

doi: 10.12052/gdutxb.160161

基于混合纵横交叉算法的机组组合优化

马留洋, 孟安波, 殷 豪

(广东工业大学 自动化学院, 广东 广州 510006)

摘要: 针对机组组合问题的特点, 提出一种混合纵横交叉算法(crisscross optimization, CSO), 分别用离散CSO和连续CSO优化机组的启停状态和机组的功率分配. 进化过程中, 通过横向交叉在不同粒子的相同维进行交叉保证算法的全局收敛能力, 纵向交叉在相同粒子的不同维进行交叉克服局部最优问题, 两者通过一个竞争算子完美地结合, 既加快了收敛速度, 又能保证全局收敛能力. 引入启发式策略修正最小开停机时间、功率平衡等约束条件, 有效地提高了求解的精度. 通过2个算例的仿真以及与其他算法的比较, 验证了该算法的可行性和优越性.

关键词: 机组组合; 纵横交叉; 横向交叉; 纵向交叉; 启发式策略

中图分类号: TM734 文献标志码: A 文章编号: 1007-7162(2017)06-0068-05

Unit Commitment Optimization Based on Hybrid Crisscross Optimization Algorithm

Ma Liu-yang, Meng An-bo, Yin Hao

(School of Automation, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: With regard to the characteristics of the unit commitment problem, a hybrid crisscross optimization algorithm (CSO) is proposed, discrete CSO and continuous CSO are used for the unit status scheduling and power economic dispatch, respectively. In the process of evolution, horizontal crossover performs crossover on the same dimension of different particles to guarantee the global convergence ability, and vertical crossover performs crossover on different dimensions of the same particle to overcome the local optimal convergence, and the perfect combination of the two crossovers by a competitive algorithm not only speeds up the convergence rate but also ensures the ability of the global constraints. The heuristic strategy is introduced to modify the constraint conditions, such as the minimum up and down time and the power balance, which can effectively improve the accuracy of the solution. The feasibility and superiority of the proposed algorithm are verified by the simulation of two examples and the comparison with other algorithms.

Key words: unit commitment; crisscross; horizontal crossover; vertical crossover; heuristic strategy

机组组合问题是电力系统中一个重要的非线性混合整数优化问题, 它的优化目标是在一定的调度周期内合理地安排机组的启停状态以及合理地分配运行机组的机组出力使得总的运行费用最小. 由于要满足功率平衡约束、旋转备用约束、最小开停机时间约束、爬坡约束等一些约束条件, 使得机组组合问题的求解十分困难.

机组组合问题是电力系统中最难求解的问题之一, 它的求解能够带来显著的经济效益, 一直以来是众多学者研究的热点问题. 许多方法已经被用来求

解机组组合问题, 如优先列表法^[1]、动态规划法^[2]、拉格朗日松弛法^[3]等传统方法, 遗传算法^[4-5]、粒子群算法^[6]、差分算法^[7]等人工智能算法以及一些混合算法^[8-12]. 传统方法简单方便, 但往往容易陷入局部最优解, 特别是在求解规模比较大的系统时, 传统方法存在诸多困难, 甚至难以求解. 人工智能算法和混合算法由于其易于实施, 求解复杂问题时的全局收敛能力强, 能够寻找到最优的或者接近最优的解决方案, 近年来已经得到了广泛的认可. 文献^[13]采用离散PSO(Particle Swarm Optimization)分时段优化机组

的启停状态,连续PSO优化机组的负荷分配,并在种群更新时引入临界算子,改进可行解的判别条件.但是分时段优化必然会降低机组组合问题在时间维度上的耦合关系,使得问题的求解精度降低.文献[14]采用改进的离散粒子群算法结合原对偶内点法分别对机组组合问题的整型变量和连续变量进行交替迭代优化求解,但是在处理约束问题时某些时段未能满足功率平衡约束.文献[9]混合遗传算法和差分算法求解机组组合问题,并采用基于优先列表的方法和启发式策略修正约束问题,取得了分别比单独使用遗传算法和差分算法较好的结果,但是随着系统规模的增大,该方法收敛速度慢,并且容易陷入局部最优解.

