IET Generation, Transmission & Distribution

Research Article

Multi-group particle swarm optimisation for transmission expansion planning solution based on LU decomposition

ISSN 1751-8687
Received on 14th June 2016
Revised 29th November 2016
Accepted on 15th January 2017
E-First on 6th March 2017
doi: 10.1049/iet-gtd.2016.0923
www.ietdl.org

Shengjun Huang[1,2] ✉, *Venkata Dinavahi*[1]
[1]*Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada*
[2]*College of Information System and Management, National University of Defense Technology, Changsha, People's Republic of China*
✉ *E-mail: shengjun@ualberta.ca*

Abstract: As power systems are being highly stressed with the boost of loading levels and the introduction of new generation sources, transmission expansion planning (TEP) has regained its significance as a pivotal problem to be solved. To ameliorate the performance on both efficiency and accuracy for the solution of TEP from the aspect of algorithm design, a static DC TEP without generation redispatch is investigated by the proposed multi-group particle swarm optimisation (MGPSO) algorithm. MGPSO is based on the discrete PSO framework with several beneficial enhancements involved, such as Sobol sequence initialisation method, multi-group co-evolution strategy, and mutation mechanism. For the solution of linear programming subproblem within the framework of MGPSO, a linear equation system is extracted and then addressed with efficient LU decomposition approach. Case studies have been implemented on five classical benchmarks, ranging from 6-bus to 118-bus, between the MGPSO and commercial software Lingo 11.0 to validate the superiority of MGPSO. Speedup analysis as well as performance evaluation of different acceleration strategy involved in MGPSO are implemented and discussed.

1 Introduction

Continued increase in loading levels and an emphasis on accessing far-flung renewable and distributed generation have underscored the necessity of building new transmission lines worldwide, which demands intensive research for an efficient solution of transmission expansion planning (TEP) problem [1, 2]. Originally proposed in 1970 [3], the TEP is a decision-making problem that tries to design the expansion plan at the lowest cost while meeting constraints for safe and efficient operation. During the last decades, several TEP problems have emerged along with the evolution of power system, which can be categorised according to different aspects:

- Transmission network model: Four types of models representing the transmission network have been investigated for TEP study: (i) DC power flow model, which is the most widely accepted model with a characteristic of mixed-integer, non-linear, non-convex, and NP-hard [4]. (ii) Transportation model is a mixed-integer linear programming (MILP) problem, which can be generated from DC power flow model by eliminating the constraints of Kirchhoff's voltage law. It is no longer popular since its inaccuracy. (iii) Disjunctive model is also derived from DC power flow model by replacing integer decision variables with binary decision variables and introducing large constant to linearise the power flow constraint, resulting in a MILP problem with the whole characteristic of DC power flow model kept. This model introduces large numbers of decision variables and constraints during the process of linearisation. (iv) AC power flow model, which is the most accurate one since both active and reactive power flow are considered, but it is usually considered only at a later stage of the planning process when the most attractive topologies have been determined [5].
- Planning horizon: According to the time span considered, TEP can be classified into static and dynamic TEP, where the former considers only one stage, while the later divides the whole period into several stages, leading to more accurate decision making as well as heavily computational burden.
- Generation redispatch: Generation redispatch is required for industry practice, especially when intermittent renewable energy generators are utilised. However, it is reported in [6] that TEP with generation redispatch is easier than the case of planning without redispatch in computational complicity.
- Security criterion: Modelling security drastically increases the complexity of the resulting problem since the unavailability of system components needs to be characterised. One of the most extensively adopted criteria in literature on TEP is the $n-1$ security criteria.
- Uncertainty: The uncertainty usually comes from the fluctuated renewable energy generation and load demand [7].

From the perspective of realistic operation, AC power flow model, multi-stage planning, generation redispatch, security criterion, and uncertainty should all be considered as a whole problem, which makes the problem so complicated that the computation is intensive or even intractable. According to Silva *et al*. [8], the methodology suitable for the basic TEP can be extended to some more complicated TEP problems. Thus, the static DC TEP (SDCTEP) without generation redispatch, security criterion, and uncertainty is investigated to provide inspiration for the addressing of the whole problem in the future.

All the efforts devoted into solving the TEP problem can be classified into three categories (for more comprehensive reviews, the interested reader is referred to [9, 10]): (i) deterministic methods [1, 2], (ii) heuristic methods [11], and (iii) meta-heuristic methods [12–21]. To transform power system equations into mathematical optimisation models, the original system should be largely simplified with some aspects discarded, which is difficult due to the requirement of rigorous analytical background for both mathematical and power systems. On the other hand, meta-heuristics methods [22] are relatively straightforward. Furthermore, it is also possible to find the sub-optimal or global optimal solutions for most problems regardless of whether they are non-linear, non-convex, combinatorial, or even NP-hard due to their global convergence capability [23].

However, the execution time still remains a bottleneck for meta-heuristic algorithms when dealing with large-scale TEP problems due to the long-time consumed by thousands of times of linear programming (LP) solution (which has been reported by Hashimoto *et al*. [24] that it may take up to 90% of the total elapsed time). Therefore, for the purpose of improving the performance, two alternatives have been emphasised in the literature: (i) an efficient LP solver [25], and (ii) a better meta-heuristic algorithm [11]. To fill this gap, an efficient LP solution process was proposed by Hashimoto *et al*. [24], where the LP subproblem was transformed into another equivalent LP problem with reduced numbers of constraints and variables, resulting in less solution time. On the other hand, as a popular meta-heuristic method, particle swarm optimisation (PSO) gained widespread application in the solution of TEP.

Many successful results have been reported in the solution of TEP with classical PSO and its variants [12–19], of which the most popular enhancement strategies include high diversity initialisation [15, 16], replication [17], mutation [16, 17, 19], selection [17], and evolutionary adaptation strategy [18]. In addition, four types of PSO are discussed in [20] as a survey, and a multi-objective PSO has also been proposed in [21].

The population of the majority of the above PSO variants are led by a single global best, which usually results into premature and local optimal solution; therefore, their applications are mostly limited to small- or medium-scale systems. In this paper, a multi-group co-evolution strategy is employed on the classical PSO to increase the global search capability, resulting in a multi-group PSO (MGPSO), which is then enhanced by several premature elimination strategies, such as Sobol initialisation and mutation mechanism. Furthermore, to improve the efficiency of thousands of times of LP solution, a linear equation system (LES) is generated for each LP, which is then tackled by the efficient LU decomposition approach.

Five case studies are employed to verify the proposal ranging from small-scale to large-scale: the Garver 6-bus system, the IEEE 24-bus system, the Brazilian 46-bus and 79-bus systems, and the IEEE 118-bus system. The results indicate that MGPSO performs better than other meta-heuristic algorithms, and achieves considerable speedup than commercial software Lingo 11.0 in execution time.

The rest of the paper is organised as follows. The TEP problem formulation as well as LP subproblem are presented in Section 2. Section 3 highlights the MGPSO and its implementation for TEP. Case studies and discussion are reported in Section 4. Finally, conclusions are drawn in Section 5.

