Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A pooled-neighbor swarm intelligence approach to optimal reactive power dispatch*

GUO Chuang-xin†, ZHAO Bo
(*School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)
†E-mail: guochuangxin@vip.sina.com
Received Oct. 20, 2005; revision accepted Feb. 21, 2006

Abstract: This paper presents a pooled-neighbor swarm intelligence approach (PNSIA) to optimal reactive power dispatch and voltage control of power systems. The proposed approach uses more particles' information to control the mutation operation. The proposed PNSIA algorithm is also extended to handle mixed variables, such as transformer taps and reactive power source installation, using a simple scheme. PNSIA applied for optimal power system reactive power dispatch is evaluated on an IEEE 30-bus power system and a practical 118-bus power system in which the control of bus voltages, tap position of transformers and reactive power sources are involved to minimize the transmission loss of the power system. Simulation results showed that the proposed approach is superior to current methods for finding the optimal solution, in terms of both solution quality and algorithm robustness.

Key words: Reactive power dispatch, Swarm intelligence, Multi-agent systems, Global optimization
doi:10.1631/jzus.2006.A0615 **Document code:** A **CLC number:** TM73; TM74

INTRODUCTION

The reactive power dispatch is aimed at minimizing the active power loss in the transmission network by allocating the reactive power generation under several security constraints. The reactive power dispatch problem has significant influence on secure and economic operation of power systems. The reactive power generation affects the overall generation cost via transmission loss. A procedure which allocates the reactive power generation so as to minimize the transmission loss, will consequently result in the lowest production cost. Obviously, this problem is in nature a global optimization problem, which may have several local minima with the conventional optimization methods easily leading to local optimum. Conventional gradient-based optimization algorithms widely used to solve this problem for decades (Dommel and Tinney, 1968; Hong *et al*., 1990) have many mathematical assumptions (such as analytic and differential properties of the objective functions and unique minima existing in problem domains) which have to be given to simplify the problem, otherwise it is very difficult to calculate the gradient variables in the conventional methods. Furthermore, in practical power system operation, the data acquired by the SCADA (Supervisory Control and Data Acquisition) system are contaminated by noise. Such data may cause difficulties in computation of gradients. Consequently, the optimization could not be carried out on many occasions.

In the last decade, many new stochastic search methods have been developed for the global optimization problems, such as simulated annealing, genetic algorithms and evolutionary programming. Genetic Algorithms (GAs) are a class of stochastic search algorithms that start with a population of randomly generated candidates and 'evolve' towards better solutions by applying genetic operators (crossover, mutation, selection, etc.), modelled on the genetic processes occurring in nature. Evolutionary compu-

* Project supported by the National Natural Science Foundation of China (No. 60421002) and the Outstanding Young Research Investigator Fund (No. 60225006), China

tation techniques have recently found many applications in power systems, especially in the economic operation area (Yin and Germay, 1991; Iba, 1994; Wu *et al*., 1998; Wong and Wong, 1994; Lee *et al*., 1995). A reliable global optimization approach to reactive power dispatch problem has always been of considerable value to both secure and economical operation of power systems.

In this paper, a novel approach for solving optimal reactive power dispatch and voltage control problem of power systems has been developed, based on the recently introduced swarm intelligence algorithms. Swarm intelligence, as demonstrated by natural biological swarms, exhibits numerous powerful features that are desirable in many engineering systems and can be applied to nonlinear and non-continuous optimization problems (Bonabeau *et al*., 1999; Dorigo *et al*., 1996; Kennedy and Eberhart, 1995). The swarm intelligence technique can generate high-quality solutions in shorter calculation time and stabler convergence characteristic than other stochastic methods. This work is devoted to developing a pooled-neighbor swarm intelligence approach (PNSIA) to optimal reactive power dispatch and voltage control of power systems. The convergence property of the proposed PNSIA is analyzed using standard results from dynamic system theory and guidelines for proper algorithm parameter selection are derived. A new adaptive strategy for choosing parameters is also proposed to assure PNSIA method convergence. The proposed PNSIA algorithm is also extended to handle mixed variables, such as transformer taps and reactive power source installation, using a simple scheme. The PNSIA is evaluated on an IEEE 30-bus power system and a practical 118-bus power system in which the control of bus voltages, tap position of transformers and reactive power sources are involved to minimize the power transmission loss. Simulation results showed that the proposed approach is superior to current methods for finding the best solution, in terms of both solution quality and algorithm robustness.