本文在纵横交叉算法的基础上,提出一种用于求解机组组合问题的混合纵横交叉算法,离散CSO(Crisscross Optimization)和连续CSO在迭代过程中对机组的启停状态和负荷的经济分配进行交替优化求解,结合一种贪婪的选择机制,使得种群朝着全局最优的方向进化,加快了种群的收敛速度.对于功率平衡、最小开停机时间、爬坡限制等一些约束的处理,采用启发式修正方法代替传统的罚函数法,提高和改善求解结果的精度.

1 机组组合问题的数学模型

1.1 目标函数

机组组合问题的目标函数是在一定的调度周期内使总的燃料费用和启动费用之和最小,即

$$\min TC = \sum_{t=1}^T \sum_{i=1}^N [F_i(P_{it}) + SC_{it}(1 - u_{i(t-1)})]u_{it}, \quad (1)$$

式(1)中,TC为系统总运行费用; P_{it} 为机组*i*在时段*t*的出力值; u_{it} 表示机组*i*在时段*t*的状态; N 为机组数; $F_i(P_{it})$ 和 SC_{it} 分别为机组*i*在时段*t*的发电燃料费用和启动费用,其计算公式可分别表示为

$$F_i(P_{it}) = a_i + b_i P_{it} + c_i P_{it}^2, \quad (2)$$

$$SC_{it} = K_i + B_i [1 - \exp(-\text{toff}_i^{t-1} / \tau_i)]. \quad (3)$$

式(2)、(3)中: a_i 、 b_i 、 c_i 分别为机组*i*的燃料费用系数; K_i 、 B_i 、 τ_i 分别为机组*i*的启动耗量系数; toff_i^{t-1} 为机组*i*在*t*时刻前的连续停机时间.

1.2 约束条件

机组组合问题一般需要满足以下约束条件:

(1) 功率平衡约束.

$$\sum_{i=1}^N (P_{it} u_{it}) = \text{Load}_t. \quad (4)$$

式(4)中, Load_t 表示*t*时段的负荷需求.

(2) 旋转备用约束.

$$\sum_{i=1}^N P_i^{\max} u_{it} \geq \text{Load}_t + \text{SR}_t. \quad (5)$$

式(5)中, $P_i^{\max}$ 为机组*i*的最大有功出力; SR_t 为*t*时段的旋转备用需求.

(3) 最小开停机时间约束.

$$\begin{cases} \text{ton}_i^t \geq \text{MUT}_i, \\ \text{toff}_i^t \geq \text{MDT}_i. \end{cases} \quad (6)$$

式(6)中, MUT_i 和 MDT_i 分别为机组*i*的最小连续开机和停机时间; ton_i^t 表示机组*i*在*t*时段的连续开机时间.

(4) 机组出力约束.

$$u_{it} P_i^{\min} \leq P_{it} \leq u_{it} P_i^{\max}. \quad (7)$$

式(7)中, $P_i^{\min}$ 为机组*i*的最小有功出力.

(5) 机组爬坡约束.

$$\begin{cases} P_{it} - P_{i,t-1} \leq \text{UR}_i, \\ P_{i,t-1} - P_{it} \leq \text{DR}_i. \end{cases} \quad (8)$$

式(8)中, UR_i 为机组出力增加的速度上限, DR_i 为机组出力的减少的速度下限.

考虑机组的爬坡约束限制时,机组的出力约束需要被修改为

$$\begin{aligned} \max(P_i^{\min}, P_{i,t-1} - \text{DR}_i) &\leq P_{it} \leq \\ \min(P_i^{\max}, P_{i,t-1} + \text{UR}_i). \end{aligned} \quad (9)$$

2 混合纵横交叉算法

纵横交叉算法^[15]是最近发展起来的新型群体智能搜索算法.它采用一种双向搜索机制,即横向交叉和纵向交叉,结合一个简单的竞争机制,形成CSO的基本搜索行为.在每一次迭代过程中,CSO从两个不同的方向对种群进行交叉产生子代,这些子代被称为中庸解.随后,这些新产生的中庸解通过一个贪婪的选择机制和父代进行比较,只有比父代适应度更好的解才能够存活下来,否则就被淘汰.这些在竞争中存活下来的解被称为占优解,作为下一次交叉的父代.很显然,在种群迭代过程中,CSO始终保持个体的最优解,这样的竞争机制无疑会加快求解的收敛速度.

(1) 横向交叉.