2 TEP problem formulation

2.1 DC power flow model

DC power flow model is the basic network model for TEP study, which can be formulated as follows:

$$\min \sum_{(i,j)\in\Omega} c_{ij} n_{ij} + \alpha \sum_{k\in\Gamma} r_k, \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{S f} + \boldsymbol{g} + \boldsymbol{r} = \boldsymbol{d}, \tag{2}$$

$$f_{ij} - \gamma_{ij}\left(n_{ij}^0 + n_{ij}\right)\left(\theta_i - \theta_j\right) = 0, \tag{3}$$

$$|f_{ij}| \le \left(n_{ij}^0 + n_{ij}\right)\bar{f}_{ij}, \tag{4}$$

$$\boldsymbol{0} \le \boldsymbol{g} \le \bar{\boldsymbol{g}}, \tag{5}$$

$$\boldsymbol{0} \le \boldsymbol{r} \le \boldsymbol{d}, \tag{6}$$

$$0 \le n_{ij} \le \bar{n}_{ij},\ n_{ij}\ \text{integer},\ \theta_i \text{ and } \theta_j \text{ unbounded}. \tag{7}$$

where the objective function (1) comprises of the construction cost $\sum_{(i,j)\in\Omega} c_{ij} n_{ij}$ and the load curtailment penalty $\alpha \sum_{k\in\Gamma} r_k$, with c_{ij} and n_{ij} represent the cost and the number of parallel candidate circuit to be built for corridor *i–j*, α is the penalty factor, r_k is the loss of load for bus k, Ω and Γ are sets of candidate circuits and buses, respectively. Equation (2) is the bus balance constraint, where $\boldsymbol{S}$ is the bus-branch incidence matrix defined by (8), $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{r}$, and $\boldsymbol{d}$ denote vectors of power flow of corridor, generation, load curtailment, and load, respectively. Equations (3) and (4) are DC power flow constraints, where γ_{ij}, n_{ij}^0 and $\bar{f}_{ij}$ are the susceptance, initial number, and maximum power flow of circuit *i–j*, θ_i and θ_j are voltage phase angles. Equations (5)–(7) are limit constraints for the generation, load curtailment, and the number of new lines to be built.

$$\boldsymbol{S}_{ij} = \begin{cases} -1, & \text{if branch } i-j \text{ originates from bus } i, \\ 1, & \text{if branch } i-j \text{ terminates at bus } j, \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

2.2 Linear programming subproblem

The SDCTEP (1) is a mixed-integer non-linear programming (MINLP) problem, which has been proved to be a typical hard combinatorial problem [24, 25]. However, if a candidate solution $\boldsymbol{n}$ is given, i.e. n_{ij} is fixed, problem (1) can be converted to a LP problem and rewritten as follows:

$$\min \sum_{(i,j)\in\Omega} c_{ij} n_{ij} + \alpha \sum_{k\in\Gamma} r_k, \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{B\theta} + \boldsymbol{g} + \boldsymbol{r} = \boldsymbol{d}, \tag{10}$$

$$\gamma_{ij}\left(n_{ij}^0 + n_{ij}\right)\left|\theta_i - \theta_j\right| \le \left(n_{ij}^0 + n_{ij}\right)\bar{f}_{ij}, \tag{11}$$

$$\text{Constraints (5)} - \text{(7)}. \tag{12}$$

where constraints (10) and (11) are derived from (2) and (4), respectively, with f_{ij} replaced by $\gamma_{ij}\left(n_{ij}^0 + n_{ij}\right)\left(\theta_i - \theta_j\right)$ according to (3). Accordingly, $\boldsymbol{Sf}$ is rewritten into $\boldsymbol{B\theta}$, with $\boldsymbol{\theta}$ represents the vector of voltage phase angles, and $\boldsymbol{B}$ is a symmetric singular matrix generated by

$$\boldsymbol{B} = -\boldsymbol{S}\left(\left(\left(\gamma .* \left(\boldsymbol{n}^0 + \boldsymbol{n}\right)\right)1_{1\times n_b}\right) .* \boldsymbol{S}^{\mathrm{T}}\right), \tag{13}$$

where $\mathbf{1}_{1\times n_b}$ is a vector of ones by the scale of $1 \times n_b$ and '$.*$' means element-wise multiplication operator in Matlab, $\boldsymbol{\gamma}$, $\boldsymbol{n}^0$, and $\boldsymbol{n}$ are vectors of γ_{ij}, n_{ij}^0, and n_{ij}, respectively, n_b is the total number of buses.

Since MINLP (1)–(7) contains large numbers of constraints, which provides great obstacles for meta-heuristic algorithm with conventional constraint handling methods, such as penalty function method, therefore, another more efficient strategy [shown in Algorithm 1 (see Fig. 1)] has been proposed and employed by the majority of meta-heuristic algorithms when tackling TEP problems.

2.3 Linear programming transformation

To be solved more efficiently, the LP (9)–(12) was transformed into a new form by Hashimoto *et al*. [24] via a series of processes, resulting in the number of decision variables reduced from $2n_b$ to n_b, and the number of constraints decreased from $2n_b + 2n_c$ to $n_b + 2n_c + 1$, where n_c is the number of candidate circuits. Due to the reduction in problem scale, the revised LP formulation gained a better performance than Minos 5.4 software in [24]. However, it was pointed by the authors themselves that the proposed transformation presented a disadvantage of requiring the inversion of matrix $\boldsymbol{B}$ in an explicit way, which is relatively computationally intensive.

Instead of transforming the original LP into another LP with reduced scales, a LES is extracted in this paper since it is much easier to be addressed. LES can be solved by matrix inverse or LU decomposition in only one time while large numbers of iterations are required for LP solution in either simplex method or interior point method. To fulfil the transformation from LP to LES, the following two assumptions are proposed:

Algorithm 1

Propose a candidate solution vector $\boldsymbol{n}$.
while Terminate conditions are not met **do**
  Evaluate the quality of $\boldsymbol{n}$ by LP (9) – (12).
  Evolve to a new solution $\boldsymbol{n}$ by several operation processes, such as crossover, mutation, velocity adjusting, etc.
end while
return The final solution.

Fig. 1 ***Algorithm 1*** *Pseudo code of meta-heuristic algorithms for TEP solution*

- Assumption 1: $r_k = 0$ for all k. It is required that all load buses are satisfied by generation buses for the optimal solution, i.e. for each bus the loss of load is 0, which means $r_k = 0$.
- Assumption 2: $\boldsymbol{g} = \bar{\boldsymbol{g}}$. This assumption holds for all the TEP without the consideration of generation redispatch, such as the real Brazilian 46-bus system [5, 6, 8, 15].

On the basis of Assumptions 1 and 2, LP problem (9)–(12) can be modified into:

$$\min \sum_{(i,j)\in\Omega} c_{ij} n_{ij} + \alpha \sum_{(i,j)\in\Omega} s_{ij}, \tag{14}$$

$$\text{s.t.} \quad \boldsymbol{B\theta} + \bar{\boldsymbol{g}} = \boldsymbol{d}, \tag{15}$$

$$\gamma_{ij}\left(n_{ij}^0 + n_{ij}\right)\left|\theta_i - \theta_j\right| \le \left(n_{ij}^0 + n_{ij}\right)\bar{f}_{ij} + s_{ij}, \tag{16}$$

$$s_{ij} \ge 0. \tag{17}$$

where s_{ij} is the relaxed coefficient. Compared with (9)–(12), decision variables $\boldsymbol{g}$ and $\boldsymbol{r}$ are eliminated, as well as the constraints of limits for them.

