

求解机组组合问题的改进离散粒子群算法

刘 涌, 侯志俭, 蒋传文

(上海交通大学电子信息与电气工程学院, 上海市 200030)

摘要: 电力系统机组组合问题是一个高维数、离散、非线性的大规模复杂工程优化问题。文中提出了一种基于改进离散粒子群优化算法求解机组组合问题的新方法。首先采用新的策略生成粒子, 以保证所有生成的粒子均为满足基本约束条件的可行解, 使整个算法只在可行解区域进行优化搜索; 然后引入优化窗口的概念和启发式的规则以缩短计算时间和提高优化精度。仿真结果表明所提出的算法具有解的质量高、收敛速度快的特点, 充分证明了它能很好地解决机组组合问题。

关键词: 机组组合; 离散粒子群优化算法; 优化窗口; 启发式规则

中图分类号: TM732

0 引言

电力系统机组优化组合问题是在满足系统供电要求和机组运行技术要求的情况下, 确定未来一个调度周期(一般为 24 h)内各机组的开停机时间和负荷的经济分配^[1]。它是一个高维数、非凸、离散、非线性的复杂工程优化问题, 很难找到理论上的全局最优解。但由于其能带来显著的效益, 人们一直在探索, 提出了各种解决方法, 如优先顺序法^[2]、动态规划法^[3,4]、拉格朗日松弛法^[5,6]、遗传算法(GA)^[7,8]、模拟退火算法^[9]等。优先顺序法简便快速, 但其过于简化以致得到的结果往往距最优解较远; 动态规划法则由于其“维数灾”的痼疾而难以处理大规模系统; 拉格朗日松弛法其算子选择比较困难, 选择不当会使收敛速度缓慢, 甚至不收敛。

文献[10]提出了一种新的进化计算方法——粒子群优化(PSO)算法。它与 GA 类似, 是一种基于迭代的优化工具。系统初始化为一组随机解, 通过迭代搜寻最优值。它没有 GA 用的交叉和变异, 而是利用在解空间追随最优的粒子进行搜索。与 GA 相比, PSO 算法的优势在于简单、容易实现, 且不必调整许多参数, 非常适于工程应用。文献[11]将离散粒子群优化(BPSO)算法首次应用到机组组合问题的求解中, 不过该文存在一些明显不足: ①其仿真结果并不准确, 因为漏算了初始时刻中开机机组的开机费用, 而且其仿真结果并不比文献[7]中 GA 的结果更好; ②其粒子的生成过于耗时, 从而影响了整个算法的计算速度。文献[12]对文献[11]的算法进行了改进, 将算法分为内外 2 层嵌套优化, 加入了修

补策略和启发式变异算子, 但该文算法对相同算例的优化结果并不明显优于文献[7]GA 的结果。

本文则在 BPSO 算法的基础上, 采取新的策略来产生粒子, 以保证只在可行解空间内寻优, 同时引入了优化窗口的概念和启发式规则以进一步提高算法的精度。仿真实验的结果表明, 这种算法简单快速, 优化效果比现有文献的结果好, 适用于解决机组组合问题。

1 机组优化组合的描述

机组优化组合问题在数学上可以表示为满足不同约束条件的混合整数非线性规划模型。其目标为在一个调度周期内总的运行费用最小, 运行费用包括发电费用和启动费用。目标函数描述如下:

$$\min F = \sum_{j=1}^T \sum_{i=1}^N [C_i(P_{ij})u_{ij} + S_{ij}u_{ij}(1 - u_{i(j-1)})] \quad (1)$$

系统功率平衡约束:

$$\sum_{i=1}^N P_{ij}u_{ij} - P_{Dj} = 0 \quad j = 1, 2, \dots, T \quad (2)$$

旋转备用约束:

$$\sum_{i=1}^N P_i^{\max} u_{ij} \geq P_{Dj} + P_{Rj} \quad (3)$$

机组出力上下限约束:

$$P_i^{\min} \leq P_{ij} \leq P_i^{\max} \quad (4)$$

最小开机时间和最小停机时间约束:

$$\begin{cases} t_{ij} \geq T_{MU,i} & t_{ij} > 0 \\ -t_{ij} \geq T_{MD,i} & t_{ij} < 0 \end{cases} \quad (5)$$

式中: N 为机组数; T 为调度时段数; F 为总运行费用; P_{ij} 为机组 i 在时段 j 的出力值; $C_i(P_{ij})$ 为机组 i 在时段 j 的发电费用, 通常用二次函数表示,