MATHEMATICAL FORMULATION OF OPTIMAL REACTIVE POWER DISPATCH

The objective of the reactive power dispatch is to minimize the active power loss in the transmission network, which can be described as follows:

$$f_{\mathrm{Q}}=\sum_{k\in N_{\mathrm{E}}}P_{k\mathrm{loss}}=\sum_{k\in N_{\mathrm{E}}}g_k(V_i^2+V_j^2-2V_iV_j\cos\theta_{ij}),\qquad(1)$$

where $k=(i,j)$, $i\in N_{\mathrm{B}}$, $j\in N_i$. The symbols in Eq.(1) and in the following context are given in (Wu *et al*., 1998). The minimization of the above function is subject to a number of constraints:

$$0=P_{G_i}-P_{D_i}-V_i\sum_{j\in N_i}[V_j(G_{ij}\cos\theta_{ij}+B_{ij}\sin\theta_{ij})],\ i\in N_0,\qquad(2)$$

$$0=Q_{G_i}-Q_{D_i}-V_i\sum_{j\in N_i}[V_j(G_{ij}\sin\theta_{ij}-B_{ij}\cos\theta_{ij})],\ i\in N_{\mathrm{PQ}},\qquad(3)$$

and

$$V_i^{\min}\le V_i\le V_i^{\max},\ i\in N_{\mathrm{B}};\ T_k^{\min}\le T_k\le T_k^{\max},\ k\in N_{\mathrm{T}};$$
$$Q_{G_i}^{\min}\le Q_{G_i}\le Q_{G_i}^{\max},\ i\in N_{\mathrm{G}};\ Q_{C_i}^{\min}\le Q_{C_i}\le Q_{C_i}^{\max},\ i\in N_{\mathrm{C}},$$

where power flow equations are used as equality, constraints, reactive power source installation restrictions, reactive power generation restrictions and transformer tap-setting restrictions, and bus voltage restrictions are used as inequality constraints.

In most nonlinear optimization problems, the constraints are considered by generalizing the objective function using penalty terms. In the reactive power dispatch problem, the generator bus voltages, V_{PV} and V_{S}, the tap position of transformer, T, the amount of the reactive power source installation Q_{C}, are control variables which are self-constrained. Voltages of PQ-bus, V_{PQ}, and injected reactive power of PV-bus, Q_{G}, are constrained by adding them as penalty terms to the objective function [Eq.(1)]. The above problem is generalized as follows:

$$F_{\mathrm{Q}}=f_{\mathrm{Q}}+\sum_{i\in N_{\mathrm{V}}^{\lim}}[\lambda_{V_i}(V_i-V_i^{\lim})^2]+\sum_{i\in N_{\mathrm{Q}}^{\lim}}[\lambda_{G_i}(Q_{G_i}-Q_{G_i}^{\lim})^2],\qquad(4)$$

where λ_{V_i} and λ_{G_i} are the penalty factors, $V_i^{\lim}$ and $Q_{G_i}^{\lim}$ are defined as

$$V_i^{\lim}=\begin{cases}V_i^{\max}; & V_i>V_i^{\max},\\ V_i^{\min}; & V_i<V_i^{\min},\end{cases}\qquad(5)$$

$$Q_{G_i}^{\lim} = \begin{cases} Q_{G_i}^{\max}; & Q_{G_i} > Q_{G_i}^{\max}, \\ Q_{G_i}^{\min}; & Q_{G_i} < Q_{G_i}^{\min}. \end{cases} \quad (6)$$

POOLED-NEIGHBOR SWARM INTELLIGENCE APPROACH (PNSIA)

Particle swarm optimization approach (PSO)

