**5 数值实验---算法在电力系统经济调度中应用**

本部分讨论本文所建算法**在电力系统经济调度中应用，从而检验算法**的数值实效性。本数值实验平台为MATLAB R2016a 与OPTI 版本 2.28，IPOPT 3.12.9，软件运行环境为Windows 7(64bite), Intel(R) Core(TM) i7 -4790 CPU 3.60GHz RAM 8.

**5.1 电力系统经济调度模型及分块**

电力系统经济调度问题是在发电机组启停状态确定情况下，满足发电机物理与系统约束条件下, 寻求供电系统发电总费用最小的电力调度(发电)方案。参考文献[31,32], **电力系统经济调度数学模型通常可描述如下**.

1. 目标函数：

其中为发电机在时段的出力，,,和为发电机费用函数系数，*T*为优化时段数，*N*为电力系统中发电机个数.

2) 系统功率平衡约束：

1. 发电机爬坡约束：

，

其中,分别为发电机的上下爬坡速率限制, 并默认 =.

1. 发电机出力上下限约束：

，

其中,分别为发电机的最小、最大出力。

显然，以上电力系统经济调度模型~为光滑非凸非线性规划（NLP）问题， 其变量数为，约束数为。为了便于使用本文算法求解经济调度模型~，引入辅助变量，将发电机爬坡约束由不等约束转换为等式约束。则电力系统经济调度模型可等价表述为形如(2.1)的线性箱子约束约束优化：

(5.5)

考虑到电力系统经济调度问题工程特征, 我们按机组数将优化问题(5.5)均衡地分为两分块. 取, 设，,，，上述优化问题(5.5)可以表述为如下形式：

其中，,,,,,,,; = 为爬坡约束与功率平衡约束矩阵， 其矩阵结构如图5-1所示。

显然，模型(5.6)是一个与问题(2.1)同结构的两分块中大规模优化问题，于是可用包括本文算法在内有相关算法尝试求解, 以验证和比较算法的有效性.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | |  | | | | | |  |  |
| 爬坡约束 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  | -1 | 1 |  |  |  |  |  |  |  |  |  |
|  |  | 1 |  | -1 | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  | -1 | 1 |  |  |  |  |  |
|  |  |  |  |  |  | 1 |  | -1 | 1 |  |  |  |
|  |  |  |  |  |  |  |  | 1 |  | -1 | 1 |  |
| 功率平衡 | 1 |  |  |  |  |  | 1 |  |  |  |  |  |  |
|  |  | 1 |  |  |  |  |  | 1 |  |  |  |  |
|  |  |  |  | 1 |  |  |  |  |  | 1 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | | | | | | | | | | | |  |

表5-1 2台发电机3时段经济调度约束系数矩阵结构图

**5.2 数值报告与数值效果**

下面基于电力系统经济调试模型中的5机组问题和20个中大规模机组问题,测试本文算法B(记为ADMM-SQP-B)和C(简记为ADMM-SQP-C)的数值算法效果,并与著名的OPTI求解器[33]、传统SQP算法、BADMM算法[34]等相关算法进行比较。

分三部分报告与分析。第一部分以5机组算例测试本文算法ADMM-SQP-B与ADMM-SQP-C。第二部分将算法ADMM-SQP-B和ADMM-SQP-C的求解效果与直接调用OPTI求解器相比较。第三部分将算法ADMM-SQP-B和ADMM-SQP-C与传统SQP算法和BADMM算法进行比较。

数值实验数学模型为电力系统经济调度问题(5.1)-(5.4),其标准化两分块模型为（5.6），各具体算例数据来自文献[31]。按电力系统特征与惯例，在所有测试问题中，均取优化时段*.*

数值实验中设置统一的参数值，即，，,,实验终止准则,.各算例的初始迭代点为：

, ,.

此外，QP子问题（2.4）及（2.5）中的二次系数矩阵分别取为相应目标函数的Hessian阵，即

.

由于电力系统经济调度模型目标函数中各系数 均非负，故 在出力约束 内均是半正定的.另外，由图5-1可见矩阵 故算法收敛性假设条件2.1恒成立.