$C_i(P_{ij}) = a_i + b_i P_{ij} + c_i P_{ij}^2$, a_i, b_i, c_i 分别为机组 i 的燃料费用系数; P_{Dj} 为时段 j 的系统负荷; P_{Rj} 为时段 j 的系统旋转备用; $P_i^{\min}$ 和 $P_i^{\max}$ 分别为机组 i 的最小与最大技术出力; $T_{MU,i}$ 为机组 i 的最小连续开机时间; $T_{MD,i}$ 为机组 i 的最小连续停机时间; u_{ij} 为机组 i 在时段 j 的启停状态, 0 为停机, 1 为开机; S_{ij} 为机组 i 在时段 j 的启动费用, 可以表示为:

$$S_{ij} = \begin{cases} C_{HS,i} & T_{MD,i} < -t_{ij} \leq T_{MD,i} + T_{CS,i} \\ C_{CS,i} & -t_{ij} > T_{MD,i} + T_{CS,i} \end{cases}$$

式中: $C_{HS,i}$, $C_{CS,i}$ 分别为机组 i 的热启动成本和冷启动成本; t_{ij} 为机组 i 在时段 j 时已连续运行(或停运)的时段数, 正值表示连续运行累计时间, 负值表示连续停机累计时间; $T_{CS,i}$ 为机组 i 的冷启动时间。

2 BPSO 算法

BPSO 算法的数学描述如下^[13]:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{best,id} - x_{id}^k) + c_2 r_2 (g_{best,d} - x_{id}^k) \quad (6)$$

$$x_{id}^{k+1} = \begin{cases} 1 & r_3 < S(v_{id}^{k+1}) \\ 0 & \text{其他情况} \end{cases} \quad (7)$$

式中: v_{id}^k 为粒子 i 在第 k 次迭代时的速度, 一般有 $v_i^{\min} \leq v_i \leq v_i^{\max}$; ω 为惯性系数, 一般取 0.4~0.9, 按 $\omega = \omega^{\max} - l(\omega^{\max} - \omega^{\min})/l^{\max}$ 线性递减^[14], l 为迭代次数, $l^{\max}$ 为最大迭代次数; c_1, c_2 为加速常数, 一般取 2; r_1, r_2, r_3 为在 0~1 之间均匀分布的随机数; x_{id}^k 为粒子 i 在第 k 次迭代时的位置; $p_{best,id}$ 为粒子 i 本身找到的最优解, 即个体极值; g_{best} 为整个种群找到的最优解, 即全局极值; $S(v) = 1/(1 + e^{-v})$ 为 sigmoid 转换函数。

3 基于 BPSO 的机组组合求解算法

本文在 BPSO 算法的基础上, 采取新的策略来产生粒子, 以保证只在可行解空间内进行寻优, 同时引入了优化窗口的概念和启发式规则以进一步提高算法的精度。

3.1 编码表示目标函数

用 0 或 1 表示各机组的启停状态, 在某一调度时刻, 所有机组的状态就由一组 0 或 1 组成的子串来表示。所有调度时刻的子串即组合成整个调度时刻所有机组的启停状态, 如图 1 所示。所以, 对于一个求解 T 个调度时段内 N 台机组的机组优化组合问题, 整个个体的长度为 $N \times T$ 。

3.2 适应值函数

本文直接采用机组优化组合问题的目标函数作为粒子群算法的适应值函数。适应值最低的粒子运行费用最低, 就是所求的最优解。

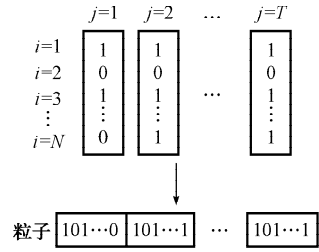


图 1 机组状态的编码表示
Fig. 1 Coding of the unit's state

3.3 粒子的产生策略

本文采取新的策略来生成粒子, 以保证产生的粒子均为可行解, 使算法始终在可行解区域内寻优。

从前面的编码可知, 整个调度时段全部机组状态是由 u_{ij} 的组合来表示的。为统一符号表示, 将应用于机组组合的 BPSO 的公式修改为:

$$v_{ij,m}^{k+1} = \omega v_{ij,m}^k + c_1 r_1 (p_{best,ij,m} - u_{ij,m}^k) + c_2 r_2 (g_{best,ij} - u_{ij,m}^k) \quad (8)$$

$$u_{ij,m}^{k+1} = f(v_{ij,m}^{k+1}) = \begin{cases} 1 & r_3 < S(v_{ij,m}^{k+1}) \\ 0 & \text{其他情况} \end{cases} \quad (9)$$