Swarm intelligence appears in biological swarms of certain insect species (Bonabeau *et al*., 1999; Dorigo *et al*., 1996) and gives rise to complex and often intelligent behavior through complex interaction of thousands of autonomous swarm members. Interaction is based on primitive instincts with no supervision. The end result is accomplishment of very complex forms of social behavior and fulfillment of a number of optimization and other tasks. The main principle behind these interactions is called stigmergy, or communication through the environment. Recently, based on the simulation of bird swarm, Kennedy and Eberhart (1995; 2001) developed a particle swarm optimization (PSO) concept mainly based on simulation of bird flocking in 2D space. According to the research results for a flock of birds, birds find food by flocking (not by each individual). The observation leads to the assumption that every information is shared inside flocking. Moreover, according to observation of human groups behavior, the behavior of each individual (agent) is also based on group decided behavior patterns such as customs and other behavior patterns based on the experiences of each individual. The position of each agent is represented by the *xy*-axis position and the velocity (displacement vector) is expressed by v_x (the velocity of x-axis) and v_y (the velocity of y-axis). Modification of the agent position is realized by using the position and the velocity information. The PSO algorithm retains the conceptual simplicity of the genetic algorithm while being much easier to implement and apply to design problems with both discrete and continuous design parameters. Particle swarm adaptation and its modifications have been shown to successfully optimize a wide range of continuous functions (Clerc and Kennedy, 2002; Mendes *et al*., 2004).

Searching procedures by PSO based on the above concept can be described as follows: a flock of agents optimizes a certain objective function. Each agent knows its best value so far (*pbest*) and its *xy* position. The information corresponds to the personal experiences of each agent. Moreover, each agent knows the best value so far in the group (*gbest*) among pbests. The information corresponds to the knowledge of how the other agents around them have performed. Namely, each agent tries to modify its position using the following information:

(1) The distance between the current position and *pbest*, p_t;

(2) The distance between the current position and *gbest*, $\hat{p}_t$.

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_{t+1} = c_0 v_t + c_1 r_1(t) \times (p_t - x_t) + c_2 r_2(t) \times (\hat{p}_t - x_t)\,, \quad (7)$$

where c_0, c_1 and c_2 are positive constant coefficients, r_1 and r_2 are uniformly distributed random numbers in [0,1], v_t is the current velocity of the particle at iteration t, x_t is current position of the particle at iteration t, v_{t+1} is the modified velocity.

The right-hand-side (RHS) of Eq.(7) consists of three terms. The first term is the previous velocity of the agent. The second and third terms are utilized to change the velocity of the agent. Without the second and third terms, the agent will keep on ‘flying’ in the same direction until it hits the boundary. Namely, it tries to explore new areas and, therefore, the first term corresponds to diversification in the searching procedure. On the other hand, without the first term, the velocity of the ‘flying’ agent is only determined by using its current position and its best positions in history. Namely, the agents will try to converge to the their pbests and/or *gbest* and, therefore, the terms correspond to intensification in the searching procedure.

Using the above Eq.(7), a certain velocity that gradually gets closer to pbests and *gbest* can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$x_{t+1} = x_t + v_{t+1}. \quad (8)$$

Pooled-neighbor swarm intelligence approach

In social science context, PSO algorithm combines a social-only component model and a cognition-only model (Clerc and Kennedy, 2002). The real strength of the particle swarm derives from the social interactions among particles as they search the space collaboratively (Clerc and Kennedy, 2002). Recently, Mendes *et al.*(2004) proposed an alternative that is conceptually more concise and potentially performs more effectively than the traditional particle swarm algorithm. In this new version, the particle uses information from all its neighbors, rather than just the best one. In this section, an improved adaptation strategy with enhanced social interactions is proposed for particle swarm optimisation (PSO) algorithm. This adaptation strategy uses more particles' information to control the mutation operation and extends the original formulas of PSO method, which can search the global optimal solution more effectively.

