mathbf{v}_{\text{max}} & |\mathbf{v}_{i}^{k+1}| \geq v_{\text{max}} \\
-\mathbf{v}_{\text{max}} & |\mathbf{v}_{i}^{k+1}| \leq -v_{\text{max}} 
\end{cases}
\end{align*}
\]

(19)

(20)

\[
\omega = \omega_{\text{max}} - \frac{\omega_{\text{max}} - \omega_{\text{min}}}{\text{Iter}_{\text{max}}} \times \text{Iter}
\]

(21)

Parameter \(v_{\text{max}}\), shown in (20), determines the resolution or boundary, which identifies the region to be searched from the present position. If \(v_{\text{max}}\) is set too high, particles may pass good solutions. If \(v_{\text{max}}\) is too small, particles may not be able to explore beyond local solutions. In addition, the constants \(c_1\) and \(c_2\) in (19) represent the weight of the stochastic acceleration terms which pull each individual towards the Pbest and Gbest positions. Low-weight value settings will allow each individual to search far beyond local optimal regions before being pulled back if no global optimal solution is found. On the other hand, high-weight values will result in premature convergence, thereby trapping it into a local optimal. Suitable selection of inertia weight \(\omega\), as shown in (21), provides a balance between global and local explorations, thus requiring less iteration to locate the optimal global solution. Generally, the above-mentioned parameters are set by simulation experiences, for example: \(v_{\text{max}}\) was often set to half of the entire searching space, \(c_1\) and \(c_2\) to 2.0, and \(\omega\) often decreases linearly from approximately 0.9 to 0.4 during a run, as shown in (21).

Therefore, the new position can be easily calculated by (22), as stated below:

\[
s_i^{k+1} = s_i^k + \mathbf{v}_{i}^{k+1}
\]

(22)

Simulated annealing (SA) [25] belongs to a stochastic search algorithm with jumping properties and is motivated by the similarity between the solids’ annealing procedure and optimization problems. The most significant characteristic of SA is the probabilistic jumping property, called the metropolis process (i.e., a worse solution has a probability factor of being accepted as the new solution). However, by adjusting the temperature, the metropolis process for jumping probability can be controlled. In particular, probability is high when the temperature is high (at the beginning of search process) and decreases as the temperature decreases. It has been theoretically proven that, under certain conditions, SA is globally convergent in probability ONE [23,24]. The two main procedures for SA are the metropolis process and cooling schedule, as stated below.

The metropolis process [26] randomly generates a new state for \(S' \in \mathcal{N}_s\) and calculates the difference by \(\Delta F = F(S') - F(S)\), where \(F()\) is a fitness function, then randomly selects a number \(R_0 \in [0, 1]\). When \(\Delta F > 0\), it means the new state was improved for increasing fitness function. Then, state \(S'\) was accepted as \(S\) (i.e., \(S = S'\)). Otherwise, state \(S\) was accepted according to the criterion of \(\exp(\Delta F/T_i)\) > \(R\). Where \(T_i\) is the temperature of SA. The left-hand side of this criterion is the Boltzmann distribution probability. This acceptance criterion is generally known as the Metropolis criterion.

The cooling schedule gradually decreases the temperature value of \(T_i\) by \(T_{i+1} = \alpha \times T_i\), where \(\alpha\) represents the cooling rate, and is repeated until the convergence criterion is reached.

4.2. SA–PSO procedure

As with the GA method, the main drawback of SA is ease of premature convergence. One reason is that, in PSO, Gbest and Pbest are used, as shown in (19). Consequently, all particles have the tendency to fly to the current best solution, which may be a local optimum, or to a solution near local optimum. Therefore, all particles will concentrate in a small region, thereby weakening the global exploration ability. This means that if Pbest is not the global optimum, the algorithm evolved from (19) may be trapped in a local optimum. Therefore, a modified optimization strategy, named SA–PSO, which incorporates the mechanism of SA with PSO is proposed. As previously mentioned, in a standard PSO, movement is entirely determined by (19) without any conditional probabilities and, once Pbest and Gbest are trapped in a local optimum, it is difficult to escape from. Thus, modification to movement–acceptance procedures, by adding an SA-like cooling scheme and a metropolis-probability decision process, are helpful in remedying premature convergence.