横向交叉之前需要对种群中的粒子两两不重复

配对,然后对配对粒子的相同维进行算术交叉.假设对粒子 $X(i)$ 和 $X(j)$ 的第 d 维交叉,公式为

$$\begin{cases} MS_{id} = r_1 X_{id} + (1 - r_1) X_{jd} + c_1 (X_{id} - X_{jd}), \\ MS_{jd} = r_2 X_{jd} + (1 - r_2) X_{id} + c_2 (X_{jd} - X_{id}). \end{cases} \quad (10)$$

式(10)中, r_1, r_2 为 $[0, 1]$ 上的随机数; c_1, c_2 为 $[-1, 1]$ 之间的随机数; X_{id} 和 X_{jd} 分别为粒子 $X(i)$ 和 $X(j)$ 的第 d 维; MS_{id} 和 MS_{jd} 分别为横向交叉后的第 d 维子代.

由式(10)可以观察到横向交叉由两个部分组成:第1部分是内部信息的交换,以较大的概率在父代粒子形成的超立方体内进行搜索;第2部分是外部信息的探索,通过在父代粒子之间施加一个扩展系数,以较小的概率在超立方体的外缘进行搜索.这种搜索机制可以有效地减少搜索盲点,提高CSO的全局搜索能力.

(2) 纵向交叉.

与横向交叉不同的是,纵向交叉是在同一个粒子不同维的交叉,假设对粒子 $X(i)$ 的第 d_1 和 d_2 维执行纵向交叉,公式为

$$\begin{aligned} MS_{i,d_1} &= r \cdot X_{i,d_1} + (1 - r) \cdot X_{i,d_2} \quad i \in (1, M), \\ d_1, d_2 &\in (1, D). \end{aligned} \quad (11)$$

式(11)中, M 为粒子的个数; D 为粒子的维数; r 为 $[0, 1]$ 上的均匀分布随机数; MS_{i,d_1} 为粒子 $X(i)$ 的第 d_1 维子代.

由于不同维的上下限往往不同,所以纵向交叉之前需要对粒子的各维进行归一化处理,纵向交叉结束后还需要对所产生的中庸解进行反归一化处理.一般算法在迭代过程中某些维容易陷入局部最优而停止更新,但不同的维一般都有差异.纵横交叉算法正是观察到这种现象,使用纵向交叉在不同维进行算术交叉,可以有效地帮组陷入局部最优的维跳出困境,而且每次交叉只产生一个子代,在保持种群多样性的同时而不破坏有可能比较好的维.相比于其他搜索算法,纵向交叉是纵横交叉算法的一个明显的特征.

根据前面的阐述可知,CSO是连续线性的群体智能搜索算法,不利于求解机组状态优化的离散问题,因此有必要对CSO进行二进制化.本文采用Sigmoid函数对交叉后的中庸解进行转换,公式为

$$\begin{cases} B_{id} = 1 \text{ if } R < S(MS_{id}), \\ B_{id} = 0 \text{ if } R \geq S(MS_{id}). \end{cases} \quad (12)$$

式(12)中, R 为 $[0, 1]$ 上的均匀分布随机数; B_{id} 为二进制化结果; $S(x)$ 为Sigmoid函数,可表示为式(13)

$$S(x) = 1/[1 + \exp(-x)]. \quad (13)$$

3 混合纵横交叉算法用于机组组合问题

3.1 约束条件的处理

在机组组合问题中,约束条件的处理是一个难点,一般采用的是罚函数法,给不满足约束的粒子设定一个惩罚系数,使其朝着可行解的方向进化.这种方法使得收敛速度变慢,而且收敛得到的解很难绝对的满足约束条件,使得求解精度降低.本文采用启发式的策略代替罚函数法修正约束条件,提高求解的速度和精度.