The third term added to the right-hand-side of the velocity Eq.(7) is derived from the successes of others, and is considered a "social influence" term. It was found that when this effect is removed from the algorithm, performance is abysmal (Clerc and Kennedy, 2002). So the social interaction is an important factor to improve the PSO performance. To enhance the social interactions in the algorithm, this paper proposes a new method to improve PSO using some fittest particles' information to modify the particle's position and velocity. Namely, at *i*th iteration, we rearrange the particles in ascending order according to their fitness and select the last *n* particles to modify the particle's position and velocity. Let $\hat{p}_{i,t}$ denote the current position of the particle *i* in these particles at iteration *t*. The updating equations of PNSIA method can be described as follows:

$$v_{t+1}=c_0v_t+c_1r_1(t)\times(p_t-x_t)+\frac{1}{n}\sum_{i=1}^{n}[c_{2,i}r_{2,i}(t)\times(\hat{p}_{i,t}-x_t)], \tag{9}$$

$$x_{t+1}=x_t+v_{t+1}. \tag{10}$$

Each PNSIA method particle modifies its position and velocity using the best solution the particle achieved and several gbests of neighborhood particles. It is similar to the social society in that a group of leaders could make better decisions. However, in PSO, only one *gbest* of neighborhood particles is employed. This process using some neighborhood particles can be called 'intensifying' and 'enhancing' the social influence. Based on this understanding, we should intensify these particles which could lead individuals to better fitness. As a particle swarm population searches over time, individuals are drawn toward one another's successes, with the usual result being clustering of individuals in optimal regions of the space.

The convergence analysis can be conducted similarly based on the form of the recurrence relation of the particle's position derived (Clerc and Kennedy, 2002). The analysis can assure convergence of the proposed algorithm via appropriately chosen parameters satisfying the convergence conditions. One popular choice of updating parameters is c_0=0.7298, c_1=1.49618 and $\sum_{i=1}^{n}c_{2,i}=1.49618$ (Clerc and Kennedy, 2002).

Mixed-variable handing methods

In its basic form, the proposed PNSIA algorithm can only handle continuous variables. However, tap position of transformer and reactive power source installation are discrete variables or integer variables in optimal reactive power dispatch problem. To handle integer variables, simply truncating the real values to integers to calculate fitness value will not affect the search performance significantly. The truncation is only performed in evaluating the fitness function. That is, the swarm will 'fly' in a continuous search space regardless of the variable type.

For discrete variables of the *i*th particle X_i, the most straightforward way is to use the indices of the set of discrete variables with n_D elements:

$$X_i^\mathrm{D}=[x_{i,1}^\mathrm{D},x_{i,2}^\mathrm{D},\ldots,x_{i,n_\mathrm{D}}^\mathrm{D}]. \tag{11}$$

Let X_i^C denote the continuous variables with n_C elements:

$$X_i^\mathrm{C}=[x_{i,1}^\mathrm{C},x_{i,2}^\mathrm{C},\ldots,x_{i,n_\mathrm{C}}^\mathrm{C}],$$

then particle *i* is denoted by $X_i=[X_i^\mathrm{C},X_i^\mathrm{D}]$. For particle *i*, the index value *j* of the discrete variable $x_{i,j}^\mathrm{D}$ is then optimized instead of the discrete value of the variable directly. In the population, the indices of the discrete variables of the ith particle should be the float point variables before truncation. That is, $j\in[1,n_D+1)$, n_D is the number of discrete variables. Hence, the objective function of the ith particle X_i can be expressed as follows:

$$f(X_i),\quad i=1,2,...,M, \tag{12}$$

where

$$X_i=\begin{cases} x_{i,j}, & x_{i,j}\in X_i^C, j=1,...,n_C, \\ x_{i,int(j)}, & x_{i,int(j)}\in X_i^D, j\in[1,n_D+1), \end{cases} \tag{13}$$

where $X_i^C\in\mathbb{R}^{n_C}$ and $X_i^D\in\mathbb{R}^{n_D}$ denote the feasible subsets of continuous and discrete variables of particle X_i, respectively. $int(x)$ denotes the greatest integer less than the real value x.