The key point for solving LP (14)–(17) is deriving $\boldsymbol{\theta}$ from LES (15), after which, the other decision variable s_{ij} is very easy to be extracted from (16). There are several methods to tackle with LES, such as LU decomposition. The details to solve (14)–(17) will be explicated by Algorithm 2 in Section 3.6.

3 Multi-group particle swarm optimisation

PSO was first proposed by Kennedy and Eberhart [26] in 1995 with the core idea of sharing information between particles, local best, and global best, which can be expressed as follows:

$$v_i^{k+1} = w_0 v_i^k + c_1 w_1 (p_i^k - x_i^k) + c_2 w_2 (g^k - x_i^k), \tag{18}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{19}$$

where v_i^{k+1} and v_i^k are the velocities of particle i at the $(k+1)$th and (k)th iteration; x_i^{k+1} and x_i^k are the positions of particle i at the $(k+1)$th and (k)th iteration; $w_0 \in [0, 1]$ is the inertia weight; $c_1, c_2 \in [0, 2]$ are the self-knowledge and social-learning factors; $w_1, w_2 \in [0, 1]$ are random numbers; p_i^k is the local best of particle i till (k)th iteration; g^k is the global best till (k)th iteration.

On the basis of the classical PSO, a MGPSO has been proposed for the solution of SDCTEP problem, with the most important features are detailed as follows.

3.1 Problem codification

The design of any iterative meta-heuristic algorithm requires an encoding of the solution, which plays a crucial role in the efficiency and effectiveness [27]. Several alternative codification approaches were proposed in the literature, including binary codification, independent bits, and decimal codification, etc. In this paper, the decimal codification is employed based on the following two considerations:

- Both binary and independent bits have Hamming cliffs, introducing great obstacles for convergence. For example, two successive numbers 3 and 4 in decimal will be expressed totally inverse by binary bits as '011' and '100', respectively.
- The individual length is longer if the solution is represented by binary bits and the transform process from binary to decimal will be executed for thousands of times, which requires more computational resources.

3.2 Population initialisation

To gain higher diversity, Sobol sequence [28] is adopted in this paper. Different from the classical pseudo-random sequences, the Sobol sequence is a quasi-random sequence. It can be observed from Fig. 2 that the Sobol sequence is more uniformly sampled than pseudo-random sequence, where a total number of $N = 500$ points is sampled for each method in a 1×1 square area. To do a numerical analysis, both axes are divided into $W = 1, \ldots, 10$ segments, resulting in W^2 square subareas. If the points are ideally distributed in even, each subarea should have N/W^2 points. Suppose subarea ij has y_{ij} points $(i = 1, \ldots, W,\ j = 1, \ldots, W)$, then the abstract biases for each subarea from its ideal value can be derived, and their average value E can be used to illustrate the uniformity of each sequence.

$$E = \frac{\sum_{i=1}^{W}\sum_{j=1}^{W} |y_{ij} - N/W^2|}{W^2}. \tag{20}$$

Fig. 3 shows the average bias E across different W and N. It can be seen that for the fixed N, E is getting smaller as W increases, which is due to the reduction of the ideal value N/W^2. Another important feature is that the average bias of Sobol sequence is always smaller than pseudo sequence for any fixed N and W, indicating that the Sobol sequence is evenly distributed, i.e. diversity is guaranteed.

3.3 Particle evolution

Particle evolution is the kernel of all kinds of PSO ranging from single-group to multi-group. For the discrete version of PSO, the most commonly utilised method is regulating velocity and position to be integer using functions such as *fix*(), *round*(), and *ceil*(). As the convergence continues, the current solution x_i^k is getting close to its local best p_i^k and global best g^k, thus the real value of velocity is usually within an interval of (−1, 1) according to (18). At this circumstance, *fix*() and *ceil*() may have huge error, such as *fix*(0.9) = 0 and *ceil*(0.1) = 1, therefore *round*() is utilised by Murugan [15] with (18) is modified as

$$v_i^{k+1} = \mathrm{round}\left(w_0 W_0 v_i^k + c_1 W_1 (p_i^k - x_i^k) + c_2 W_2 (g^k - x_i^k)\right), \tag{21}$$

where W_0 is a random number taking discrete values of 0, 1, or −1; W_1 and W_2 are random discrete numbers of either 0 or 1; all the other variables and parameters share the same definition as (18) and (19). Different from (18), (21) provides more possibilities by

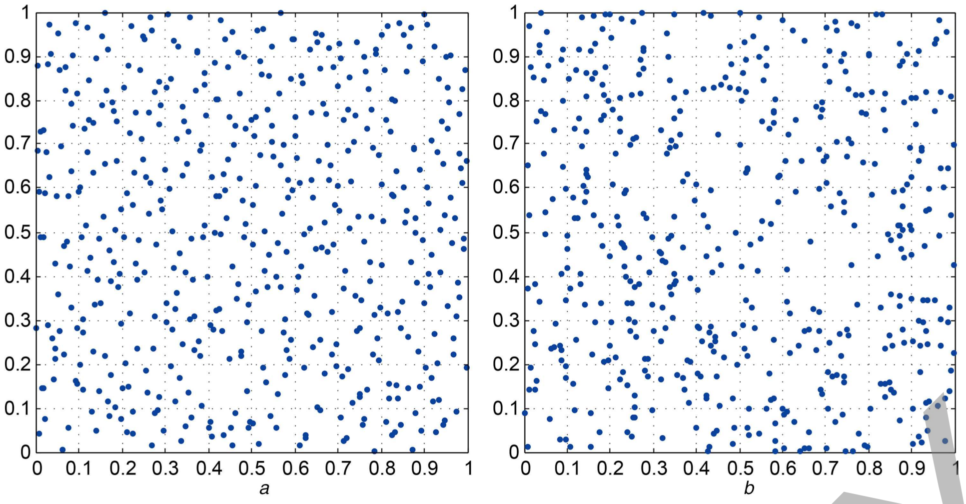

Fig. 2 *Sample points of different random sequences*
(a) Sobol sequence, ***(b)*** pseudo-random sequence

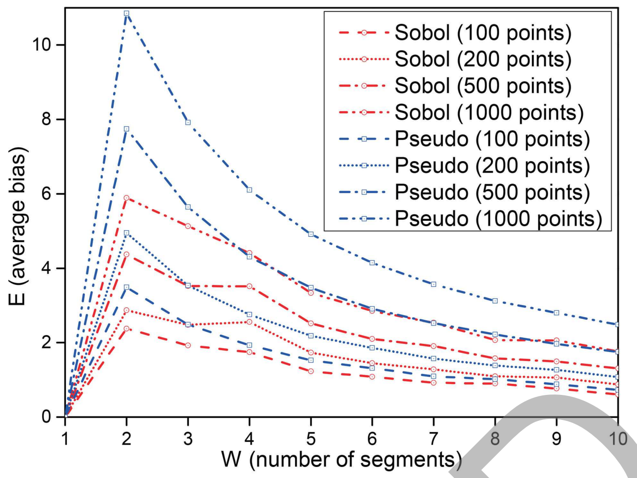

Fig. 3 *The relationship between E and W for different N*

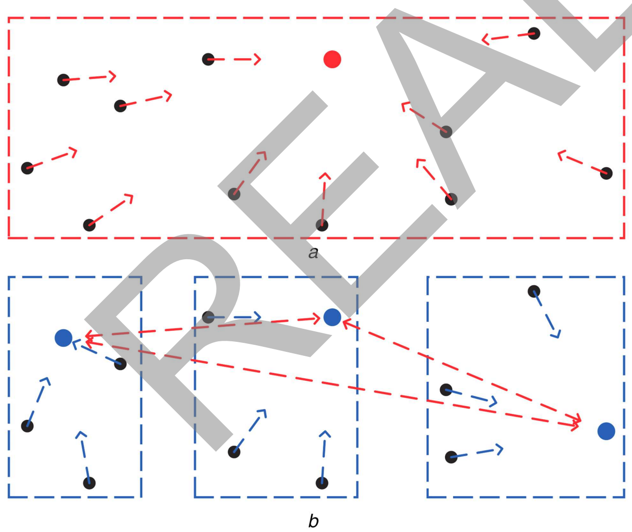

Fig. 4 *Illustration of PSO evolution strategies*
(a) Single-group evolution, ***(b)*** Multi-group co-evolution

