

Binary glowworm swarm optimization for unit commitment

Mingwei LI (✉), Xu WANG, Yu GONG,
Yangyang LIU, Chuanwen JIANG



Abstract This paper proposes a new algorithm—binary glowworm swarm optimization (BGSO) to solve the unit commitment (UC) problem. After a certain quantity of initial feasible solutions is obtained by using the priority list and the decommitment of redundant unit, BGSO is applied to optimize the on/off state of the unit, and the Lambda-iteration method is adopted to solve the economic dispatch problem. In the iterative process, the solutions that do not satisfy all the constraints are adjusted by the correction method. Furthermore, different adjustment techniques such as conversion from cold start to hot start, decommitment of redundant unit, are adopted to avoid falling into local optimal solution and to keep the diversity of the feasible solutions. The proposed BGSO is tested on the power system in the range of 10–140 generating units for a 24-h scheduling period and compared to quantum-inspired evolutionary algorithm (QEA), improved binary particle swarm optimization (IBPSO) and mixed integer programming (MIP). Simulated results distinctly show that BGSO is very competent in solving the UC problem in comparison to the previously reported algorithms.

Keywords Binary glowworm swarm optimization, Correction method, Priority list, Unit commitment

1 Introduction

Unit commitment is an important optimization problem in the power system. Its objective is to determine the on/off status of each unit and the economic dispatch of power demand in a scheduling period in order to minimize the total system production cost under generating units' constraints and power system's constraints. Since the unit commitment (UC) problem has the characteristic of high-dimension, discreteness and non-linearity, it takes lots of time to get the exact best solution of this problem by the enumeration method, and the computation time increases dramatically with the size of unit [1].

Because of its significant economic benefits, researchers around the world have done a lot of research and proposed many methods. Reference [2] used the dynamic programming method to solve the UC problem. In order to save the computation time, the units were classified and all the units formed different kinds of groups. As a result, the combinations of the units, as well as the computation time decreased. The extended priority list (EPL) method was introduced in [3]. The EPL method consisted of two steps. At first, disregarding the operational constraints, we got the original solutions by priority list (PL) algorithm very quickly, secondly, some heuristic processes were used to ensure that all the solutions satisfy the operational constraints. Reference [4] concentrated on the implementation aspects of Lagrangian relaxation (LR) method applied to realistic and practical UC problem, which aided in confirming the viability of this technique especially for large scale thermal UC programs. On this basis, [5] presented the enhanced adaptive Lagrangian relaxation (ELR) method with novel method to decide the on/off status of the units, new way of initializing the Lagrangian multipliers, unit classification, and adaptive adjustment of Lagrangian multiplier. As a result, the production cost was less expensive

CrossCheck date: 18 November 2014

Received: 18 November 2014 / Accepted: 20 November 2014 /
Published online: 9 December 2014
© The Author(s) 2014. This article is published with open access at
Springerlink.com
M. LI, X. WANG, Y. GONG, Y. LIU, C. JIANG, Department of
Electrical Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
(✉) e-mail: lmw0546@126.com



than the Lagrangian relaxation method. Furthermore, the CPU time is much smaller. Although these conventional optimization algorithms have the advantage of high speed and accuracy in solving small-scale UC problem, with the increase in the size of the generating units, the quality of the solution decreases and some of these algorithms are easy to fall into the “curse of dimensionality”.

Reference [6] presented a genetic algorithm (GA) to solve the UC problem. Since the selection scheme, mutation operation and the corresponding correction method were used, the GA provided flexibility in modeling both time-dependent and coupling constraints. Simulated annealing (SA) method had the probabilistic jumping property which existed in the whole searching process and varied with time. When it was applied to the UC problem, it helped to keep the diversity of the feasible solutions and improved the probability of getting the best solution [7, 8]. Particle swarm optimization (PSO) method, first proposed by Eberhart and Kennedy, was easy to code and did not have many parameters to adjust in comparison with the GA mentioned above. Therefore, it not only obtained the better solution, but also considered more constraints such as the realistic nonlinear time-dependent startup cost, limits of the ramp rate and the prohibited zone [9]. Instead of the random mutation in the PSO method, the improved particle swarm optimization algorithm (IPSO) takes into account more information of the particles, thus the particles had more probability of moving to the better solution. Penalty factor was applied to the solutions that violate any of the constraints [10]. On this basis, [11] presented a new improved binary PSO (IBPSO) method, which was used to deal with the on/off status of the units. Meanwhile, the Lambda-iteration method was adopted to dispatch the load economically. In the iterative process, some heuristic strategies were used to repair the solutions that violate the system constraints or operational constraints. Reference [12] proposed the evolutionary programming (EP) method. All the feasible solutions changed randomly and competed with each other; then the better solutions were selected and got into the next iteration. The power output of the units in the whole scheduling period was represented by a string of symbols. The quantum-inspired evolutionary algorithm (QEA) proposed in [13] was improved by integrating the quantum theory. To be specific, the on/off status of the units was represented by the quantum bits and used the rotation gates to keep the diversity of the feasible solutions and move to the better solutions. The best solution of the QEA method was less expensive than that of the previous methods and the execution time increased linearly with the size of the generating units. To a certain extent, these intelligent algorithms solve the problem brought by the augment of the scale of the generating units. However, they have the disadvantage of falling into local optimal solution prematurely. With the