Implementation of PNSIA for reactive power optimal dispatch

The general process of PNSIA method is described in the above section. Its application to the optimal reactive power dispatch is described as follows:

Pseudo code for the PNSIA algorithm:
Set k=1;
Randomly initialize positions and velocities of all particles;
WHILE (the termination conditions are not met)
FOR (each particle i in the swarm)
Calculate fitness: Calculate the fitness value $f(X_i^k)$ of the current particle using based on the Newton-Raphson power flow analysis results and Eq.(12);
Update $pbest$: Compare the fitness value of $pbest$ with $f(X_i^k)$. If $f(X_i^k)$ is better than the fitness value of $pbest$, then set $pbest$ to the current position X_i^k;
Update $gbest$: Select n $gbest$ particles in the population according to ascending order of fitness values;
Update velocities: Calculate velocities V_i^k using Eq.(9);
Update positions: Calculate positions X_i^k using Eq.(10);
END FOR
Set k=k+1;
END WHILE

SIMULATION RESULTS

To verify the effectiveness and efficiency of the proposed PNSIA based reactive power optimization approach, the IEEE-30 bus power system and a practical 118-bus area power system are used as the test systems. The PNSIA has been implemented in Matlab 6.5 programming language and numerical tests were carried on a Pentium IV 2.0 G computer.

Some parameters must be assigned before PNSIA is used to solve reactive power optimization dispatch. The population size is set to 50 and the maximal generation is set to 300. To evaluate uncertain value combinations of n of PNSIA method, they have been executed 30 times to solve the above reactive power optimization dispatch problem under various value combinations. The results showed that the best solution can be obtained by PNSIA method when n=4.

1. IEEE 30-bus power system

The IEEE 30-bus system is shown in Fig.1 and the system data and operating conditions are given in (Wu *et al*., 1998). The network consists of 48 branches, 6 generator-buses and 22 load-buses. Four branches, (6,9), (6,10), (4,12) and (27,28), are under load tap setting transformer branches. The possible reactive power source installation buses are 3, 10 and 24. Six buses are selected as PV-buses and V_θ-buses as follows: PV-buses: bus 2, 5, 8, 11,13, V_θ-bus: bus 1. The others are PQ-buses. The variable limits are listed in Table 1. The transformer taps and the reactive power source installation are discrete variables with the changes step of 0.01 p.u.

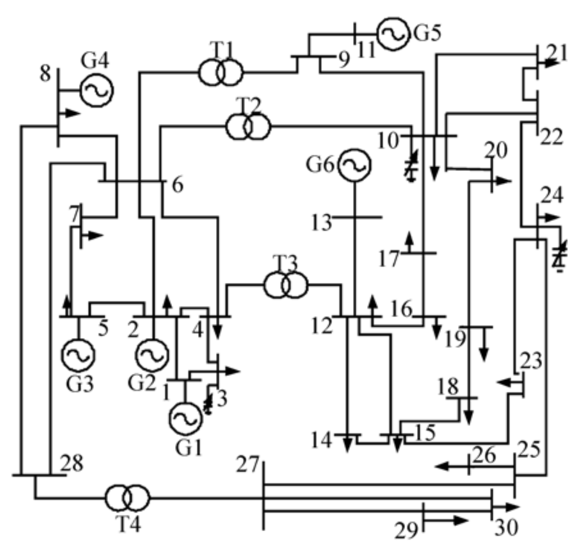

Fig.1 IEEE 30-bus power system

To demonstrate the superiority of the proposed PNSIA approach, simulation results have been compared with various techniques available in literature, namely, standard genetic algorithm (SGA), adaptive GA in (Wu *et al*., 1998), EP method presented in (Lai and Ma, 1997), Broyden's non-linear programming method (Das and Patvardhan, 2002) and PSO method (Clerc and Kennedy, 2002). The initial conditions for all the methods are the same and given as: P_{load}=2.834 p.u., Q_{load}=1.262 p.u.

The initial generator bus voltages and transformer taps are set to 1.0. The total generations and power losses are obtained as follows:

$$\sum P_G = 2.893857 \text{ p.u.}, \sum Q_G = 0.980199 \text{ p.u.},$$
$$P_{\text{loss}} = 0.059879 \text{ p.u.}, \ Q_{\text{loss}} = -0.064327 \text{ p.u.}$$

The voltages outside the limits on three PQ-buses are given as follows:

$$V_{26}=0.932; \ V_{29}=0.940; \ V_{30}=0.928.$$

Table 2 summarizes the results of the optimal settings as obtained by different methods. These results showed that the optimal dispatch solutions determined by the PNSIA lead to lower active power loss than that found by other methods, which confirms that the PNSIA is well capable of determining the global or near-global optimum dispatch solution. Moreover, these results showed that maximum saving is obtained by the PNSIA method. At the same time, the proposed method succeeds in keeping the dependent variables within their limits.