1. **5机组实例仿真结果**

此部分利用文献[31]的5机组算例（N=5）对本文算法ADMM-SQP-B与ADMM-SQP-C进行测试和比较。实验结果如图5-1，5-2,5-3及表5-2所示，从多维度报告和分析两者的数值性能。图5-1表示算法在迭代过程中与最优解的均方误差。图5-2，5-3分别表示算法迭代中逼近程度 和目标函数值 表5-2报告迭代次数及计算时间.

从图5-1，5-2，5-3及表5-2可见，ADMM-SQP-B与ADMM-SQP-C两个算法在5机组实例测试是有效的，且后者略占优势。然后，结合后面20个中大规模问题的测试报告表5-4可见，两者的数值效果几无差异。当然，这仅对电力系统经济调度模型(5.1)-(5.4)而言。对于其他类型问题而言，两者的优劣有待进一步实验。

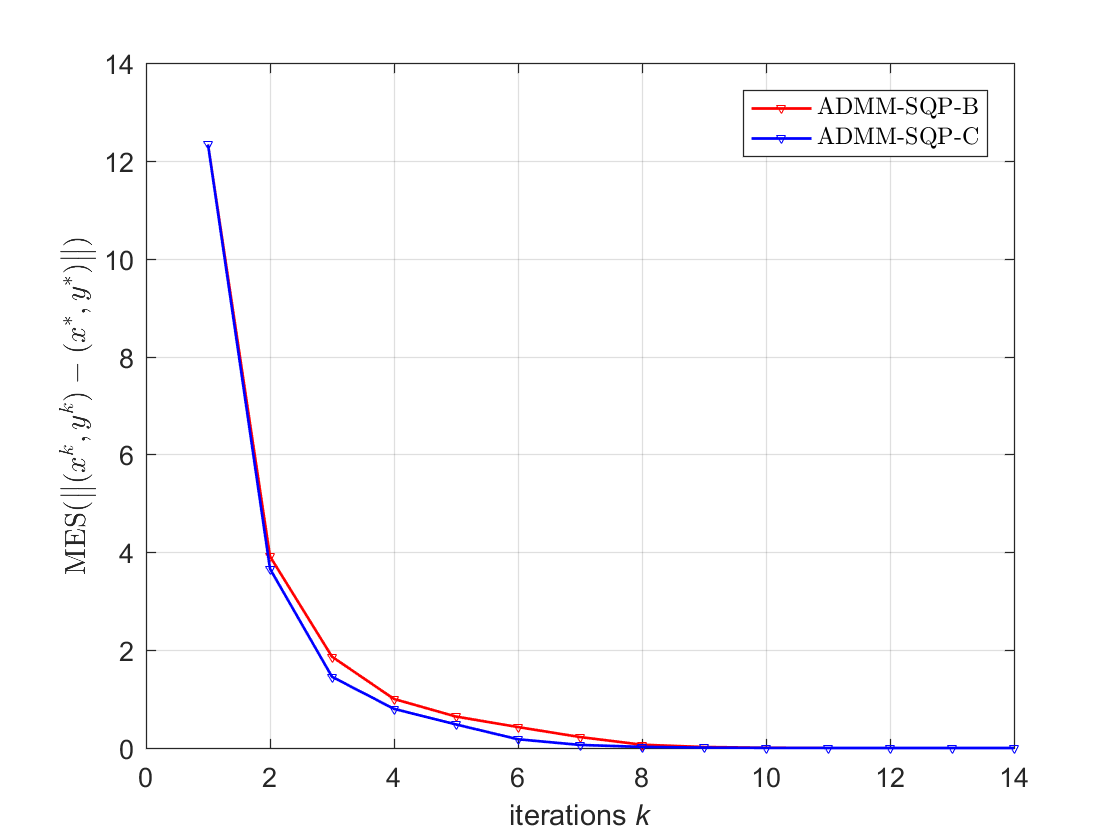


图5-1 ADMM-SQP-B与ADMM-SQP-C算法性能比较一(迭代点的逼近)