the introduction of W_0, W_1, and W_2, such as temperately eliminating the influence of velocity, local best, and global best by setting W_0, W_1, and W_2 into 0 respectively. The search direction can even be turned totally inverse if $W_0 = -1$. These possibilities bring more diversity to the searching process. The performance of this strategy has been verified by Murugan [15], and it will be employed as the evolution strategy for each particle of MGPSO in this paper. If the velocity is too high, particles might fly past good solutions; if it is too small, particles may not explore sufficiently beyond locally good regions [14]. Therefore, the velocity is bounded into [−2, 2] in this paper.

3.4 Multi-group co-evolution

Although particle evolution strategy shown above has a good performance reported in [15], beneficial improvements are still possible to be investigated. For example, the current global best g^k involves in the evolution process of all particles in (21), which forces the whole population to converge to a small space dominated by g^k, leaving large areas unexplored, where the real global optimal might exist, i.e. the final solution is a local optimal. Therefore, to reduce the strong influence of single global best, the whole population was divided into n_g groups in this paper to keep diversity, and each group ($j = 1 \cdots n_g$) has a global best g_j^k until iteration k. The multi-group co-evolution strategy is guided by the following two rules:

- All the global best of different groups are different. If $g_{j_1}^k = g_{j_2}^k$ while $j_1 \neq j_2$, then a mutation process should be triggered on either $g_{j_1}^k$ or $g_{j_2}^k$. This rule makes each group driven by different 'leader', which forces the whole population to explore more space.
- The global bests of different groups should share information with each other. Two-point crossover was employed to perform the information exchange on randomly selected $g_{j_1}^k$ and $g_{j_2}^k$, with two obvious benefits: (i) improving the gene characteristics of poor fitness individuals, and (ii) introducing diversity on good fitness candidates.

Fig. 4 shows the difference between single-group evolution and multi-group co-evolution, where particles are represented by solid circles and the influences are illustrated by dashed arrows. In Fig. 4*a*, all the particles are influenced by the population global best, while in Fig. 4*b* the particles are driven by each group's global best, and each global best is influenced by other global bests, which maintains a good trade-off between population diversity and global convergence.

3.5 Mutation mechanism

To prevent premature convergence and enable the MGPSO algorithm escape from the local optimal, the mutation process has been carried out in three steps:

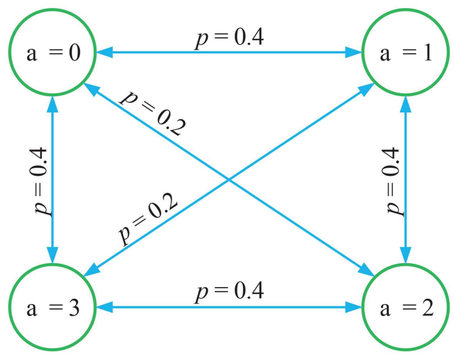

Fig. 5 *Transform probability for point mutation*

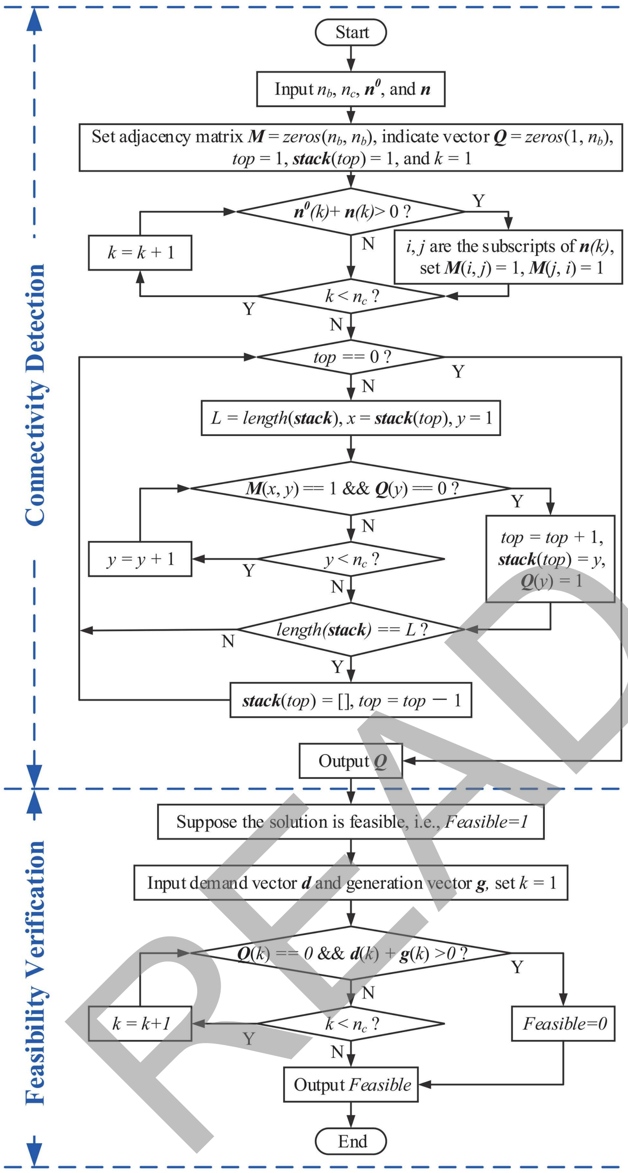

Fig. 6 *Flowchart for network analysis*

- Determine the candidate mutation individual according to a mutation rate r_m. Instead of doing the mutation directly on points of each individual with the same probability, this step could enable large numbers of particles undisturbed, leaving the feasibility maintained.
- Replace the candidate mutation individual by its local best. This process gives a good starting point for mutation, i.e. the mutation always finds the points near the local best, representing a higher probability to get good results.
- Perform the mutation process on the points determined by a selection probability of r_p. The point mutation process is carried out according to the transformation probability shown in Fig. 5. For example, the probability for $a = 0$ to mutate into $a = 1$, $a = 2$, and $a = 3$ is 0.4, 0.2, and 0.4, respectively.

3.6 Fitness evaluation

Fitness evaluation and constraint handling consume the largest part of the time for almost all meta-heuristic algorithms. According to the power balance constraints, if the candidate solution $\boldsymbol{n}$ is infeasible, there must be some loss of load for buses, which then will be multiplied by α for penalty. To save computational effort, the sum of power imbalance is directly valued as P for all infeasible candidate without any solution of LP. On the other hand, the LES (15) should be figured out to get the objective function value for feasible solutions.