Owing to the randomness in SGA, PSO and PNSIA, the algorithms are executed 30 times when applied to the test system. The best and worst reactive power dispatch solutions together with the associated power loss and the standard deviations found by the three methods are tabulated in Table 3. PNSIA shows good consistency by keeping the difference between the best and worst solutions within 1%. In addition, the average execution times summarized in Table 3 show that PNSIA is faster than SGA and PSO in speed. Table 4 lists the best control variables found by the above three methods in the 30 run times.

2. A practical 118-bus power system

The proposed PNSIA method was applied to a practical 118-bus power system. The power system has 181 transmission elements, 17 generators for AVR control, 9 transformers with 9 to 25 positions and 14 reactive power source installation buses. At initial operating condition, system loss is 141.84 MW and represents about 2.72% of the total real-power

Table 1 Variable limits (p.u.)

Reactive power generation limits			Voltage and tap-setting limits						Var source installments and voltage limits			
Bus	Q_G^{max}	Q_G^{min}	V_G^{max}	V_G^{min}	V_{load}^{max}	V_{load}^{min}	T_K^{max}	T_K^{min}	Q_C^{max}	Q_C^{min}	V_C^{max}	V_C^{min}
1	0.596	−0.298	1.10	0.90	1.05	0.95	1.05	0.95	0.36	−0.12	1.05	0.95
2	0.480	−0.240	–	–	–	–	–	–	–	–	–	–
5	0.600	−0.300	–	–	–	–	–	–	–	–	–	–
8	0.530	−0.265	–	–	–	–	–	–	–	–	–	–
11	0.150	−0.075	–	–	–	–	–	–	–	–	–	–
13	0.155	−0.078	–	–	–	–	–	–	–	–	–	–

Table 2 Comparison of optimal transmission loss for different methods (p.u.)

	$\sum P_G$	$\sum Q_G$	P_{loss}	Q_{loss}	P_{SAVE}	P_{SAVE} (%)
Broyden	2.88986	0.93896	0.055860	−0.32304	0.00402	6.7100
SGA	2.88380	1.02774	0.049800	−0.23426	0.01008	16.8400
AGA	2.88326	0.66049	0.049260	−0.60151	0.01062	17.7400
EP	2.88362	0.87346	0.049630	−0.38527	0.01025	17.1200
PSO	2.88330	0.82500	0.049262	−0.22920	0.01062	17.6200
PNSIA	2.88270	0.64910	0.048711	−0.21650	0.01120	18.6493

Table 3 Comparison of simulation results in the IEEE 30-bus system (p.u.)

Compared item	SGA	PSO	PNSIA
Best P_{loss}	0.049800	0.049716	0.048711
Worst P_{loss}	0.052140	0.050769	0.048734
Average P_{loss}	0.050810	0.049973	0.048721
Standard deviations	0.001785	0.001567	0.000872
Average execution time (s)	156.34	59.21	41.93

Table 4 Control variables after optimization by SGA, PSO and MAPSO (p.u.)

	Bus	SGA	PSO	PNSIA
V_1	1	1.0751	1.0725	1.0784
V_2	2	1.0646	1.0633	1.0690
V_5	5	1.0422	1.0410	1.0472
V_8	8	1.0454	1.0410	1.0470
V_{11}	11	1.0337	1.0648	1.0325
V_{13}	13	1.0548	1.0597	1.0635
T_1	6~9	0.9400	1.0300	1.0200
T_2	6~10	1.0400	0.9500	1.0000
T_3	4~12	1.0400	0.9900	0.9900
T_4	28~27	1.0200	0.9700	0.9700
Q_3	3	0.0000	0.0000	0.0000
Q_{10}	10	0.3700	0.1600	0.3200
Q_{24}	24	0.0600	0.1200	0.1000

generation in the system. There exist 11 deviations at the initial operating point. SGA, PSO and PNSIA are compared in 300 searching iterations. The same parameters for IEEE 30-bus system are utilized in the simulation.