The entire concept of the proposed SA–PSO is depicted in Fig. 8.

The population size of this example is \(P\), which has been sorted by fitness values. The elite of the iterations will be compared and stored...
as the final result. This step may remedy the weakness of PSO and guarantee discovery of the best final solution from the whole search process. Following this step, P particles are generated by the PSO movement algorithm and then combined with an SA judgment operator. Each particle of the previous generation will be chosen to calculate the velocities for the next movement. The metropolis movement algorithm and then combined with an SA judgment operator. Each particle of the previous generation will be chosen to calculate the velocities for the next movement. The metropolis process of SA will calculate whether or not the determined movement has been accepted. If it fails to pass through as the next movement, it will then be recalculated at a new velocity by (19). Due to the random properties of (19), the new value will be different from the previous value. This process will be applied to each particle until all P particles are generated into the next movement, producing a new generation. Finally, a sorting process will be applied to rank these new particles by fitness values. This process will iterate with the cooling schedule of the SA until the final criterion has been reached.

The whole procedure of SA–PSO is described as follows:

- **Step 1: initialization:**
  - Step 1.1: initialize P particles using a random generator. Let \( k = 0 \), and randomly initialize \( s_i^0 \) and \( v_i^0 \), \( i = 1, \ldots, n \).
  - Step 1.2: evaluate the fitness value of all particles, which determines \( Gbest \) and \( Pbest \) by a simple comparison of their fitness values.
  - Step 1.3: set all parameters, including \( c_1, c_2, v_{\text{max}}, v_{\text{min}}, \text{Iter}_{\text{max}}, v_0, \text{initial temperature} T_0, \) and cooling rate \( \alpha \).
- **Step 2: repeat** this step until the stopping criterion is satisfied:
  - Step 2.1: calculate velocities \( v_i^{k+1} \) for each particle by (19), which include random search phenomena.
  - Step 2.2: calculate the fitness of each particle \( s_i^{k+1} \) for each particle by (19), which include random search phenomena.
  - Step 2.3: evaluate \( \Delta \text{Fitness} = \text{Fitness}(s_i^{k+1}) - \text{Fitness}(s_i^k) \), and then randomly select a number \( R_0 \in [0, 1] \). If \( \Delta \text{Fitness} > 0 \), meaning that the new position is improved for increasing fitness function, then position \( s_i^{k+1} \) is accepted as the new position of particle \( i \). Otherwise, \( s_i^{k+1} \) is accepted, according to the following criterion: \( \exp(\Delta \text{Fitness} / T_i) > R \). Proceed to Step 2.4 when the velocity of all particles is determined, or return to Step 2.1 for those particles failing to be accepted, and generate new velocities using the same evaluation process. Too many failures (i.e., 100 in our study) for the same particle will force the last velocity to be accepted, in consideration of computational (CPU) time, and thereby prevent entering into an endless loop. However, this did not occur during our study and simulation.
  - Step 2.4: renew each particle to the new position and modify \( Gbest \) and \( Pbest \) by simple comparison of their fitness values.
  - Step 2.5: when the evolution process has achieved a satisfactory condition (or maximum evolution number is reached), proceed to Step 3; otherwise, modify inertia weight \( \omega \) and annealing temperature \( T_{i+1} = \alpha \times T_i \), let \( k = k + 1 \), and return to Step 2.
- **Step 3: output** the best solution \( Gbest \) and its fitness value.

**Remark 1:** PSO has an inherent parallel search structure with easy implementation and only slight modifications are required in programming SA–PSO.