population of the software CPLEX, the MIP method to solve the UC problem became very effective. However, the accuracy of convergence had great impact on the computation time and the quality of the solution [14].

Glowworm swarm optimization (GSO) is a new swarm intelligence optimization algorithm [15]. The optimization process is as follows. At first, all the glowworms are randomly generated in the search space. Each of them carries luciferin, which represents the brightness of the light send out by the glowworms, then they look for the glowworms that have higher brightness within their own range of view, and move towards one of them using the roulette approach. After the move, the luciferin of the glowworms is updated. In case of having too many glowworms within their view range, every one of them adjust their view range after the move. Many researchers have applied the GSO algorithm to solve practical problems. The 0–1 knapsack problems were effectively solved by using GSO algorithm [16]. The GSO algorithm was also used to find the optimal solution for the continuous optimization problem. The results above showed that GSO performed much better than many other algorithms, especially for different kinds of various global optimization problems [17].

This paper proposes a binary glowworm swarm optimization (BGSO) algorithm to solve the UC problem. Each glowworm in the BGSO algorithm is a T-N matrix that represents all the units' on/off status in the whole scheduling period. Since the on/off status of the units are binary variables, we propose the Hamming distance to represent the distance between the glowworms creatively, instead of the Euclidean distance adopted in the original GSO. Furthermore, we thought of a new way to update the on/off status of the units in the form of probability. Meanwhile, the Lambda-iteration method is adopted to solve the economic dispatch problem. The Lambda-iteration method and the BGSO algorithm are run at the same time for the purpose of finding the solution that has the least total production cost. Furthermore, the correction method and several adjustment techniques are proposed to ensure that the solutions are diverse in the iterative process and satisfy all the constraints.

This paper is organized as follows. The mathematical formulation of the UC problem including the objective function and the constraints is illustrated in Section 2. Section 3 describes the procedure and principle of GSO. Section 4 proposes the BGSO applied to the UC problem. Furthermore, the correction method to guarantee that the solutions are feasible and several techniques to keep the diversity of the solutions and contribute to better solutions are also illustrated. The BGSO method is tested with the number of generating units in the range of 10–140 and the results are compared with the other algorithms in Section 5. The conclusion is given in Section 6.

2 Formulation of UC problem

2.1 Objective function

The objective of the UC problem is to minimize the total production cost consisting of the generation cost and the start-up cost of the generating units under the circumstance where the operational constraints and the constraints of the generating units are satisfied in the scheduling period. The objective function is expressed as

$$F = \sum_{t=1}^T \sum_{i=1}^N [C_i(p_i^t) \cdot u_i^t + S_i u_i^t (1 - u_i^{t-1})] \tag{1}$$

where F is the total production cost; T the number of hours in the scheduling period; N the number of generating units; and u_i^t on/off status of the unit i at hour t , 1 represents the on status of the unit i at hour t , 0 represents the off status of the unit i at hour t . $C_i(p_i^t)$ is the generation cost function of unit i . It is normally a quadratic polynomial represented by

$$C_i(p_i^t) = a_i(p_i^t)^2 + b_i(p_i^t) + c_i \tag{2}$$

where p_i^t generation output of unit i at hour t ; and a_i, b_i, c_i are parameters of unit i .

S_i is the start-up cost of unit i which is related to the duration time of the off state of unit i . It can be expressed by

$$S_i = \begin{cases} H_{SCi} & M_{DTi} < X_{OFFi}^t \leq M_{DTi} + C_{SHi} \\ C_{SCi} & M_{DTi} + C_{SHi} < X_{OFFi}^t \end{cases} \tag{3}$$

where H_{SCi} is hot start-up cost of unit i ; C_{SCi} the cold start-up cost of unit i ; X_{OFFi}^t the duration time during which unit i keeps off status at hour t ; C_{SHi} cold start time of unit i ; and M_{DTi} the minimum down time of unit i .