(1) 最小开停机时间约束.离散CSO在每次交叉之后产生的解一般不满足最小开停机时间约束,所以每次离散交叉之后都需要检查机组状态是否满足最小开停机时间约束.对于不满足约束的解,一般有3个调整的方向,即:调整违背约束时段之前的机组状态、调整违背约束时段之后的状态、调整违背约束时段的机组状态.本文根据违背约束时段的开停机时间与最小开停机时间的比值确定机组状态的调整方向,这一概率选择机制可以产生更多的机组组合策略,增加种群的多样性,保证算法的全局收敛能力.以违背最小开停机时间约束为例,不同方向的调整方法如式(14)所示.

$$\begin{cases} \text{if } \text{rand1} \leq \frac{\text{ton}_{it}}{\text{MUT}_i} \& \text{rand2} \leq 0.5 \\ \& \text{tof}_{i,t-(\text{MUT}_i-\text{ton}_i)} > \text{MDT}_i, \text{向前调整;} \\ \text{if } \text{rand1} \leq \frac{\text{ton}_{it}}{\text{MUT}_i} \& \text{rand2} > 0.5, \text{向后调整;} \\ \text{if } \text{rand1} > \frac{\text{ton}_{it}}{\text{MUT}_i}, \text{调整违背约束的时段.} \end{cases} \quad (14)$$

(2) 旋转备用约束.根据“上大压小”的原则,优先开启大容量的机组满足选择备用约束,之后还要检查最小开停机时间约束是否满足.当这两项约束都满足之后,检查是否有冗余的机组.冗余的机组会增加运行的成本,在满足约束的情况下尽可能地关闭冗余机组.

(3) 功率平衡约束.根据式(4),功率的不平衡量可以表示为

$$\Delta P = \text{Load}_t - \sum_{i=1}^N (P_{it} u_{it}). \quad (15)$$

这些不平衡量需要分配到运行的机组出力上以满足功率平衡约束.本文将不平衡量平均分成对应时段所开机组数目的等份,然后随机地分配到一些机组之上,使得功率平衡能够很好地满足.具体流程如图1所示.

问题时具有良好的寻优能力和鲁棒性.

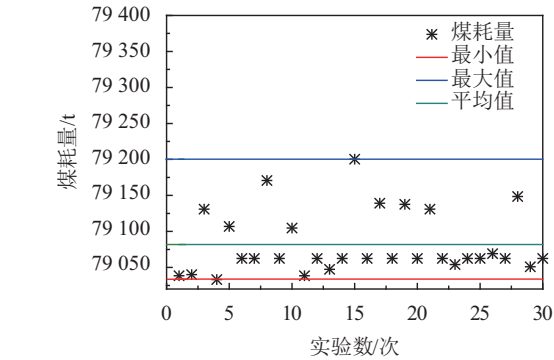


图2 实验结果分布图

Fig.2 Distribution chart of experiment results

为了进一步验证本文算法的有效性,将本文的煤耗量与其他算法的结果相比较,统计结果如表1所示.

表1 10机系统结果比较¹⁾

Tab.1 Comparison of results for 10-unit system

优化方法	机组煤耗量/t
Hopfield模拟退火混合算法 ^[16]	79 114.6
双重粒子群算法 ^[13]	79 665.8
改进离散粒子群算法 ^[14]	78 965.0*
遗传算法 ^[5]	79 807.0
遗传粒子群混合算法 ^[8]	78 862.7*
混合纵横交叉算法	79 033.7

1) *表示结果有偏差

本文所提混合纵横交叉算法总的煤耗量为79 033.2 t,其中启动费用为267.3 t. 系统优化过后的机组出力表如表2所示,出力为0表示机组停机,不为0表示机组运行. 从表1可以计算出,与Hopfield模拟退火混合算法、双重粒子群算法、遗传算法的煤耗量相比,本文算法的煤耗量分别节省81.4 t、632.6 t、773.8 t,分别降低了0.1%、0.79%、0.97%. 虽然本文算法煤耗量略高于文献[8]和[14]的所得结果,但是本文采用启发式的策略修正约束条件,使得约束条件能够很好的满足. 从文献[14]给出的机组出力分布表可以看出,系统在第17时段的机组出力比负荷需求少17 MW,没能满足机组的功率平衡约束. 文献[8]采用罚函数法处理约束条件,很难得到质量很高的解,而且根据文献中所提供的机组出力分布计算得到的煤耗量为79 307.3 t,比文献中给出的煤耗量结果要高. 从表2可以看出,本文的优化结果符合“上大压小”的原则,优先开启大容量的机组,这样既可以节约煤耗成本,又符合节能减排的目的和实际工程需求.