**Remark 2:** by adjusting the temperature, the SA-based selection can be controlled. In the early stages of evolution (i.e., the temperature is high) a particle’s position in a new PSO generation may have a high probability of being passed as the next move. SA–PSO shows strong global exploration over the entire solution space. As the temperature decreases, the probability of selecting improved solutions is greater than that of selecting worse solution. Thus, SA–PSO adjusts its search behavior by balancing exploration and exploitation. When the temperature is close to 0, almost all enabled movement is improved movement, stressing exploitation to locate good solutions in promising regions.

**Remark 3:** finally, the proposed SA–PSO remains a general optimization algorithm which can be applied to any optimization problem. In the next section, we apply this approach to a practical reactive energy scheduling problem.

### 5. Implementation by the proposed approach

SA–PSO is used as the searching tool for solving IBVT in this paper; the detailed implementation of which is discussed below.

#### 5.1. Representation of an individual particle

Implementation of a problem in the SA–PSO framework starts from the representation of the problem. In this study, integer representation is chosen for each particle and the individual particle structure is represented in Fig. 9. Parameter \( S_i^k \) describes the status of capacitor bank \( C_i \) at hour \( h \), where “1” and “0” represent “ON” and “OFF”, respectively. Note that each capacitor bank may have a different capacity \( Q_i \). The value of each particle’s position should be limited so it is not violating expansion constraints \( N_i \).

#### 5.2. Fitness function

Implementation of an optimization problem in SA–PSO is realized within the evolutionary process of a fitness function. The fitness function adopted is based on the objective corresponding to each step mentioned in Section 4. To emphasize the “best” particle

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>0.85</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>100</td>
</tr>
<tr>
<td>Pop. size</td>
<td>100</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>2</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>0.4</td>
</tr>
<tr>
<td>( v_{\text{min}} )</td>
<td>0.9</td>
</tr>
<tr>
<td>( v_{\text{max}} )</td>
<td>216</td>
</tr>
<tr>
<td>( \text{Iter}_{\text{max}} )</td>
<td>300</td>
</tr>
<tr>
<td>( V_{\text{min}} )</td>
<td>0.9 pu</td>
</tr>
<tr>
<td>( V_{\text{max}} )</td>
<td>105 pu</td>
</tr>
<tr>
<td>( p_{\text{max}} )</td>
<td>600 A</td>
</tr>
<tr>
<td>( Q_{\text{min}} )</td>
<td>300 kVar</td>
</tr>
</tbody>
</table>
and speed up convergence of the evolutionary process, the normalized objective value is defined as stated in (23) and (24). If any constraint is violated then a “1” will be given to the Penalty term, otherwise “0” is selected.

\[
\text{Fitness} = \frac{1}{N_{\text{Obj}}(S) + \text{Penalty}} 
\] (23)

\[
N_{\text{Obj}}(S) = 1 + \frac{(\text{Obj}(S) - \text{Obj}_{\text{min}})}{(\text{Obj}_{\text{max}} - \text{Obj}_{\text{min}})} 
\] (24)

6. Case studies with convergence and validation analysis

To illustrate the performance of the proposed solution methodology, we used a practical twelve-bus, 11.4 kV distribution feeder, as shown in Fig. 1, which is a portion of the Taiwan Power Company’s distribution system. There are 18 switched capacitors and 24 h scheduling should be determined. Therefore, the total searching space will have $2^{18 \times 24} \approx 1.11 \times 10^{130}$ combination, which is too large to get the actual global optimum. The line and forecast load data are compiled from the Taipei West District Office of the Taiwan Power Company. Table 1 shows the parameter setting for the implementation of the proposed approach and constraints setting according to the current practice in Taiwan.