式中: $u_{ij,m}^k$ 表示第 m 个粒子在第 k 次迭代中机组 i 在调度时刻 j 的启停状态。

由于机组启停状态还要受到机组启停时间约束的限制, 因此将式(5)与式(9)合并为:

$$u_{ij,m}^{k+1} = \begin{cases} 1 & 0 < t_{ij,m}^{k+1} < T_{MU,i} \\ 0 & 0 < -t_{ij,m}^{k+1} < T_{MD,i} \\ f(v_{ij,m}^{k+1}) & \text{其他情况} \end{cases} \quad (10)$$

这样, 各调度时刻所有机组的状态都将满足机组启停时间约束。

由前面的编码表示可知, 粒子的产生是在每个调度时刻内依次安排机组状态后组合而成的。在安排完各调度时刻内所有的机组状态后, 还有系统容量约束需要考虑。式(4)表示的不等式约束与机组状态显式无关, 但可以将式(2)和式(4)联合起来得到如下不等式约束:

$$\sum_{i=1}^N P_i^{\min} u_{ij} \leq P_{Dj} \quad (11)$$

若所有状态为 1(开机)的机组的最大技术出力之和小于当前时刻的总负荷与旋转备用之和, 即不满足式(3), 则需要按优先顺序依次将状态为 0 且满足最小停运时间约束的机组开机, 直至满足式(3)为止; 反之, 若所有状态为 1(开机)的机组的最小技术出力之和大于当前时刻的总负荷, 即不满足式(11), 则同样需要按照优先顺序将状态为 1 且满足最小运行时间约束的机组停机, 直至满足式(11)为止。

由上面介绍的粒子产生方式可能会出现如下情况: 在某个调度时刻, 所有能够开机的机组(即满足

机组启停时间约束的机组)的最大技术出力之和仍然小于当前时刻的总负荷与旋转备用之和。这种情况产生的原因是,在之前的调试时刻将某些机组停机导致当前时刻虽然负荷增长较大,但由于启停时间约束的限制无法将其开机而造成的。因此,还有如下的约束条件需要判断:

$$\sum_{i=1}^N P_i^{\max} \theta(t_{ij}) \geq P_{Dj} + P_{Rj} \quad (12)$$

式中:

$$\theta(t_{ij}) = \begin{cases} 1 & -t_{ij} \geq T_{MD,i} \\ 0 & \text{其他} \end{cases}$$

因此,在每个调度时刻的机组状态确定之后,还要往后顺推一段时间(即所有不满足最小连续停机时间约束的停运机组中还需要保持停机状态的时段数的最大值),判断其是否满足式(12)所代表的约束。如果不满足则需要重新生成该时刻的所有机组状态,直至满足为止。

3.4 优化窗口与启发式规则

为了提高整个算法的精度,本文借鉴了优先顺序法和文献[4]的思想,引入了优化窗口的概念和启发式的规则。首先,所有的机组按照一定的优先顺序进行排序。那些平均成本微增率较低且启动成本较高的机组的优先级别较高,反之则较低。然后,在每个调度时刻,预先按照这个排好的优先顺序依次安排机组,直至满足系统负荷需求和旋转备用为止。此时,将有 r 台机组被分配为开机。接着,按照优先顺序将第 1 至第 $r-m$ 台机组分配为开机(不满足启停时间约束的机组仍设为停机),将第 $r+w+1$ 台至第 N 台机组分配为停机(不满足启停时间约束的机组仍设为开机),中间这部分机组则组成一个“优化窗口”,在优化窗口内的机组将通过 BPSO 算法进行寻优。优化窗口的示意图见图 2。

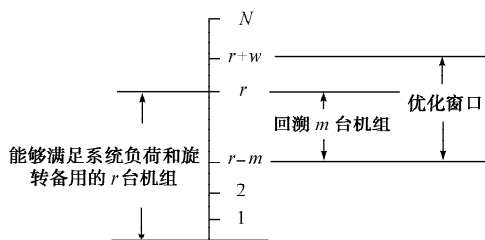


图 2 优化窗口示意

Fig. 2 Illustration of the optimization window

如果优化窗口的尺寸设为 N ,则相当于所有的机组都完全通过 BPSO 算法进行寻优。如果优化窗口设为 0,就完全等同于优先顺序法。前者过于耗时,后者则精度不高。因此,选择合适的优化窗口尺寸很重要,经过经验总结和相关测试后发现,比较适