To avoid any hazardous interpretation of optimization results, related to the choice of particular initial particles, we performed the simulation 30 times, starting from different agents randomly generated in the search space. Table 5 giving the best and worst loss values and the computational time shows that the PNSIA method with the greater possibility can generate better solution than SGA and PSO. The average loss value and the standard deviations by the proposed PNSIA method are smaller than the best results by SGA and PSO. The average execution time by PNSIA is about 3 times faster than that by SGA, and the average execution time of PNSIA is 21% less than that of PSO. Considering together more particles' information to control the mutation operation, the proposed method performs better than the PSO model, both in the quality of the solution discovered and in the velocity of convergence, and simulation results showed that PNSIA outperforms SGA and PSO, and is competent for practical reactive power optimization problems.

Table 5 Comparison of simulation results in the practical 118-bus system (p.u.)

Compared item	SGA	PSO	PNSIA
Best P_{loss}	1.332694	1.310471	1.268440
Worst P_{loss}	1.414267	1.348792	1.279830
Average P_{loss}	1.375215	1.321843	1.270150
Standard deviations	0.003546	0.003478	0.001256
Average execution time (s)	335.54	144.46	114.21

CONCLUSION

A pooled-neighbor swarm intelligence approach (PNSIA) has been developed for determination of the global or near-global optimum solution for optimal reactive power dispatch and voltage control of power systems. The pooled-neighbor swarm intelligence approach uses more particles' information to control the mutation operation. The performance of the proposed algorithm demonstrated through its evaluation on the IEEE 30-bus power system and a practical 118-bus power system shows that PNSIA can undertake global search at a fast convergence rate and has the feature of robust computation.

References

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press.

Clerc, M., Kennedy, J., 2002. The Particle Swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evol. Comput.*, **6**(1):58-73. [doi:10.1109/4235.985692]

Das, D.B., Patvardhan, C., 2002. Reactive power dispatch with a hybrid stochastic search technique. *Int. J. Elect. Power Energy Syst.*, **24**(9):731-736. [doi:10.1016/S0142-0615(01)00085-0]

Dommel, H.W., Tinney, W.F., 1968. Optimal power flow solutions. *IEEE Trans. on Power Apparatus and Systems*, **87**:1866-1876.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The ant system: optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, **26**(1):29-41. [doi:10.1109/3477.484436]

Hong, Y.Y., Sun, D.I., Lin, S.Y., Lin, C.J., 1990. Multi-year multi-case optimal AVR planning. *IEEE Trans. on Power Systems*, **5**(4):1294-1301. [doi:10.1109/59.99380]

Iba, K., 1994. Reactive power optimization by genetic algorithm. *IEEE Trans. on Power System*, **9**(2):685-692. [doi:10.1109/59.317674]

Kennedy, J., Eberhart, R.C., 1995. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, p.1942-1948.

Kennedy, J., Eberhart, R.C., 2001. Swarm Intelligence. Morgan Kaufmann Publishers.

Lai, L. L., Ma, J. T., 1997. Application of evolutionary programming to reactive power planning—Comparison with nonlinear programming approach. *IEEE Trans. on Power Systems*, **12**(1):198-204. [doi:10.1109/59.574940]

Lee, K.Y., Bai, X.M., Park, Y.M., 1995. Optimization method for reactive power planning by using a modified genetic algorithm. *IEEE Trans. on Power Systems*, **10**(4):1843-1850. [doi:10.1109/59.476049]

Mendes, R., Kennedy, J., Neves, J., 2004. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. on Evol. Comput.*, **8**(3):20-211.

Wong, K.P., Wong, Y.W., 1994. Genetic and genetic/simulated-annealing approaches to economic dispatch. *IEE Proc. Pt. C*, **141**(5):507-513.

Wu, Q.H., Cao, Y.J., Wen, J.Y., 1998. Optimal reactive power dispatch using an adaptive genetic algorithm. *Int. J. Elect. Power Energy Syst.*, **20**(8):563-569. [doi:10.1016/S0142-0615(98)00016-7]

Yin, X., Germay, N., 1991. Investigations on solving the load flow problem by genetic algorithms. *Electric Power System Research Journal*, **22**(3):151-163. [doi:10.1016/0378-7796(91)90001-4]