There exist several parameters should be determined for the implementation of the SA–PSO. The constants $c_1$ and $c_2$ represent the weighting of the stochastic acceleration terms that pull each particle toward $P_{\text{best}}$ and $G_{\text{best}}$ positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, target regions. Generally, these parameters should be determined through the experiments of the target test systems. To avoid the problem of

<table>
<thead>
<tr>
<th>Table 2 Numerical Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$Q(S)$ (times)</td>
</tr>
<tr>
<td>$\text{max} V^h (\text{p.u.})$</td>
</tr>
<tr>
<td>$\text{min} V^h (\text{p.u.})$</td>
</tr>
<tr>
<td>$SR_E$ (%)</td>
</tr>
<tr>
<td>$SR_Q$ (%)</td>
</tr>
<tr>
<td>Satisfactory?</td>
</tr>
<tr>
<td>Trade-off term?</td>
</tr>
<tr>
<td>$E_{\text{ideal}}$</td>
</tr>
<tr>
<td>$E_{\text{nonideal}}$</td>
</tr>
<tr>
<td>$Q_{\text{ideal}}$</td>
</tr>
<tr>
<td>$Q_{\text{nonideal}}$</td>
</tr>
<tr>
<td>$\text{Dis}_{E}$</td>
</tr>
<tr>
<td>$\text{Dis}_{Q}$</td>
</tr>
<tr>
<td>Continue?</td>
</tr>
</tbody>
</table>


the curse of dimensionality, the procedures and strategies can be determined following the heuristic suggestion of Refs. [21,29,30]. (1) The values of $c_1$ and $c_2$ have the same value, which implies the same weights are given between $P_{best}$ and $G_{best}$ in the evolution processes. (2) The values of $v_{max}$ and $\theta_{max}$ are varied from 1.0 to 0.5 and from 0.5 to 0.1. Therefore, through several simulation tests under different combination of parameters, the ideal or near ideal setting of parameters that provide better solution quality and efficiency can then be got. In this paper, the above-mentioned parameters are set by simulation experiences, $c_1$ and $c_2$ were set to 2.0, and $\alpha$ decreases linearly from approximately 0.9 to 0.4 during each run.

The parameter $v_{max}$ determined the resolution, or fitness, with which regions are to be searched between the present position and the target position. If $v_{max}$ is too high, particles might fly past good solutions. If $v_{max}$ is too small, particles may not explore sufficiently beyond local solutions. Generally, $v_{max}$ should have been determined through the experiments on the test systems. In my experiences with PSO, $v_{max}$ was often set at 40–60% of the range of the variable on each dimension. Some literatures even use linear decreases $v_{max}$ that just like inertia weight $W$ in (21). The population size and iteration time are almost linearly with the CPU time. Therefore, the main concern of these two parameters is how much time this application can have. However, they still can get a better setting through experiments. In this paper, the population size and iteration time are set as 100 and 3000 individually.

The temperature parameter of SA–PSO will affect the acceptance ratio of particles that are worse than before. In the beginning of searching process, exploration is more important for wildly searching and escape from local minimum. After that, a fine tuning process called exploitation to converge the searching to an optimal solution should be following. The value $\alpha$ in this paper is selected through some simulation experiments of the target test system. Also, a typical value from 0.95 to 0.8 is the most recommended from literatures. Therefore, a 0.85-cooling rate is chose in this paper.

### 6.1. Simulation cases and results

The test results are summarized in Table 2. The satisfaction rates for both the daily line loss and the number of switching operations are defined in (25), which represent the level of satisfaction within the attainable search region.

Satisfaction Rate of \( \frac{Q}{E} \) = \[ \frac{SR_Q}{SR_E} \] = \[ \frac{\max \left\{ \frac{E}{Q} \right\} - \min \left\{ \frac{E}{Q} \right\} }{\max \left\{ \frac{E}{Q} \right\} - \min \left\{ \frac{E}{Q} \right\} } \] \tag{25}

The second and third columns correspond to a single objective programming that minimizes daily line loss and switching operations, respectively. Columns four to eight show the results of the proposed procedures that consider both daily line loss and switching operations. In the third column, to achieve the minimum daily line loss, the number of switching operations required is as large as 64 and the daily loss is 3,281,756 kWh. In the second column, if all capacitors are kept off throughout the day, the daily line loss is found to be 5,691,874 kWh. These two results indicate the extreme solutions of the test system.