For the purpose of checking the feasibility of each candidate $\boldsymbol{n}$, a process of network analysis consisted of connectivity detection and feasibility verification has been proposed, which is presented in Fig. 6. The connectivity detection is implemented for each bus to check whether it is connected to the grid or not, where the depth-first search algorithm [29] from graph theory is adopted. The result is represented by a binary vector $\boldsymbol{Q}$ consists of n_b elements, if $\boldsymbol{Q}(i) = 1$, then bus i is connected, otherwise, it is disconnected. Based on $\boldsymbol{Q}$, the feasibility verification is performed. If the set of disconnected buses contains generation or load buses, then the power balance will be destroyed, i.e. the solution is infeasible; otherwise, candidate $\boldsymbol{n}$ is feasible.

As discussed above, the objective value of infeasible solution will be directly valued as P, while for the feasible candidate, LES (15) should be solved. It is required that the matrix $\boldsymbol{B}$ in (15) should be non-singular for the utilisation of LU decomposition. However, the original $\boldsymbol{B}$ is singular, thus two steps are implemented: (i) delete column i and row i of $\boldsymbol{B}$ if $\boldsymbol{Q}(i) = 0$, resulting in $\boldsymbol{B}_{\text{connected}}$; (ii) select a slack bus j and eliminate the corresponding row and column to get the final non-singular matrix $\boldsymbol{B}_{\text{nonsingular}}$. Similarly, the singular LES (15) can be transformed into a non-singular LES:

$$\boldsymbol{B}_{\text{non-singular}}\boldsymbol{\theta}_{\text{non-singular}} + \bar{\boldsymbol{g}}_{\text{non-singular}} = \boldsymbol{d}_{\text{non-singular}}, \tag{22}$$

where $\boldsymbol{d}_{\text{non-singular}}$ and $\bar{\boldsymbol{g}}_{\text{non-singular}}$ are demand and generation vectors with slack bus and those with $\boldsymbol{Q}(i) = 0$ eliminated. The process, based on LU decomposition since it is more efficient than matrix inverse method, to derive the objective function value of feasible candidate is illustrated in Algorithm 2 (see Fig. 7).

$$[\boldsymbol{L}, \boldsymbol{U}] = lu(\boldsymbol{B}_{\text{non-singular}}), \tag{23}$$

$$\boldsymbol{\theta}_{\text{non-singular}} = \boldsymbol{U} \backslash (\boldsymbol{L} \backslash (\boldsymbol{d}_{\text{non-singular}} - \bar{\boldsymbol{g}}_{\text{non-singular}})). \tag{24}$$

3.7 Terminate condition

The algorithm terminates if the incumbent (the best solution found so far) does not improve after a specified number n_t of iterations or the maximum numbers of iteration G is reached.

3.8 Implementation framework

The overall implementation framework of MGPSO is illustrated by algorithm 3 (see Fig. 8) with pseudo code, where the key functions in line 6–9 have been explicated in the above subsections.

4 Case studies and discussion

Two types of tools are employed to perform the case studies, Matlab 2015b and Lingo 11.0, which are all run on a Windows desktop with an Intel Xeon E5-2620 CPU at 2.10 GHz with 32 GB

Algorithm 2

Get the lower triangle matrix $\boldsymbol{L}$ and the upper triangle matrix $\boldsymbol{U}$ of $\boldsymbol{B}_{nonsingular}$ by (23), where $lu()$ is the LU decomposition function defined in Matlab.
Derive the $\boldsymbol{\theta}_{nonsingular}$ by (24), where "\" is the left division operator in Matlab.
Set the value of θ_i for the eliminated buses (slack and those with $\boldsymbol{Q}(i) = 0$) as 0.
Generate the slack variable values s_{ij} by (16).
Return the objective function value of (14).

Fig. 7 ***Algorithm 2*** *LU decomposition method for LP (14)*

Algorithm 3

Input original data set Φ, where the number of candidate circuits n_c is included;
Initialize parameters: population size m, number of group n_g, maximum generation G, power imbalance penalty value P, penalty factor α, and terminate criterion n_t;
Initialize variable matrices and vectors: population $X = \textit{Sobol}(m, n_c)$, velocity $V = -1 + 2 \times \textit{rand}(m, n_c)$, global best solution and its fitness value for each group $S_g = \textit{ones}(n_g, n_c)$ and $F_g = \textit{inf}(n_g, 1)$, local best solution and its fitness value for each individual $S_l = \textit{ones}(m, n_c)$ and $F_l = \textit{inf}(m, 1)$; Section 3.2
Set unimproved counter $c = 0$ and the global best of the whole population $f_b = \infty$;
for $i = 1$ to G **do** Section 3.7
  $F = \textit{Fitness Evaluation}(X, \Phi, \alpha, P)$; Section 3.6
  $[F_g, F_l, S_g, S_l] = \textit{Group-based Coevolution}(F, n_g, F_g, F_l, S_g, S_l, \Phi, \alpha, P)$; Section 3.4
  $[X, V, S_l] = \textit{Particle Mutation}(X, V, S_l)$; Section 3.5
  $[X, V] = \textit{Particle Evolution}(X, V, S_g, S_l)$; Section 3.3
  if $\min(F_g) < f_b$ **then**
    Let $f_b = \min(F_g)$ and $c = 0$;
  else
    $c = c + 1$;
  end if
  if $c >= n_t$ **then** Section 3.7
    break;
  end if
end for
return global best f_b and the total number of iteration i.

Fig. 8 ***Algorithm 3*** *Pseudo Code of MGPSO*

Table 1 Scale and complexity of test cases considered

Systems	$\bar{n}_{ij}$	n_b	n_c	Search space size		
6-bus	4	6	15	5^{15}	$\simeq$	3.05×10^{10}
24-bus	3	24	41	4^{41}	$\simeq$	4.84×10^{24}
46-bus	3	46	79	4^{79}	$\simeq$	3.65×10^{47}
79-bus	3	79	143	4^{143}	$\simeq$	1.24×10^{86}
118-bus	2	118	186	3^{186}	$\simeq$	5.55×10^{88}

RAM. In addition, results from the literature are also included for comparison and discussion.

4.1 Case studies

Five systems without generation redispatch are considered in this study: the Garver 6-bus system [3, 5], the IEEE 24-bus system with the third future generation plan [30], the South Brazilian 46-bus system [6], the Southeast Brazilian 79-bus system [31, 32], and the IEEE 118-bus system [33]. The former four are the same with the systems studied in [4], while the last one is modified from [33] by reducing the maximum capacity of each line to 40% of its original value due to over sufficient condition. All of these are classical benchmark systems and have been investigated by several researchers, an overview of whose scale and complexity is shown in Table 1.

4.2 Parameter settings

Control parameters are very significant for algorithm performance in relation to solution quality as well as convergence speed, which are usually problem dependent. In this work, all control parameters related to MGPSO are manually tuned based on a few preliminary experiments, which are given in Table 2. It should be noted that these parameters may not be optimal since the comprehensive test is not fully implemented. Apart from the above parameters related with cases, c_1, c_2, and w_0 are robust for all four cases, with $c_1 = 2.0$, $c_2 = 2.0$, and w_0 is expressed as follows:

$$w_0 = w_{\max} - (w_{\max} - w_{\min}) \times i/G, \tag{25}$$

where $w_{\max} = 0.6$ and $w_{\min} = 0.2$, i is the current iteration count.