合的值为 $m+w=0.4N$ 及 $w=2$,此时既能保证优化结果的精度,又能减少优化计算的时间。

在引入了优化窗口的概念后,还加入了一些启发式规则。第 1 条规则是:当在优化窗口内按照优先级顺序进行机组状态决策时,如果运行机组的最大技术出力之和已经能够满足系统负荷和旋转备用的需求,则停止本时段的寻优决策,将剩下的机组全部设置为停机(不满足最小连续运行时间的机组仍然保持运行状态)。第 2 条规则是:假设按照前面的算法,机组 i 在调度时刻 j_1 由运行状态转为停运状态,之后在调度时刻 j_2 重新开机,则这段时间内机组 i 的成本为调度时刻 j_2 的启动费用 S_{ij_2} 。而如果机组 i 在这段时间内保持运行状态,由最优负荷分配的等微增率原则易知该机组应该维持在最小技术出力,此期间内机组 i 的费用应为:

$$\sum_{j=j_1}^{j_2} [a_i + b_i P_i^{\min} + c_i (P_i^{\min})^2]$$

则这段时间内 2 种决策的总费用之差为:

$$\begin{aligned} \Delta F = S_{ij_2} - \sum_{j=j_1}^{j_2} C_i(P_i^{\min}) + \sum_{j=j_1}^{j_2} [C_i(P_{(i-1)j}) - C_{i-1}(P_{(i-1)j} - P_i^{\min})] = \\ S_{ij_2} - \sum_{j=j_1}^{j_2} [a_i + (b_i - b_{i-1})P_i^{\min} + (c_i + c_{i-1})(P_i^{\min})^2 - 2c_{i-1}P_{(i-1)j}P_i^{\min}] \end{aligned}$$

其中,机组 $i-1$ 是在优先队列中排在机组 i 前一位的机组,一般说来, $a_i \gg b_i \gg c_i$,因此上式可化简为 $\Delta F = S_{ij_2} - a_i(j_2 - j_1)$ 。因此,可以定义这样一条启发式规则:当机组 i 的参数满足 $S_{ij} > a_i T_{MD,i}$,而且在 $T_{MD,i}$ 个时段后的系统负荷比当前时段要高时,则机组 i 就不应被停机,而保持开机状态。

3.5 算法步骤

从前面的阐述,可得整个算法的步骤如下:

- 1) 输入系统及机组参数,定义粒子群的大小和迭代总次数。
- 2) 将所有机组按照平均成本微增率减去启动成本之后的值进行排序,值越低则优先级越高。
- 3) 在每个调度时段重复如下步骤:①预先按照优先级分配机组,直到满足系统容量约束为止,由此得到优化窗口的尺寸;②排在优化窗口之前的机组全部设为运行状态(不满足启停时间约束的机组仍设为停机状态),排在优化窗口之后的机组全部设为停机状态(不满足启停时间约束的机组仍设为开机状态);③按照式(8)和式(10)对优化窗口内的机组进行 BPSO 算法的寻优;④当前时刻的全部机组状态确定后,检验其是否满足系统容量约束,如果不满足则按照优先级顺序开机或停机,直到满足约束

为止;⑤往后推算若干时段检验是否满足式(13)所示的约束条件,如果不满足,则转至③。

4)当全部粒子产生之后,计算其适应值,同时保留个体最佳值和全局最佳值。

5)如果已达到最大迭代次数,则转步骤6,否则转步骤3。

6)将全局最佳值作为最终的优化结果输出。

4 仿真实验

本文采用文献[7]中10机、20机、40机、80机和100机几个算例,机组参数和旋转备用取值等见文献[7]。算法采用C语言编制,在Pentium III

800 MHz个人机器上运行,取得了不错的效果。所有算例中粒子个数均设为20个,迭代次数为300次,每个算例都各计算50次。表1列出了采用本文算法得到的10台机组算例的详细优化结果。

总计发电费用为552 245.72美元,启动费用为6 560美元,旋转备用为2 971 MW,机组运行费用为558 805.72美元。表2列出了所有算例优化的性能统计及与文献[7,12]优化的对比。从表中可以看出,本算法最终的优化解明显好于文献[7,12],由于硬件环境的不同,计算时间没有可比性,但本文算法的计算速度仍然是很快的。全部算例均能得到满意解也说明了本算法的鲁棒性和实用性。