The fourth column represents the first result of the proposed algorithm. Comparing the fourth column with the third column, it is obvious that $SR_Q$ degraded slightly from 100% to 64.3%, while $SR_E$ greatly improved from 0% to 79.7%. Obviously, it is worth performing such capacitor dispatches.

The suitability of Result 1 should be judged by the DMs of electricity utilities. If they think that Result 1 is not suitable for the policy of the utilities, then further compromise can be made according to their directions as dictated. Unlike other approaches that indicate many unknown parameters such as weight values for further search, the DMs only have to choose one of the objectives as the trade-off (compromised) term and then the searching procedure continues to find another best-compromise and desirable solution for the bi-objective problem. If it is decided to take more switching operations for daily line loss reduction, the number of switching operations is selected as the trade-off term. The parameters of $E_{ideal}$ and $Q_{ideal}$ in Result 1 are then modified for the next search. Note that the decision region is bounded by the ideal and nonideal values of both $Q$ and $E$, such that it shifts toward the region of interest as indicated by the DMs. The values $Dis_E$ and $Dis_Q$ can help the DMs to understand the maximum improvement that
a further step can achieve. If they think that the maximum improvement in the desired term is too small to make further searching worthwhile, then they can stop the process. Assuming that the further step is allowed by the DMs, Result 2 shows the consequent result. Again, it is a flexible and valuable trade-off solution within the decision region. However, if the DMs thought that the switching times should be further reduced, a similar procedure can be easily applied, as shown in Results 2.

Continuing from Result 2, if further steps in the search for loss reduction are allowed by the DMs, Result 3 shows the consequent result. Gradually, the decision region will become smaller and more focused on the intention of the DMs. SRQ has degraded from 67.2% to 34.4%, while SR has improved from 77.0% to 91.5%, respectively. Note that all the constraints are satisfied by the proposed method.

The current practice of the Taiwan Power Company is to dispatch the switched capacitors by a fixed-schedule basis. For comparison purpose, Fig. 10 lists the daily line loss as well as the number of switching operations for both the fixed-schedule dispatch and the proposed dispatch. From strategies “5~8”, daily line loss is reduced if more switching operations are performed. However, as the number of switching operations is greater than a threshold value, it will result in saturation effects concerning loss reduction. For example, when the number of switching operations increases from 42 to 64, only approximately 6% loss reduction is produced in reward. Comparing strategies “1~4” with “5~6”, it is obvious from both the daily line loss obtained and switching operations required that the proposed capacitor dispatch is superior to the fixed-schedule capacitor dispatches. Also, comparing strategy “7” with strategy “3”, the daily line loss can be significantly reduced with only a slight increase in switching operations.

It is obvious that the proposed SA–PSO can have a better solution quality and faster convergence property.

6.3. Validation analysis

To demonstrate the promising results of SA–PSO and the validation of IBVT about finding the valuable trade-off solutions, three simulation processes to verify the salient features of this paper are conducted below.

Table 3 shows the comparative simulation results of SA–PSO with GA [27] and PSO [28]. The results show that the SA–PSO method can offer a better solution quality than PSO and GA methods. The CPU time of SA–PSO have a small range variation compared to GA or PSO that is because the SA process has some rejection rate during solution process. Also, CPU time consuming for SA–PSO is longer than PSO or GA in our simulation experience. The reactive energy scheduling problem is part of daily operations but not a real time operating problem. Hence, the CPU time consuming for SA–PSO is still within an acceptable range in this application.