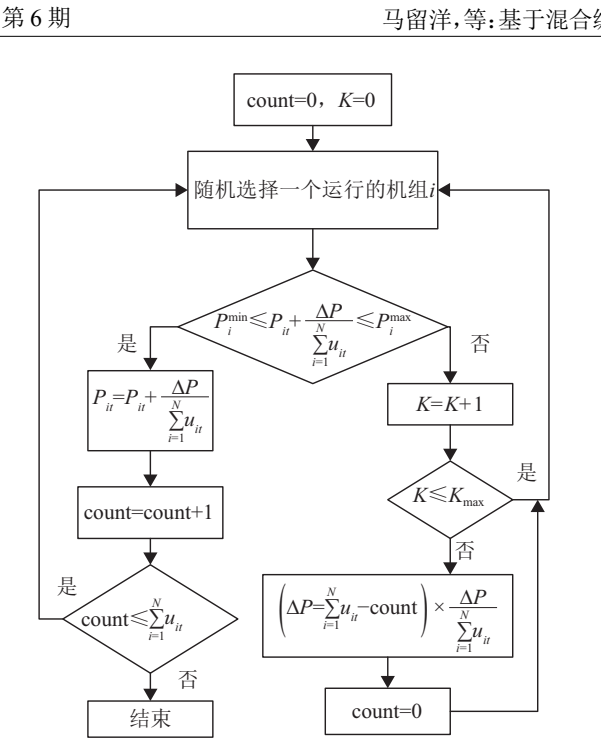


图1 功率平衡修正流程图

Fig.1 Flow chart of power balance correction

3.2 算法的具体流程

结合机组组合问题的数学模型和混合纵横交叉算法的原理,混合纵横交叉算法应用到机组组合问题主要包括以下步骤:

- (1) 设置参数,初始化种群;
- (2) 修正约束,计算粒子的适应度值;
- (3) 离散CSO和连续CSO分别对机组状态变量和机组出力变量交叉产生子代;
- (4) 每一次交叉之后修正子代的约束并计算适应度值与父代进行比较,适应度好的解保留到占优解里作为下一次交叉的父代;
- (5) 判断是否达到最大迭代次数,如果是,输出最优解,否则转到步骤(3).

4 算例分析

本文首先对10机24 h的系统进行机组组合优化分析,机组参数和负荷数据参见文献[3]. 种群规模为20,迭代次数为100次.

由于搜索算法的随机性,为了减少统计误差,本文对测试系统进行30次独立的实验,结果分布如图2所示. 30次独立重复实验的最小值、平均值和最大值分别为79 033.7 t、79 081.8 t和79 200.2 t. 最大值和最小值相差为166.5 t,相对于最小值的偏差仅为0.21%,而且由图2可以看出,大部分解都分布在平均值以下,证明了混合纵横交叉算法在求解机组组合

表2 10机系统机组状态及功率分配
Tab.2 Unit status and power dispatch of 10-unit system

时段	机组有功出力/MW						
	1~4	5	6	7	8	9	10
1	0.0	150.0	184.1	213.2	433.7	512.3	506.7
2	0.0	150.0	181.5	210.8	429.0	506.9	501.7
3	0.0	150.0	176.3	206.1	419.8	496.2	491.6
4	0.0	148.2	171.3	201.6	411.0	485.9	481.9
5	0.0	139.8	164.6	195.5	399.1	472.1	468.9
6	0.0	144.0	167.9	198.6	405.1	479.0	475.4
7	0.0	150.0	0.0	213.8	434.8	513.6	507.9
8	0.0	143.7	0.0	198.3	404.6	478.5	474.9
9	0.0	114.0	0.0	176.6	362.1	429	428.3
10	0.0	0.0	0.0	178.5	365.9	433.3	432.4
11	0.0	0.0	0.0	166.3	342.0	405.5	406.2
12	0.0	0.0	0.0	150.0	310.2	368.4	371.4
13	0.0	0.0	0.0	150.0	310.2	368.4	371.4
14	0.0	0.0	0.0	144.6	299.6	356.1	359.8
15	0.0	0.0	0.0	141.8	294.3	349.9	353.9
16	0.0	0.0	0.0	144.6	299.6	356.1	359.8
17	0.0	0.0	0.0	158.1	326.1	387	388.8
18	0.0	0.0	0.0	174.4	357.9	424.0	423.7
19	0.0	0.0	0.0	198.8	405.6	479.6	476.0
20	0.0	0.0	0.0	218.4	443.8	520	517.8
21	0.0	150.0	0.0	213.8	434.8	513.6	507.9
22	0.0	148.2	171.3	201.6	411.0	485.9	481.9
23	0.0	150.0	177.6	207.3	422.1	498.9	494.1
24	0.0	150.0	182.8	212.0	431.4	509.6	504.2