表1 本文算法得到的10机算例的最优值
Table 1 Optimized solution of the 10-unit system with the proposed method

时段	发电费用/ 美元	启动费 用/美元	旋转备 用/MW	机组状态	机组出力分配/MW									
					1	2	3	4	5	6	7	8	9	10
1	13 683.13	0	210	1 100 000 000	455	245	0	0	0	0	0	0	0	0
2	14 554.50	0	160	1 100 000 000	455	295	0	0	0	0	0	0	0	0
3	16 301.89	0	60	1 100 000 000	455	395	0	0	0	0	0	0	0	0
4	18 597.67	1 800	122	1 100 100 000	455	455	0	0	40	0	0	0	0	0
5	19 608.54	0	72	1 100 100 000	455	455	0	0	90	0	0	0	0	0
6	21 860.29	1 120	102	1 101 100 000	455	455	0	130	60	0	0	0	0	0
7	23 261.98	1 100	182	1 111 100 000	455	410	130	130	25	0	0	0	0	0
8	24 150.34	0	132	1 111 100 000	455	455	130	130	30	0	0	0	0	0
9	26 588.96	340	112	1 111 110 000	455	455	130	130	110	20	0	0	0	0
10	29 365.95	520	97	1 111 111 000	455	455	130	130	162	43	25	0	0	0
11	31 219.63	60	102	1 111 111 100	455	455	130	130	162	80	25	13	0	0
12	33 205.25	60	107	1 111 111 110	455	455	130	130	162	80	25	53	10	0
13	29 365.95	0	97	1 111 111 000	455	455	130	130	162	43	25	0	0	0
14	26 588.96	0	112	1 111 110 000	455	455	130	130	110	20	0	0	0	0
15	24 318.01	0	82	1 101 110 000	455	455	0	130	140	20	0	0	0	0
16	20 895.88	0	152	1 101 100 000	455	440	0	130	25	0	0	0	0	0
17	20 020.02	0	202	1 101 100 000	455	390	0	130	25	0	0	0	0	0
18	21 860.29	0	102	1 101 100 000	455	455	0	130	60	0	0	0	0	0
19	24 318.01	340	82	1 101 110 000	455	455	0	130	140	20	0	0	0	0
20	30 164.03	1 220	122	1 111 110 110	455	455	130	130	162	48	0	10	10	0
21	26 588.96	0	112	1 111 110 000	455	455	130	130	110	20	0	0	0	0
22	21 879.33	0	70	1 111 000 000	455	385	130	130	0	0	0	0	0	0
23	17 795.28	0	140	1 110 000 000	455	315	130	0	0	0	0	0	0	0
24	16 052.85	0	240	1 110 000 000	455	215	130	0	0	0	0	0	0	0

表2 优化结果性能比较
Table 2 Quality comparison of the optimized solutions

机组数	本文算法					GA ^[7]		文献[12]算法	
	最好解/美元	最差解/美元	平均解/美元	标准偏差	平均计算时间/s	最好解/美元	最差解/美元	最好解/美元	最差解/美元
10	558 806	559 658	559 101	246.98	9	565 825	570 032	565 229	567 251
20	1 114 066	1 116 634	1 115 527	470.79	15	1 126 143	1 132 059		
40	2 230 313	2 230 809	2 230 711	65.16	29	2 251 911	2 259 706		
60	3 347 711	3 348 642	3 348 180	170.51	50	3 376 625	3 384 252		
80	4 460 365	4 460 892	4 460 692	126.48	84	4 504 933	4 510 129		
100	5 577 045	5 577 745	5 577 487	143.39	124	5 627 437	5 637 914		

5 结语

本文提出了一种改进BPSO算法用于求解机组优化组合问题。该算法采取了新的策略来生成粒子,使所有粒子均为可行解,也使整个算法始终在可行解区域内寻优,提高了算法的运算速度和优化性能。在此基础上,吸收了优先顺序法的思想,引入了优化窗口的概念和一些启发式的规则,有效地提高整个算法的计算速度和精度。仿真实验的结果也清楚地表明了本算法的有效性和鲁棒性。