2.2 Constraints

The constraints of the UC problem are listed as follows:

- 1) System power balance constraint

$$\sum_{i=1}^N u_i^t p_i^t = D^t \tag{4}$$

- 2) Spinning reserve constraint

$$\sum_{i=1}^N u_i^t p_i^{\max} \geq D^t + R^t \tag{5}$$

- 3) Generation limit constraint

$$p_i^{\min} \leq p_i^t \leq p_i^{\max} \tag{6}$$

- 4) Minimum up time constraint

$$(u_i^{t-1} - u_i^t)(X_{ONi}^{t-1} - M_{UTi}) \geq 0 \tag{7}$$

- 5) Minimum down time constraint

$$(u_i^t - u_i^{t-1})(X_{OFFi}^{t-1} - M_{DTi}) \geq 0 \tag{8}$$

where D^t is power demand at hour t ; R^t the spinning reserve at hour t ; p_i^{\max} the maximum power generation of unit i ; p_i^{\min} the minimum power generation of unit i ; M_{UTi} the minimum up time of unit i ; and X_{ONi}^t the duration time during which unit i keeps on status at hour t .

3 Glowworm swarm optimization

In the GSO algorithm, a group of glowworms are initialized randomly in the solution space of the objective function and each of them has the same value of luciferen. The brightness of the glowworm is proportional to the value of luciferen. Moreover, the fitness value of the glowworm is closely related to the luciferen. The larger the value of a glowworm's luciferen is, the more strongly it attracts the other glowworms within their own scope, which is called the local-decision range. In the iterative process, glowworm i moves towards one of the glowworms that both have better fitness value and are within the i th glowworm's local-decision range with a certain probability. Then the i th glowworm's local-decision range is adjusted for the purpose of controlling the quantity of the glowworms within it. The procedure of GSO algorithm is presented as:

- 1) Luciferin update phase

$$l_i(t) = (1 - \rho) \cdot l_i(t - 1) + \gamma \cdot J(x_i(t)) \tag{9}$$

where $x_i(t)$ is the location of glowworm i at iteration t ; $l_i(t)$ the luciferin of glowworm i at iteration t ; ρ the luciferin decay constant; γ the luciferin enhancement constant; and $J(x_i(t))$ the objective function of glowworm i .

- 2) Movement phase

Within glowworm i 's local-decision range, it selects glowworm j from all the glowworms that have larger value of luciferin by the way of roulette probability.

Roulette probability formula

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \tag{10}$$

Distance formula

$$d_{ij}(t) = ||x_i(t) - x_j(t)|| \tag{11}$$

Location update formula

$$x_i(t + 1) = x_i(t) + s \left(\frac{x_j(t) - x_i(t)}{d_{ji}(t)} \right) \tag{12}$$

where $N_i(t)$ is the numbers composed by all the glowworms that have larger value of luciferin within glowworm i 's



local-decision range; $d_{ij}(t)$ the distance between glowworm i and glowworm j ; $\|\cdot\|$ the standard Euclidean norm operator; and s the size of step.

3) Local-decision range update phase

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (13)$$

where $r_d^i(t)$ is the local-decision range of glowworm i at iteration t ; r_s the maximum local-decision range parameter used to control the rate of changing local-decision range; n_t the parameter used to control the number of glowworms within the local-decision range; and $|N_i(t)|$ the total number of glowworms that have the larger luciferin within the local-decision range.

4 BGSO for UC problem

4.1 Binary glowworm swarm optimization

The GSO algorithm is used to solve the problems that contain continuous variables. When it comes to the UC problem, the variables representing the on/off state of the units are binary, hence the BGSO is proposed to solve the UC problem. The modification of GSO is shown as follows.

1) Computation of distance

Instead of the Euclidean distance adopted in the GSO, Hamming distance is proposed to represent the distance between glowworm i and glowworm j . The Hamming distance between two glowworms is the number of locations where one has a “0” and the other a “1” [18]. It can be expressed by

$$hm_d_{ij}(t) = hamming_distance(x_i(t), x_j(t)) \quad (14)$$

2) Location update

The location of every glowworm is composed of m binary variables. In the location update process, the moving step is ignored and each dimension of glowworm i 's location is updated in the form of the probability. It can be expressed by

$$x_{i,k}(t+1) = \begin{cases} x_{i,k}(t) & r(k) < p_1 \\ x_{j,k}(t) & p_1 \leq r(k) \leq p_2 \\ round(\xi) & p_2 \leq r(k) \end{cases} \quad (15)$$

where $x_{i,k}(t)$ is the location of dimension k of glowworm i at iteration t ; $r(k)$ the parameter generated randomly $r(k) \in [0, 1](1 \leq k \leq m)$; p_1, p_2 the parameters used to control the update probability; and ξ the random variable generated between 0 and 1.