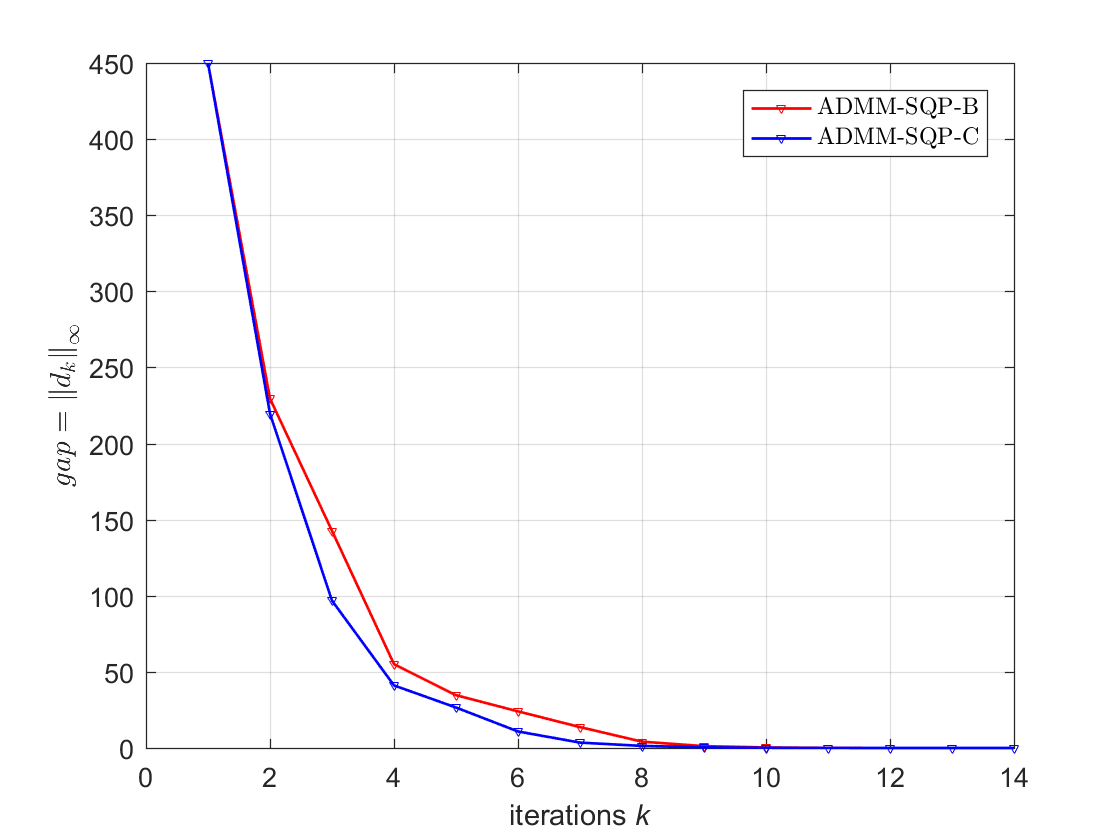


图5-2 ADMM-SQP-B与ADMM-SQP-C算法性能比较二(逼近程度

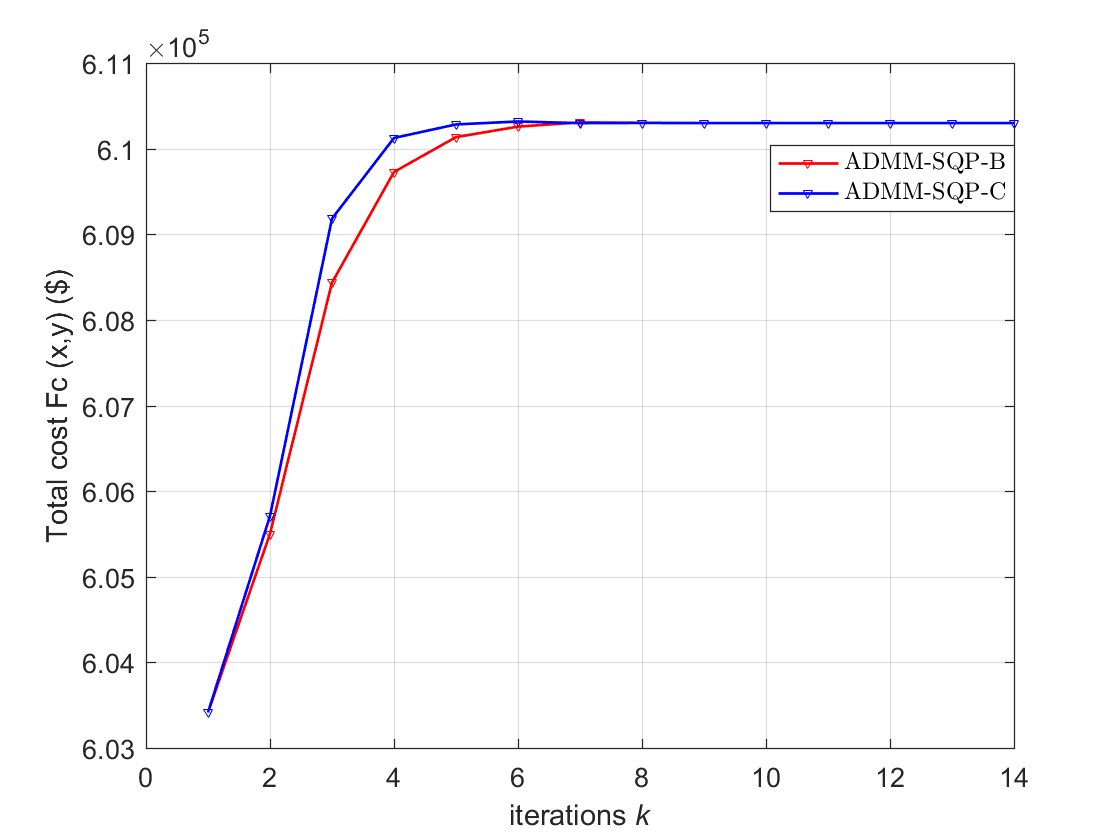


图5-3 ADMM-SQP-B与ADMM-SQP-C算法性能比较三(目标值的逼近

|  |  |  |
| --- | --- | --- |
| 目 次  算 法 | 迭代次数 | CPU计算时间（秒） |
| ADMM-SQP-B | 14 | 3.03 |
| ADMM-SQP-C | 14 | 2.83 |

表5-2 ADMM-SQP-B与ADMM-SQP-C算法性能比较四(迭代次数及计算时间

1. **中大规模机组实例仿真结果—与OPTI比较**

如文献[31]中所示，通过复制5机组数据以创建20个中大规模的互不相同测试实例，每个实例的机组个数如表5-3所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Unit | | | | | Total  Units N | No. | Unit | | | | | Total  Units N |
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 2 | 2 | 10 | 11 | 20 | 24 | 27 | 20 | 19 | 110 |
| 2 | 3 | 3 | 3 | 3 | 3 | 15 | 12 | 22 | 26 | 29 | 22 | 21 | 120 |
| 3 | 4 | 4 | 4 | 4 | 4 | 20 | 13 | 26 | 30 | 30 | 22 | 22 | 130 |