本文对机组的发电成本函数采取二次曲线形式,使得在进行经济负荷分配时计算比较简单。如果成本函数比较复杂,则不仅进行经济负荷分配时会比较复杂,而且也会使启发式规则复杂化,从而影响计算速度。其次,本文在考虑约束条件时并没有考虑机组爬坡速率约束和工作死区等问题。若要考虑这些约束,则在生成粒子时还需要多加一些判断条件以确保机组状态组合不会明显违反机组爬坡速率约束和工作死区约束,或者采用添加罚函数的方法,这将使进行经济负荷分配时的计算变得更复杂。

参考文献

- [1] WOOD A J, WOLLENBERG B F. Power Generation, Operation and Control. New York (NY, USA): John Wiley & Sons, 1984.
- [2] SENJO T, SHIMABUKURO K, UEZATO K et al. A Fast Technique for Unit Commitment Problem by Extended Priority List. IEEE Trans on Power Systems, 2003, 18(2): 882—888.
- [3] SNYDER W L, POWELL H D, RAYBURN J C. Dynamic Programming Approach to Unit Commitment. IEEE Trans on Power Systems, 1987, 2(2): 339—350.
- [4] OUYANG Z, SHAHIDEPOUR S M. An Intelligent Dynamic Programming for Unit Commitment Application. IEEE Trans on Power Systems, 1991, 6(3): 1203—1209.
- [5] ZHUANG F, GALIANA F D. Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation. IEEE Trans on Power Systems, 1988, 3(2): 763—773.

- [6] CHENG C P, LIU C W, LIU C C. Unit Commitment by Lagrangian Relaxation and Genetic Algorithms. IEEE Trans on Power Systems, 2000, 15(2): 707—714.
- [7] KAZARLIS S A, BAKIRTZIS A G, PETRIDIS V. A Genetic Algorithm Solution to the Unit Commitment Problem. IEEE Trans on Power Systems, 1996, 11(1): 83—92.
- [8] SWARUP K S, YAMASHIRO S. Unit Commitment Solution Methodology Using Genetic Algorithm. IEEE Trans on Power Systems, 2002, 17(1): 87—91.
- [9] MANTAWY A H, ABDEL-MAGID Y L, SELIM S Z. A Simulated Annealing Algorithm for Unit Commitment. IEEE Trans on Power Systems, 1998, 13(1): 197—204.
- [10] KENNEDY J, EBERHART R. Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Vol 4. Perth (Australia): 1995. 1942—1948.
- [11] GAING Zhe-lee. Discrete Particle Swarm Optimization Algorithm for Unit Commitment. In: Proceedings of IEEE Power Engineering Society General Meeting, Vol 1. Toronto (Canada): 2003. 418—424.
- [12] 袁晓辉,王乘,袁艳斌,等.一种求解机组组合问题的新型改进粒子群方法.电力系统自动化,2005,29(1):34—38.
YUAN Xiao-hui, WANG Cheng, YUAN Yan-bin et al. A New Enhanced Particle Swarm Optimization Method for Unit Commitment. Automation of Electric Power Systems, 2005, 29(1): 34—38.
- [13] KENNEDY J, EBERHART R. A Discrete Binary Version of the Particle Swarm Algorithm. In: Proceedings of IEEE International Conference on Computational Cybernetics and Simulation, Vol 5. Orlando (FL, USA): 1997. 4104—4108.
- [14] SHI Y, EBERHART R C. Empirical Study of Particle Swarm Optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, Vol 3. Washington (DC, USA): 1999. 1945—1950.

刘涌(1976—),男,博士研究生,研究方向为电力市场分析与智能计算。E-mail: yongliu888@163.com

侯志俭(1942—),男,教授,博士生导师,研究方向为电力市场分析与电力系统运行。

蒋传文(1966—),男,博士,副教授,研究方向为电力系统优化运行与控制、智能计算。

Unit Commitment via an Enhanced Binary Particle Swarm Optimization Algorithm

LIU Yong, HOU Zhi-jian, JIANG Chuan-wen

(Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: Unit commitment is a large scale, discrete and non-linear optimization problem. A solution to unit commitment via an enhanced binary particle swarm optimization (BPSO) algorithm is presented. A new strategy for particles generation is proposed which can make all the particles feasible and narrow the search space within the feasible solutions. Optimization window and heuristic rules are introduced into the methodology to improve the speed and the solution precision. The simulation results clearly show that the proposed method is effective.

Key words: unit commitment; binary particle swarm optimization (BPSO) algorithm; optimization window; heuristic rule