4.2 Initialization of glowworms for UC problem

In this paper, the initialization process is not only to generate a quantity of initial feasible solutions that satisfy all the constraints, but also to keep the diversity of the solutions. At the end of the initialization process, the total production cost corresponding to each glowworm is calculated, which is the basis of the following iterative process.

4.2.1 Structure of glowworms

Each glowworm is a $T \cdot N$ matrix, the elements of which represent all the units' on/off status in the whole scheduling period. For example, u_i^t in row t and column i represents the on/off status of unit i at hour t .

$$U = \begin{bmatrix} u_1^1 & u_2^1 & \cdots & u_N^1 \\ u_1^2 & u_2^2 & \cdots & u_N^2 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^T & u_2^T & \cdots & u_N^T \end{bmatrix}$$

4.2.2 Initialization of glowworms

In order to improve the quality of the initial solutions, the priority list and the decommitment of redundant unit are applied. In this paper, the priority list is based on the capacity of units. The unit that has the maximum capacity has the highest priority. If two units have the same capacity, the one that has lower average full-load cost has the higher priority. The procedure of the initialization of glowworms is as follows.

- Step 1: Set $t = 1$.
- Step 2: If $t = 1$, set the units whose initial status is a positive number and less than its minimum up time be on status. Else duplicate the on/off status of the units at hour $t - 1$.
- Step 3: Check the maximum output of the committed units. If the committed units cannot satisfy the spinning reserve constraint (5) at hour t , commit the unit in the ascending order of the priority list until (5) is satisfied.
- Step 4: Search for redundant unit that have the following properties in the descending order of the priority list at hour t .

- 1) This unit satisfies minimum up time constraint.
 - 2) After this unit is decommitted, the spinning reserve constraint (5) is still satisfied.
- If such a unit is found, it is decommitted with fifty percent probability, ensuring the diversity of the initial solutions.

Step 5: Update the status of the unitstatus when they meet the

$$\begin{cases} X_{ONi}^t = \begin{cases} X_{ONi}^{t-1} + 1 & \text{if } u_i^t = 1 \\ 0 & \text{if } u_i^t = 0 \end{cases} \\ X_{OFFi}^t = \begin{cases} 0 & \text{if } u_i^t = 1 \\ X_{OFFi}^{t-1} + 1 & \text{if } u_i^t = 0 \end{cases} \end{cases} \quad (16)$$

Step 6: If $t < T$, set $t = t + 1$ and return to step 2. Else go to step 7.

Step 7: Computation of total production cost
The total production cost consists of generation cost and start-up cost. These two parts are discussed separately as follows.

1) Generation cost

We have already got the on/off status of the units at every hour so we can apply the Lambda-iteration method to dispatch the load economically; then use (2) to compute the generation cost of each hour

2) Start-up cost

According to the changing time of the units' on/off status we can use (3) to compute the start-up cost very easily.

Step 8: Update the value of luciferin by using (9).

4.3 Iterative process for UC problem

After the initialization of the glowworms, the glowworms move in the iterative process. However, they may move out of the solution space, which means that the solutions violate at least one of the constraints so the correction method is adopted to keep the solutions feasible. Furthermore, several techniques are proposed in case of falling into the local optimal solution.

4.3.1 Formation of new glowworm by BGSO

In the iterative process, each of the glowworms moves towards another glowworm. The procedure is as follows.

Take glowworm i for example.

Step 1: Computation of the hamming distances between glowworm i and the other glowworms by using (14).

Step 2: Since the objective function is to minimize the production cost, glowworm i is attracted by the glowworms that have less luciferin within its own local-decision range.

Step 3: Use (10) to compute the possibility of moving to glowworm j .

Step 4: Select a glowworm with the roulette approach and use (15) to update the location of glowworm i .

4.3.2 Correction of newly formed glowworm

Instead of the penalty factor, the correction method is adopted to make the newly formed glowworm satisfy all the constraints.

Take the units' on/off status at time t for example.

Step 1: Set the units that violate (7) be on status and the units that violate (8) be off status at hour t .

Step 2: Set the units be off status when they meet the following two conditions.

- 1) The load at hour t is less than the load at hour $t - 1$.
- 2) The unit is off status at hour $t - 1$.

Step 3: Check whether the spinning reserve constraint (5) is satisfied.

If not, firstly commit the units that are on status at hour $t - 1$ ($t > 1$) and off status at hour t ; then commit the units in the ascending order of the priority list until (5) is satisfied.

Step 4: Update the status of the unit as shown in (16).