4.3 Results

It will be instructive to review the results reported in the literature before introducing the results obtained by MGPSO. In 2014, a work [4] was done to analyse the complexity of TEP with the branch-and-reduce optimisation navigator (BARON). The result is given in Table 3, where ϵ_r is the relative termination tolerance. It was concluded that BARON converged very fast for the 6-bus and 24-bus systems, however, the execution time increased sharply for the 46-bus system, and it even could not get the historical best solution after a computation time of 8 h for the 79-bus system.

Table 4 illustrates the main result of commercial software Lingo 11.0 and MGPSO programmed with Matlab 2015b running on the same desktop; result reported by modified PSO (MPSO) [15] is also included for referencing rather than comparison since the simulation platform was different.

Different versions of MPSO were reported in [15], the fastest one with convergence rate over 50 and 100% are chosen for referencing. For the 6-bus and 24-bus systems, MPSO is slower

Table 2 Control parameters of MGPSO for different cases

Systems	m	n_g	α	P	G	n_t
6-bus	20	4	70	400	200	30
24-bus	200	20	30	400	400	80
46-bus	800	40	400	800	1000	100
79-bus	2000	80	600	2000	2000	300
118-bus	2000	80	600	2000	2000	300

than BARON, but it is a little faster for the 46-bus system. The convergence rate of these systems can reach 100% for 10 times of trial. No result has been reported by MPSO for the large-scale 79-bus system.

The default setting of non-linear programming solver Lingo is adopted in this paper due to its fast solution process. The precision of elapsed time for Lingo is 1 s. A total number of 30 times has been tested for each instance. It can be concluded that Lingo performs better than MPSO on the execution time; however, the convergence rate of Lingo is relatively low for medium- and large-scale systems, the reason is that the option 'global solver' is disabled in the default setting. Although the Lingo is very fast with the default setting, the global optimal solution cannot be guaranteed, which is due to the non-convex and multi-modal feature of TEP. On the other hand, the global optimal can be always reached if 'global solver' is enabled, at the expense of much longer solution time as its low efficiency. For example, as shown in Table 4, the average solution time of the 24-bus system with Lingo on default setting is only 26.900 s, but it turns to be 09:24:11 (33,851 s, 1258 times longer) after calling the 'global solver'. The same solver has also been implemented on 46-bus and 79-bus systems, but the solution process cannot terminate after a running of 2 days.

A detailed comparison is conducted between MGPSO and Lingo. For the 6-bus system, both methods converge very fast and accurate, but when it comes to the 24-bus system, the global best of 218 million US$ (MUS) is not reachable for some trials of Lingo, even the running time is longer than MGPSO, for which the convergence rate is 100%. MPSO, Lingo, and MGPSO get the same global optimal with [34, 35] for the 46-bus system of 154.42 MUS with a probability of 100, 63.33, and 100%, respectively. For the 79- and 118-bus system, no common acceptable global optimal has been reported, thus the best result gained by MGPSO of 457.8 and 929.4 MUS are regarded as the criterion for convergence judgment. As shown in Table 4, Lingo performs better or no worse than MGPSO for 15 and 13 times; however, the average cost is slightly higher and running time is more than 12 times longer.

Table 3 Run time of different simulations (s) [4]

Systems	6-bus	24-bus	46-bus	79-bus
BARON $\epsilon_r = 0.1$	1	2	972	28,800[a]
BARON $\epsilon_r = 0.01$	1	4	5347	28,800[a]
BARON $\epsilon_r = 0.001$	1	4	6418	28,800[a]

[a] Best solution has not been found by 28,800 s.

4.4 Discussion

4.4.1 Speedup analysis: To do the numerical speedup analysis, the average execution time for Lingo to solve the 6-bus system in Table 4 is approximately assumed to be 0.3 s, therefore the speedup for MGPSO over Lingo is ×1.7. For the other systems, a speedup of ×3.9, ×6.8, ×12.5, and ×12.8 is also achieved, respectively, which is illustrated in Fig. 9. The dashed line from 6-bus to 79-bus system indicates that the slope goes higher as system scale increases, i.e. the speedup is higher for larger systems. The suddenly flattened trend from 79-bus to 118-bus system is due to their similar search space size shown in Table 1. Actually, the speedup has a linear relationship (shown in Fig. 10) with the search space size rather than the number of buses.

4.4.2 Performance evaluation of multi-group co-evolution: To identify the performance improvement brought by multi-group co-evolution, both single- and multi-group PSO have been implemented. The 46-bus system was determined as the target playground due to its proper difficulty. Fig. 11 depicts the behaviour of convergence for both algorithms. It can be noticed that the multi-group co-evolution strategy brings two influences: (i) compared with single-group PSO, the MGPSO converges slower

Table 4 Summary of results for the case studies

Systems	Alg.	Cost (×1,000,000 US $)			Convergence		Time (s)			Iterations
		Best	Worst	Avg.	Trial	Sucd.	Min.	Max.	Avg.	
6-bus	MPSO[a]	0.200	0.231	0.203	10	9	2.750	3.203	2.845	≤200
	MPSO[b]	0.200	0.200	0.200	10	10	10.672	10.828	10.733	≤400
	LINGO	0.200	0.200	0.200	30	30	<1.000	<1.000	<1.000	—
	MGPSO	0.200	0.200	0.200	30	30	0.139	0.232	0.175	48∼57
24-bus	MPSO[a]	218.000	284.000	232.800	10	7	—	—	26.050	126∼245
	MPSO[b]	218.000	218.000	218.000	10	10	—	—	14.884	173∼342
	LINGO	218.000	243.000	227.200	30	19	16.000	38.000	26.900	—
	MGPSO	218.000	218.000	218.000	30	30	5.780	9.581	6.927	192∼208
6-bus	MPSO[a]	154.420	166.040	158.810	10	6	279.700	1146.300	—	≤1500
	MPSO[b]	154.420	154.420	154.420	10	10	633.140	1170.300	816.270	≤2500
	LINGO	154.420	164.752	158.597	30	11	200.000	1616.000	644.570	—
	MGPSO	154.420	154.420	154.420	30	30	79.142	113.800	95.070	375∼394
79-bus	LINGO	431.900	478.500	458.207	30	15	2279.000	65594.000	12661.200	—
	MGPSO	457.800	457.800	457.800	30	30	876.789	1096.293	1014.784	721∼755
118-bus	LINGO	915.800	967.600	933.138	30	13	2452.000	72384.000	14211.700	—
	MGPSO	929.400	929.400	929.400	30	30	926.813	1259.915	1110.269	742∼783

—: Data is not been reported by the literature.

[a] and [b]: The fastest version generated from [15] with convergence rate over 50 and 100%.