另外,为了验证本文算法在处理大规模系统时的全局收敛能力,对40机24 h的系统进行机组组合优化分析,机组参数和负荷数据参见文献[11]。随着系统规模的增大,机组的启动状态数目会呈指数增加,一般算法很容易陷入局部最优,使得求解更加困难。根据研究发现^[15],大部分算法在陷入局部最优时都是某些维停滞了更新。本文算法的纵向交叉是对不同的维进行交叉,能够很好地帮组停滞的维跳出困境,然后通过横向交叉将信息扩散至整个种群。不同算法的收敛结果如表3所示。从表3可以看出,本文算法的优化结果不仅优于改进的算法,而且还优于混合的算法,证明了本文算法在求解大规模机组组合系统时良好的全局寻优能力。

表3 40机系统结果比较
Tab.3 Comparison of results for 40-unit system

优化方法	优化结果/\$
差分算法 ^[7]	2 246 274
混合遗传差分算法 ^[9]	2 243 522
改进离散粒子群算法 ^[17]	2 248 581
改进量子进化算法 ^[18]	2 245 151
混合纵横交叉算法	2 242 957

5 结论

本文提出了一种新型的混合纵横交叉算法求解机组组合问题,该算法分别采用离散CSO和连续CSO处理机组状态更新和负荷出力更新。进化过程中横向交叉和纵向交叉交替进行,不仅加快了算法的收敛速度,而且可以保证算法的全局收敛能力。在处理约束问题时,采用启发式的策略修正约束,避免了使用罚函数法处理约束时机组出力小于负荷需求的现象。通过2个算例的仿真可以看出,本文算法能够很好地用于机组组合问题,在机组数目增多时,能够克服维数灾难,取得满意的收敛结果。

参考文献:

[1] SENJYU T, SHIMABUKURO K, UEZATO K, *et al.* A fast technique for unit commitment problem by extended priority list [J]. IEEE Transactions on Power Systems, 2003, 18(2): 882-888.

[2] 白晓民, 于尔铿. 用动态规划法进行电力系统机组组合最优化[J]. 中国电机工程学报, 1984, 4(1): 13-21.

BAI X M, YU E K. Optimization for unit commitment of electric power system by dynamic programming [J]. Proceedings of the CSEE, 1984, 4(1): 13-21.

[3] 韩学山, 柳焯. 考虑发电机组输出功率速度限制的最优机组组合[J]. 电网技术, 1994, 18(06): 11-16.

HAN X S, LIU Z. Optimal unit commitment considering unit's ramp-rate limits [J]. Power System Technology, 1994, 18(06): 11-16.

[4] DATTA D. Unit commitment problem with ramp rate constraint using a binary-real-coded genetic algorithm [J]. Applied Soft Computing, 2013, 13(9): 3873-3883.

[5] 蔡超豪, 蔡元宇. 机组优化组合的遗传算法[J]. 电网技术, 1997, 21(1): 44-47.

CAI C H, CAI Y Y. Optimization of units commitment by genetic algorithm [J]. Power System Technology, 1997, 21(1): 44-47.

[6] 周俊. 机会约束规划下含风电场的机组组合优化[D]. 广州: 广东工业大学自动化学院, 2014.

[7] DATTA D, DUTTA S. A binary-real-coded differential evolution for unit commitment problem [J]. International Journal of Electrical Power & Energy Systems, 2012, 42(1): 517-524.

[8] 张炯, 刘天琪, 苏鹏, 等. 基于遗传粒子群混合算法的机组组合优化[J]. 电力系统保护与控制, 2009, 37(9): 25-29.

ZHANG J, LIU Q T, SHU P, *et al.* Unit commitment optimization based on genetic algorithm and particle swarm optimization hybrid algorithm [J]. Power System Protection and Control, 2009, 37(9): 25-29.

[9] TRIVEDI A, SRINIVASAN D, BISWAS S, *et al.* A genetic algorithm-differential evolution based hybrid framework: Case study on unit commitment scheduling problem [J]. Information Sciences, 2016, 354: 275-300.