4.3.3 Adjustment techniques of newly formed glowworm

In the iterative process these techniques are applied to the feasible solutions in order to provide the probability of better solution and guarantee the diversity of the solutions. The detailed discussion is shown below.

1) Decommitment of redundant unit

Step 1: Search for the redundant unit that have the following two properties in the descending order of the priority list at hour t .

- a) This unit satisfies (7).
- b) After this unit is decommitted, the spinning reserve constraint (5) is still satisfied.

Step 2: If such a unit is found, use (17) to determine this unit's on/off status at hour t .

$$u_i^t = \begin{cases} 0 & \text{rand} < (j - 0.5)/M_{DTi} \\ 1 & \text{rand} \geq (j - 0.5)/M_{DTi} \end{cases} \quad (17)$$



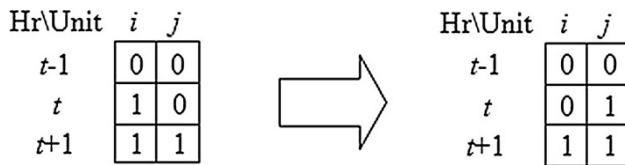


Fig. 1 Switch of the commitment order



Fig. 2 Switch of the decommitment order

The way to decide the value of *j*.

- a) Set $k = 1, j = 1$.
- b) If $k + t = 25$, set $j = M_{DTi}$ and break.
Else if $D^{t+k} > D^t$, set $j = k$ and break.
- c) If $k \leq M_{DTi}$, set $k = k + 1$ and return to step b).
Else set $j = k$ and break.

Equation (17) takes into account the load information in the following hours so it is more likely to produce better solution; moreover, the usage of the form of probability and random number ensures the diversity of the feasible solutions.

2) Conversion from cold start to hot start

Normally, cold start cost is much larger than hot start cost. Therefore, for the unit *i* that is committed at hour *t* when it has been kept off status for exactly $M_{DTi} + C_{SHi} + 1$ hours, if it is committed one hour ahead, the total production cost may decrease. The detailed procedure is as follows

Step 1: Calculate the generation cost at hour *t* - 1 represented by C_{g1} .

- Step 2: Commit unit *i* at hour *t* - 1 and recalculate the generation cost at hour *t* - 1 represented by C_{g2} .
- Step 3: Compare $C_{g2} - C_{g1}$ and the difference between cold start cost and hot start cost of unit *i*. If the latter is larger, it is reasonable to commit unit *i* one hour ahead, which means unit *i*'s conversion from cold start to hot start.

3) Switch of two units' commitment order in the adjacent two hours

Take the unit *i* committed at hour *t* and the unit *j* committed at hour *t* + 1 shown in Fig. 1 for example. If the commitment of unit *j* instead of unit *i* at hour *t* does not violate (5) and (8), then compute the total production cost at hour *t* and *t* + 1 in these two different situations and choose the better one with seventy percent probability.

4) Switch of two units' decommitment order in the adjacent two hours

Take the unit *j* decommitted at hour *t* and the unit *i* decommitted at hour *t* + 1 shown in Fig. 2 for example. If the decommitment of unit *i* instead of unit *j* at hour *t* does not violate (5) and (7), then compute the generation cost at hour *t* in these two different situations and choose the better one with seventy percent probability.

5) Replacement of the committed unit based on the minimum up time

If unit *i* is committed at hour *t* and decommitted at hour *t* + M_{UTi} , which indicates that unit *i* may be redundant from hour *t* + 1 to *t* + M_{UTi} , replace unit *i* with the units that have less minimum up time and satisfy (8) at hour *t* with fifty percent probability.

4.4 Implementation of BGSO for solving UC problem

The procedure of the proposed BGSO for solving UC problem is presented as follows.

Table 1 10-unit system data

Unit	P_{max}/MW	P_{min}/MW	$a/(\$/MW^2h)$	$b/(\$/MWh)$	$c/(\$/h)$	Min up/h	Min dn/h	Hot start cost/\$	Cold start cost/\$
1	455	150	0.00048	16.19	1000	8	8	4500	9000
2	455	150	0.00031	17.26	970	8	8	5000	10000
3	130	20	0.002	16.6	700	5	5	550	1100
4	130	20	0.00211	16.5	680	5	5	560	1120
5	162	25	0.00398	19.7	450	6	5	900	1800
6	80	20	0.00712	22.26	370	3	3	170	340
7	85	25	0.00079	27.74	480	3	3	260	520
8	55	10	0.00413	25.92	660	1	1	30	60
9	55	10	0.00222	27.27	665	1	1	30	60
10	55	10	0.00173	27.79	670	1	1	30	60

- Step 1: Set counter = 1.
- Step 2: Initialize the glowworms as in Section 4.2.2.
- Step 3: Use (9) to update every glowworm’s luciferin value.
- Step 4: Every glowworm move towards one of the other glowworms that have less production cost.
- Step 5: Modify the glowworms as in Section 4.3.2 to make them satisfy all the constraints.
- Step 6: Adjust the glowworms as in Section 4.3.3.
- Step 7: Calculate every glowworm’s total production cost.
- Step 8: Use (13) to update their local-decision range.
- Step 9: If counter < maximum iterations, counter = counter + 1 and go to step 3. Else, go to step 10.
- Step 10: Display the optimal solution.