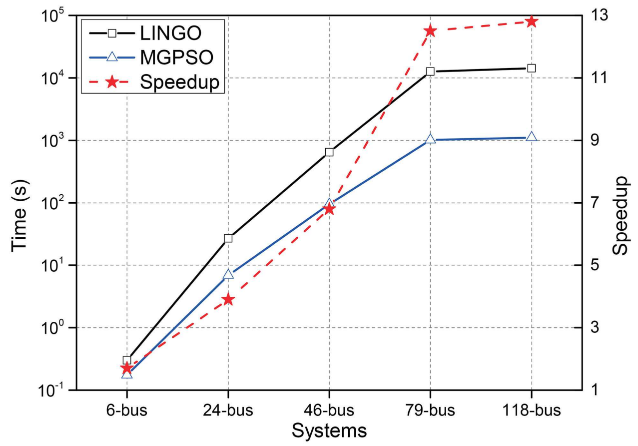

Fig. 9 *Speedup analysis between MGPSO and Lingo 11.0*

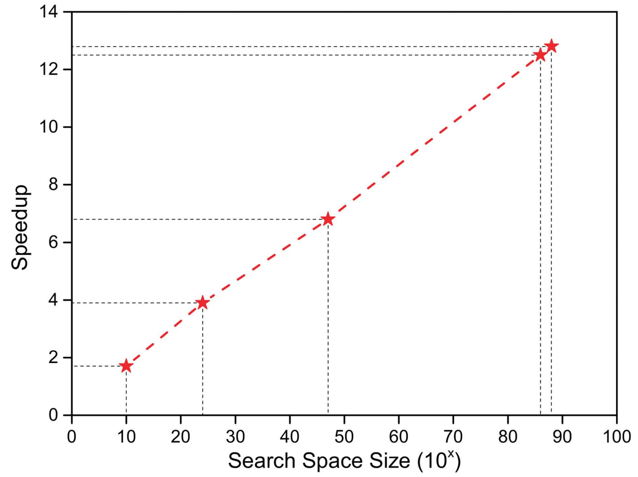

Fig. 10 *Relationship between the speedup and the search space size*

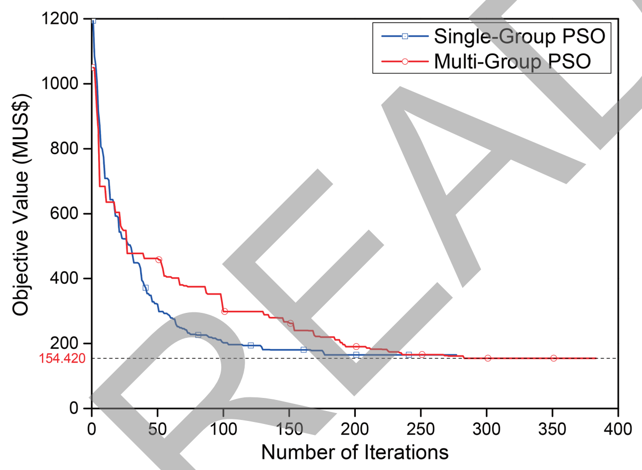

Fig. 11 *Convergence characteristic of single- and multi-group PSO for 46-bus system*

and the solution time is also longer; (ii) the quality of final solution obtained by MGPSO is better.

4.4.3 Performance evaluation of initialisation procedure: The performance evaluation of initialisation procedure cannot be distinguished with only one execution since the random sequence has been involved. After 30 times of experiment on the 46-bus system, MGPSO with and without the Sobol sequence gains a convergence rate of 100 and 91.4% respectively to the global optimal.

4.4.4 Performance evaluation of LU decomposition: The LU decomposition does not affect the convergence characteristic; it impacts the reduction of solution time of fitness evaluation at each iteration. As illustrated in Table 4, the average execution time for MGPSO is 95.070 s. In this section, two more experiments are carried out: (i) if the LU decomposition is replaced by the matrix inverse process, the solution time will increase to be 126.941 s; (ii) if the network analysis shown in Fig. 6 is also eliminated, an average time of 180.107 s should be required to finish the solution process.

4.4.5 New results: For the 79-bus system, a solution with the cost of 424.8 MUS was reported by Binato [31], however, 9.562 MW loss of load existed on bus #30. Interestingly, [34] also reported a solution of 444.49 MUS, which was indicated by the same author in [35] (where they updated the result to be 478.99 MUS without loss of load) that a total loss of load of about 37 MW was presented. In addition, a configuration with a cost of 454.1 MUS was illustrated by Robinson [32] without loss of load. In this work, a new result of 431.9 MUS without loss of load has been generated. Table 5 shows the details of each solution, where the solution with a cost of 457.8MUS is also explicated.

5 Conclusion

The candidate transmission circuits considered in the TEP problem are usually limited by the available computing resources. In this paper, five case studies with as many as 186 candidate transmission circuits considered are presented to verify the proposed MGPSO, showing that the method achieves considerable speedup compared to commercial software in execution time. The achieved speedup has a linear relationship with search space size, showing that the MGPSO algorithm is scalable. Performance evaluation has also been carried out on several enhancement strategies to distinguish their contribution. Future work will be concentrated on two aspects: for the first part, solving more complicated TEP problems with more realistic factors considered, such as generation redispatch, AC power flow, uncertainty, security constraints, etc.; for the second part, since PSO is highly suitable for massively parallel implementation, the algorithm efficiency can be improved within high performance computation context, for example, the utilisation of general-purpose graphics processing unit.

6 Acknowledgments

S. Huang was sponsored by the China Scholarship Council (CSC) under grant no. 201403170337. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Table 5 Solutions and costs for 79-bus system (×1,000,000 US $)

Solution	Cost
$n_{18-19}=1$, $n_{21-20}=1$, $n_{21-23}=1$, $n_{23-25}=1$, $n_{24-09}=2$, $n_{28-33}=1$, $n_{28-60}=1$, $n_{30-29}=1$, $n_{34-56}=1$, $n_{35-38}=1$, $n_{38-41}=1$, $n_{40-56}=1$, $n_{40-72}=1$, $n_{48-51}=1$, $n_{58-59}=1$, $n_{59-53}=1$, $n_{59-67}=1$, $n_{62-61}=2$, $n_{62-64}=1$, $n_{63-64}=1$, $n_{64-65}=1$, $n_{69-72}=1$.	424.800 [31]
$n_{18-19}=1$, $n_{21-20}=1$, $n_{21-23}=1$, $n_{24-09}=2$, $n_{25-60}=2$, $n_{30-29}=1$, $n_{34-56}=2$, $n_{34-64}=1$, $n_{35-38}=1$, $n_{38-41}=2$, $n_{40-41}=1$, $n_{40-56}=2$, $n_{40-72}=1$, $n_{48-51}=1$, $n_{48-71}=1$, $n_{58-59}=1$, $n_{59-53}=1$, $n_{59-67}=1$, $n_{63-64}=1$, $n_{64-65}=1$, $n_{69-72}=2$.	454.100 [32]
$n_{18-19}=1$, $n_{21-20}=1$, $n_{21-23}=1$, $n_{24-09}=2$, $n_{25-26}=2$, $n_{26-29}=1$, $n_{29-31}=1$, $n_{34-56}=1$, $n_{35-55}=1$, $n_{38-41}=1$, $n_{40-56}=1$, $n_{48-51}=1$, $n_{58-59}=1$, $n_{59-53}=1$, $n_{59-67}=1$, $n_{62-61}=2$, $n_{62-64}=1$, $n_{64-65}=1$.	457.800
$n_{18-19}=1$, $n_{21-20}=3$, $n_{21-23}=1$, $n_{23-25}=1$, $n_{24-09}=2$, $n_{28-33}=1$, $n_{28-60}=1$, $n_{30-29}=1$, $n_{34-56}=1$, $n_{35-38}=1$, $n_{38-41}=1$, $n_{40-56}=1$, $n_{40-72}=1$, $n_{48-51}=1$, $n_{58-59}=1$, $n_{59-53}=1$, $n_{59-67}=1$, $n_{62-61}=2$, $n_{62-64}=1$, $n_{64-65}=1$, $n_{69-72}=1$.	431.900