### Table 4
Benchmark simulation for comparative studies.

<table>
<thead>
<tr>
<th>Test functions</th>
<th>Solution methodologies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA–PSO</td>
</tr>
<tr>
<td>Function 1 (2 variables)</td>
<td></td>
</tr>
<tr>
<td>Success rate (%)</td>
<td>100</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0</td>
</tr>
<tr>
<td>Average CPU time (s)</td>
<td>14.2</td>
</tr>
<tr>
<td>Function 2 (8 variables)</td>
<td></td>
</tr>
<tr>
<td>Success rate (%)</td>
<td>100</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0</td>
</tr>
<tr>
<td>Average CPU time (s)</td>
<td>61.4</td>
</tr>
<tr>
<td>Function 3 (50 variables)</td>
<td></td>
</tr>
<tr>
<td>Success rate (%)</td>
<td>85</td>
</tr>
<tr>
<td>Mean</td>
<td>0.758</td>
</tr>
<tr>
<td>Average CPU time (s)</td>
<td>508.2</td>
</tr>
</tbody>
</table>
solution which is more efficient and of higher quality and one system. The encouraging simulation results have shown that the fixed-schedule dispatch currently adopted by the Taipower has been demonstrated by a Taipower feeder and proven to have superior features, including high-quality solutions, stable quality were considered simultaneously. The proposed approach constraints of the system together with daily load variation processes. To obtain a more realistic solution, the practical procedures are proposed and proven by systematic simulation.

7. Conclusion

A new approach to address reactive energy scheduling issues has been presented. The novel IBVT and hybrid SA–PSO procedure are proposed and proven by systematic simulation processes. To obtain a more realistic solution, the practical constraints of the system together with daily load variation have been considered. The two objectives of economic and quality were considered simultaneously. The proposed approach has been demonstrated by a Taipower feeder and proven to have superior features, including high-quality solutions, stable convergence characteristics, and good computational accuracy. The salient feature of the proposed approach is that it can provide a set of flexible and valuable trade-off solutions to help system operators determine the optimal reactive energy scheduling. Results obtained show that both the daily line loss and the number of switching operations can be greatly reduced with proper dispatch of switched capacitors in comparison with the fixed-schedule dispatch currently adopted by the Taipower system. The encouraging simulation results have shown that the proposed method is capable of finding a valuable trade-off solution which is more efficient and of higher quality and one that can be easily and effectively dictated by the DMs.

Furthermore, by using the new SA–PSO procedure proposed in this paper, the solution quality discovered can be further improved.

Acknowledgment

The financial support by the National Science Council of the Republic of China under Grant No: NSC96-2622-E-129-006-CC3 is gratefully acknowledged.

Appendix

The 3 test functions and their global optimum employed are given below:

Function 1 (2 Variables): Beale function
\[
f(X) = \sum_{i=1}^{3} \left( y_i - x_i(1 - x_i)^2 \right)^2, \quad y_1 = 1.5, \quad y_2 = 2.25, \quad y_3 = 2.625.
\]
Global optimum with \( f(X) = 0 \) at \((x_1 = 3, x_2 = 0.5)\)

Function 2 (8 Variables): Extended Powell function
\[
f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2}, \quad i = 1, 2
\]
\[
f_{4i-2}(x) = 5^{1/2}(x_{4i-1} - x_{4i}), \quad i = 1, 2
\]
\[
f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2, \quad i = 1, 2
\]
\[
f_i(x) = 10^{1/2}(x_{4i-3} - x_{4i})^2, \quad i = 1, 2
\]
\[
f(X) = \sum_{i=1}^{8} f_i^2(x)
\]
Global optimum with \( f(X) = 0 \) at the \((x_i = 0, i = 1–8)\).

Function 3 (50 variables): Griewank function
\[
f(X) = \sum_{i=1}^{50} \left( \frac{x^2_i}{4000} - \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) + 1
\]
Global optimum with \( f(X) = 0 \) at the \((x_i = 0, i = 1–50)\).

References