Table 2 Load demand

Hour	Demand/MW	Hour	Demand/MW
1	700	13	1400
2	750	14	1300
3	850	15	1200
4	950	16	1050
5	1000	17	1000
6	1100	18	1100
7	1150	19	1200
8	1200	20	1400
9	1300	21	1300
10	1400	22	1100
11	1450	23	900
12	1500	24	800

5 Numerical results

The BGSO algorithm is tested on the power system with 10, 20, 40, 60, 80, 100, 120 and 140 generating units in the 24-h scheduling period. The 10-unit data is shown in Table 1 and the power demand in the scheduling period is shown in Table 2. The 20, 40, 60, 80, 100, 120 and 140-unit data are obtained by duplicating the 10-unit data, whereas the power demand is proportional to the number of units. In Addition, the spinning reserve is set to be 10% of power demand.

Table 3 Unit output of the 20-unit system’s best solution

Hour	Generating unit																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	455	455	245	245	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	455	455	295	295	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	455	455	382.5	382.5	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0
4	455	455	455	455	0	0	0	0	40	40	0	0	0	0	0	0	0	0	0	0
5	455	455	455	455	0	0	130	0	25	25	0	0	0	0	0	0	0	0	0	0
6	455	455	425	425	130	0	130	130	25	25	0	0	0	0	0	0	0	0	0	0
7	455	455	455	455	130	0	130	130	45	45	0	0	0	0	0	0	0	0	0	0
8	455	455	455	455	130	130	130	130	30	30	0	0	0	0	0	0	0	0	0	0
9	455	455	455	455	130	130	130	130	97.5	97.5	20	20	25	0	0	0	0	0	0	0
10	455	455	455	455	130	130	130	130	162	162	33	33	25	25	10	10	0	0	0	0
11	455	455	455	455	130	130	130	130	162	162	73	73	25	25	10	10	10	10	0	0
12	455	455	455	455	130	130	130	130	162	162	80	80	25	25	43	43	10	10	10	10
13	455	455	455	455	130	130	130	130	162	162	33	33	25	25	10	10	0	0	0	0
14	455	455	455	455	130	130	130	130	97.5	97.5	20	20	25	0	0	0	0	0	0	0
15	455	455	455	455	130	130	130	130	30	30	0	0	0	0	0	0	0	0	0	0
16	455	455	310	310	130	130	130	130	25	25	0	0	0	0	0	0	0	0	0	0
17	455	455	260	260	130	130	130	130	25	25	0	0	0	0	0	0	0	0	0	0
18	455	455	360	360	130	130	130	130	25	25	0	0	0	0	0	0	0	0	0	0
19	455	455	455	455	130	130	130	130	30	30	0	0	0	0	0	0	0	0	0	0
20	455	455	455	455	130	130	130	130	162	162	43	43	0	0	10	10	10	10	10	0
21	455	455	455	455	130	130	130	130	105	105	20	20	0	0	10	0	0	0	0	0
22	455	455	455	455	0	0	130	0	105	105	20	20	0	0	0	0	0	0	0	0
23	455	455	432.5	432.5	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0
24	455	455	345	345	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Table 4 Cost of the 20-unit system’s best solution

Hour	Generation cost	Start-up cost	Spinning reserve	On/off status
1	27366.26	0	420	11110000000000000000
2	29109.00	0	320	11110000000000000000
3	33111.24	900	282	11110000100000000000
4	37195.34	900	244	11110000110000000000
5	39457.23	560	274	11110010110000000000
6	44157.72	2220	334	11111011110000000000
7	46008.84	0	234	11111011110000000000
8	48300.68	1100	264	11111111110000000000
9	53838.78	1200	309	11111111111100000000
10	60115.10	640	304	11111111111111100000
11	63832.12	120	314	11111111111111111000
12	67780.33	120	324	11111111111111111111
13	60115.11	0	304	11111111111111110000
14	53838.78	0	309	11111111111110000000
15	48300.68	0	264	11111111100000000000
16	43027.32	0	564	11111111100000000000
17	41283.65	0	664	11111111100000000000
18	44774.09	0	464	11111111100000000000
19	48300.68	0	264	11111111100000000000
20	61047.05	640	299	11111111111001111110
21	53891.99	0	279	11111111111100100000
22	44328.11	0	234	11110010111100000000
23	34862.51	0	182	11110000100000000000
24	30854.84	0	220	11110000000000000000