7 References

[1] Akbari, T., Tavakoli, M.: 'Approximated MILP model for AC transmission expansion planning: global solutions versus local solutions', *IET Gener. Transm. Distrib.*, 2016, **10**, (7), pp. 1563–1569

[2] Xing, H., Cheng, H., Zhang, Y., *et al.*: 'Active distribution network expansion planning integrating dispersed energy storage systems', *IET Gener. Transm. Distrib.*, 2016, **10**, (3), pp. 638–644

[3] Garver, L.: 'Transmission network estimation using linear programming', *IEEE Trans. Power Appl. Syst.*, 1970, **PAS-89**, (7), pp. 1688–1697

[4] Oertel, D., Ravi, R.: 'Complexity of transmission network expansion planning', *Energy Syst.*, 2014, **5**, (1), pp. 179–207

[5] Romero, R., Monticelli, A., Garcia, A., *et al.*: 'Test systems and mathematical models for transmission network expansion planning', *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, 2002, **149**, (1), pp. 27–36

[6] Haffner, S., Monticelli, A., Garcia, A., *et al.*: 'Branch and Bound algorithm for transmission system expansion planning using a transportation model', *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, 2000, **147**, (3), pp. 149–156

[7] Zhang, Y., Wang, R., Zhang, T., *et al.*: 'A model predictive control based operation management for a residential microgrid with considering forecast uncertainties and demand response strategies', *IET Gener. Transm. Distrib.*, 2016, **10**, (10), pp. 2367–2378

[8] Silva, I., Rider, J., Romero, R., *et al.*: 'Transmission network expansion planning with security constraints', *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, 2005, **152**, (6), pp. 828–836

[9] Latorre, G., Cruz, D., Areiza, M., *et al.*: 'Classification of publications and models on transmission expansion planning', *IEEE Trans. Power Syst.*, 2003, **18**, (2), pp. 938–946

[10] Hemmati, R., Hooshmand, A., Khodabakhshian, A.: 'Comprehensive review of generation and transmission expansion planning', *IET Gener. Transm. Distrib.*, 2013, **7**, (9), pp. 955–964

[11] Romero, R., Asada, E.N., Carreno, E., *et al.*: 'Constructive heuristic algorithm in branch-and-bound structure applied to transmission network expansion planning', *IET Gener. Transm. Distrib.*, 2007, **1**, (2), pp. 318–323

[12] Mortaz, E., Ledezma, L.F., Alcaraz, G.G., *et al.*: 'Transmission expansion planning using multivariate interpolation', *Electr. Power Syst. Res.*, 2015, **126**, pp. 87–99

[13] Mendonça, I.M., Junior, I.C.S., Marcato, A.L.M.: 'Static planning of the expansion of electrical energy transmission systems using particle swarm optimization', *Int. J. Electr. Power Energy Syst.*, 2014, **60**, pp. 234–244

[14] Jin, Y., Cheng, H., Yan, J., *et al.*: 'New discrete method for particle swarm optimization and its application in transmission network expansion planning', *Electr. Power Syst. Res.*, 2007, **77**, (3), pp. 227–233

[15] Murugan, P.: 'Modified particle swarm optimisation with a novel initialisation for finding optimal solution to the transmission expansion planning problem', *IET Gener. Transm. Distrib.*, 2012, **6**, (11), pp. 1132–1142

[16] Kamyab, G.R., Mahmood, F.F., Rashidinejad, M.: 'A PSO based approach for multi-stage transmission expansion planning in electricity markets', *Int. J. Electr. Power Energy Syst.*, 2014, **54**, pp. 91–100

[17] Gavela, P., Rueda, J.L., Vargas, A., *et al.*: 'Performance comparison of heuristic optimization methods for optimal dynamic transmission expansion planning', *Int. Trans. Electr. Energ. Syst.*, 2014, **24**, (10), pp. 1450–1472

[18] Rocha, M.C., Saraiva, J.T.: 'A discrete evolutionary PSO based approach to the multiyear transmission expansion planning problem considering demand uncertainties', *Int. J. Electr. Power Energy Syst.*, 2013, **45**, pp. 427–442

[19] Shayeghi, H., Mahdavi, M., Bagheri, A.: 'An improved DPSO with mutation based on similarity algorithm for optimization of transmission lines loading', *Energy Convers. Manage.*, 2010, **51**, (12), pp. 2715–2723

[20] Torres, S.P., Castro, C.A., Rider, M.J.: 'Transmission expansion planning by using DC and AC models and particle swarm optimization', in Fornarelli, G., Mescia, L. (Eds.): *Swarm Intelligence for Electric and Electronic Engineering*, (IGI Golbal, Hershey, PA, USA, 2013), pp. 260–284

[21] Karimi, E., Ebrahimi, A.: 'Inclusion of blackouts risk in probabilistic transmission expansion planning by a multi-objective framework', *IEEE Trans. Power Syst.*, 2015, **30**, (5), pp. 2810–2817

[22] Lee, Y., ElSharkawi, A.: '*Modern heuristic optimization techniques: theory and applications to power systems*' (Wiley, 2008)

[23] Talbi, E.: '*Metaheuristics: from design to implementation*' (Wiley, 2009)

[24] Hashimoto, M., Romero, R., Mantovani, S.: 'Efficient linear programming algorithm for the transmission network expansion planning problem', *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, 2003, **150**, (5), pp. 536–542

[25] Romero, R., Gallego, A., Monticelli, A.: 'Transmission system expansion planning by simulated annealing', *IEEE Trans. Power Syst.*, 1996, **11**, (1), pp. 364–369

[26] Kennedy, J., Eberhart, R.: 'Particle swarm optimization'. Proc. IEEE Int. Conf. Neural Network, Perth, Australia, November 1995, pp. 1942–1948

[27] The, V.L., Melab, N., Talbi, E.G.: 'GPU computing for parallel local search metaheuristic algorithms', *IEEE Trans. Computers*, 2013, **62**, (1), pp. 173–185

[28] 'The Sobol Quasirandom Sequence'. Available at https://people.sc.fsu.edu/jburkardt/cpp_src/sobol/sobol.html, accessed May 5, 2016

[29] Tarjan, R.: 'Depth-first search and linear graph algorithms', *SIAM J. Comput.*, 1972, **1**, (2), pp. 146–160

[30] Fang, R., Hill, D.: 'A new strategy for transmission expansion in competitive electricity markets', *IEEE Trans. Power Syst.*, 2003, **18**, (1), pp. 374–380

[31] Binato, S.: 'Optimal expansion of transmission networks by Benders decomposition and cutting planes'. *PhD thesis*, Federal University of Rio de Janeiro, 2000

[32] Robinson, L.: 'Meta-heuristics for electric power transmission network expansion planning problem with security restrictions'. *Master thesis*, Universidade Estadual Paulista, 2015

[33] 'IEEE 118-bus system data'. Available at http://motor.ece.iit.edu/data/IEEE118busdata figure.xls, accessed November 4, 2016

[34] Edson, L.S., Jorge, M.A., Gerson, C.O., *et al.*: 'Transmission network expansion planning under a Tabu Search approach', *IEEE Trans. Power Syst.*, 2001, **16**, (1), pp. 62–68

[35] Edson, L.S., Hugo, A.G., Jorge, M.A.: 'Transmission network expansion planning under an improved genetic algorithm', *IEEE Trans. Power Syst.*, 2000, **15**, (3), pp. 1168–1174