
算法 1: 出力分配算法

Input: $S, p, start, r, c$

矩阵 S 表示各机组的段容量矩阵

矩阵 p 表示各机组的段价矩阵

矩阵 $start$ 表示各机组的初始出力矩阵

矩阵 r 表示各机组的各机组的爬坡速率矩阵

c 表示下一个时段预报的负荷需求

Output: D, ans

矩阵 D 表示下一个时段各机组的出力分配预案中各机组出力的段位

ans 表示清算价

1 **begin**

2 初始化数组 all_p ,并把矩阵 p 的元素全放入

3 $sort(all_p)$

4 初始化前缀和矩阵 pre_S

5 **for** $i \leftarrow 1$ **to** 8 **do**

6 $sum = 0$

7 **for** $j \leftarrow 1$ **to** 10 **do**

8 $sum = sum + S_{ij}$

9 $pre_{S_{ij}} = sum$

10 **end**

11 **end**

12 初始化机组初始段位矩阵 $index$

13 **for** $i \leftarrow all_p.size()$ **to** 1 **do**

14 $sum = 0$

15 **for** $j \leftarrow 1$ **to** 8 **do**

16 **while** $start_j - r_j * 15 > pre_{S_{j,index_j}}$ **do**

17 $index_j = index_j + 1$

18 **end**

19 **for** $k \leftarrow index_k$ **to** 10 **do**

20 **if** $p_{jk} \geq all_{p_i}$ **then**

21 $index = k - 1$

22 **break**

23 **end**

24 **end**

25 $sum = sum + \min(pre_{S_{j,index_k}}, start_j + r_j * 15)$

26 **end**

27 **if** $sum \geq c$ **then**

28 $ans = all_{p_i}$

29 $D = index$

30 **end**

31 **end**

32 **return** D, ans

33 **end**