Parameters are set as follows: the number of glowworms is 50; the luciferin decay constant $\rho = 0.4$; the luciferin enhancement constant $\gamma = 0.6$; the local-decision range is twice the number of the units; $\beta = 0.08$; $n_r = 5$; $p_1 = 0.1$; $p_2 = 0.9$. The program is written in MATLAB R2011a and executed on a 2.5 GHz CPU with 4-GB RAM personal computer. In order to have a comprehensive understanding of the BGSO method, 50 trials are done on every test system.

Since the best solution of the 10-unit system of BGSO is the same as that of QEA, the units’ power output of the best solution can be seen in [13]. The best solution of the 20-unit system is shown in Table 3 and 4, which have never been illustrated in detail before. We can see that the generation cost in the scheduling period is 1114879 and the start-up cost is 8400 so the total production cost is 1123297.

The best, worst and mean values of the total production cost, together with the mean computation time by MIP, QEA, IBPSO and BGSO for different test systems are shown in Table 5. We can see that the best solution of the BGSO algorithm is better in most of the test systems and

Table 5 Comparison of simulation results for different systems

Unit	Algorithm	Cost			Mean time
		Best	Worst	Mean	
10	MIP [13]	564647			2
	QEA [12]	563938	564672	563969	19
	IBPSO [10]	563977	565312	564155	27
20	BGSO	563938	564226	563952	3
	MIP [13]	1123908			5
	QEA [12]	1123607	1125715	1124689	28
40	IBPSO [10]	1125216	1125730	1125448	55
	BGSO	1123297	1124081	1123771	12
	MIP [13]	2243020			11
60	QEA [12]	2245557	2248296	2246728	43
	IBPSO [10]	2248581	2249302	2248875	110
	BGSO	2242882	2244573	2243582	31
80	MIP [13]	3361614			29
	QEA [12]	3366676	3372007	3368220	54
	IBPSO [10]	3367865	3368779	3368278	172
100	BGSO	3361683	3364103	3363115	52
	MIP [13]	4483194			38
	QEA [12]	4488470	4492839	4490126	66
120	IBPSO [10]	4491083	4492686	4491681	235
	BGSO	4482003	4486739	4484513	76
	MIP [13]	5601857			47
140	QEA [12]	5609550	5613220	5611797	80
	IBPSO [10]	5610293	5612265	5611181	295
	BGSO	5601281	5608327	5604186	104
120	BGSO	6722634	6732546	6726644	128
140	BGSO	7891543	7905542	7898763	154

the best solution of BGSO algorithm is very close to that of the MIP method in the 60-unit test system. From Fig. 3, we can see that the proposed method is faster than the IBPSO method in all the test systems and QEA algorithm in 10, 20, 40 and 60-unit test systems. Although the calculation time of BGSO is longer than that of the MIP method, the calculation time of BGSO increases almost linear with the number of the units, which means that it has the capacity of solving large-scale UC problems.

6 Conclusion

A BGSO has been proposed for solving the UC problem. The distance between the glowworms is represented by the Hamming distance instead of the Euclidean distance and the update of the glowworm’s location is expressed by the way of probability. The priority list and decommitment of redundant unit make a big contribution to the high quality

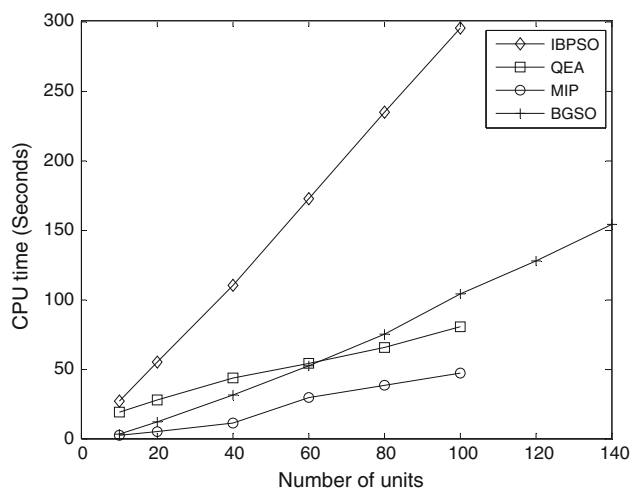


Fig. 3 Compare of different algorithms' computation time

and diversity of the initial solutions. Furthermore, in the iterative process, the correction method and several adjustment techniques help to search for the better feasible solutions. The simulated results show that the total production cost of BGSO is less expensive than those of the other methods in the range of 10–100 units except for the MIP method in the 60-unit system. In addition, the CPU time of BGSO increases almost linear with the size of the units, which is favorable for the large-scale power systems.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- [1] Wood AJ, Wollenberg BF (2012) Power generation, operation, and control. Wiley, New York
- [2] Snyder WL, Powell HD, Rayburn JC (1987) Dynamic programming approach to unit commitment. *IEEE Trans Power Syst* 2(2):339–348
- [3] Senjyu T, Shimabukuro K, Uezato K et al (2003) A fast technique for unit commitment problem by extended priority list. *IEEE Trans Power Syst* 18(2):882–888
- [4] Virmani S, Adrian EC, Imhof K et al (1989) Implementation of a Lagrangian relaxation based unit commitment problem. *IEEE Trans Power Syst* 4(4):1373–1380
- [5] Ongsakul W, Petcharakas N (2004) Unit commitment by enhanced adaptive Lagrangian relaxation. *IEEE Trans Power Syst* 19(1):620–628
- [6] Kazarlis SA, Bakirtzis AG, Petridis V (1996) A genetic algorithm solution to the unit commitment problem. *IEEE Trans Power Syst* 11(1):83–92
- [7] Wong KP, Wong YW (1995) Thermal generator scheduling using hybrid genetic/simulated-annealing approach. *IEE Proc-Gener Transm Distrib* 142(4):372–380
- [8] Wong KP, Wong YW (1996) Combined genetic algorithm/simulated annealing/fuzzy set approach to short-term generation

- scheduling with take-or-pay fuel contract. *IEEE Trans Power Syst* 11(1):128–136
- [9] Gaing ZL (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Trans Power Syst* 18(3):1187–1195
- [10] Jeong YW, Park JB, Jang SH et al (2010) A new quantum-inspired binary PSO: Application to unit commitment problems for power systems. *IEEE Trans Power Syst* 25(3):1486–1495
- [11] Yuan XH, Nie H, Su AJ et al (2009) An improved binary particle swarm optimization for unit commitment problem. *Expert Syst Appl* 36(4):8049–8055
- [12] Juste KA, Kita H, Tanaka E et al (1999) An evolutionary programming solution to the unit commitment problem. *IEEE Trans Power Syst* 14(4):1452–1459
- [13] Lau TW, Chung CY, Wong KP et al (2009) Quantum-inspired evolutionary algorithm approach for unit commitment. *IEEE Trans Power Syst* 24(3):1503–1512
- [14] Wang N, Zhang LZ, Xie GH (2010) An improved mixed integer quadratic programming algorithm for unit commitment. *Autom Electr Power Syst* 34(15):28–32 (in Chinese)
- [15] Krishnanand KN, Ghose D (2009) Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intell* 3(2):87–124
- [16] Gong QQ, Zhou YQ, Yang Y (2011) Artificial glowworm swarm optimization algorithm for solving 0-1 knapsack problem. *Adv Mater Res* 143(144):166–171
- [17] Wu B, Qian CH, Ni WH et al (2012) The improvement of glowworm swarm optimization for continuous optimization problems. *Expert Syst Appl* 39(7):6335–6342
- [18] Steane AM (1996) Error correcting codes in quantum theory. *Phys Rev Lett* 77(5):793–797

Mingwei LI received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China in 2012. In 2013, he was a visiting student with the Department of Electrical and Computer Engineering, Institution of Georgia Technology, Atlanta, USA. He is currently pursuing his M.S. degree in Shanghai Jiao Tong University, Shanghai, China. His research interests include power system optimization, renewable energy and electricity market.

Xu WANG received the B.S. degree in electrical engineering from Southeast University, Nanjing, China in 2010. He is currently pursuing his Ph.D. degree in Shanghai Jiao Tong University, Shanghai, China. His research interests include electricity market and power system risk management.

Yu GONG received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China in 2012. He is currently pursuing his M.S. degree in Shanghai Jiao Tong University, Shanghai, China. His research interests include electricity and low carbon electricity.

Yangyang LIU received the B.S. and M.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China in 2009 and 2011. He is currently pursuing his Ph.D. degree in Shanghai Jiao Tong University, Shanghai, China. His research interests include electricity market and power system optimization.

Chuanwen JIANG received the M.S. and Ph.D. degrees from Huazhong University of Science and Technology, Hubei, China, in 1996 and 2000, respectively. He is currently a professor with Shanghai Jiao Tong University, Shanghai, China. His research interests include power system optimization, load forecast in power system, and electricity market.