| 4 | 5 | 6 | 7 | 7 | 5 | 30 | 14 | 30 | 33 | 32 | 25 | 30 | 150 |
| 5 | 5 | 10 | 10 | 5 | 10 | 40 | 15 | 34 | 37 | 36 | 29 | 34 | 170 |
| 6 | 8 | 11 | 12 | 9 | 10 | 50 | 16 | 36 | 39 | 38 | 30 | 37 | 180 |
| 7 | 10 | 14 | 16 | 15 | 15 | 70 | 17 | 40 | 44 | 41 | 34 | 41 | 200 |
| 8 | 13 | 18 | 18 | 13 | 18 | 80 | 18 | 44 | 48 | 45 | 38 | 45 | 220 |
| 9 | 12 | 20 | 25 | 20 | 13 | 90 | 19 | 48 | 52 | 48 | 40 | 52 | 240 |
| 10 | 18 | 22 | 25 | 18 | 17 | 100 | 20 | 50 | 54 | 50 | 42 | 54 | 250 |

表 5-3 单个实例机组数

为验证本文算法在以上中大规模电力系统经济调度中的高效性，在同一实验平台及同一软件运行环境下, 分别用本文两个算法和OPTI求解器(直接作用于原模型(5.1)-(5.4))对上述20个实例进行求解. 数值结果如表5-4 所示, 其中，和列分别表示相应算法求得的(近似)最优值(单位为, *C*time 列表示CPU计算时间(秒)，*N*it列为算法迭代次数与列分别表示算法ADMM-SQP-B及ADMM-SQP-C与OPTI求解器获得的最优值的相对误差([**Relative**](javascript:;) [**Error**](javascript:;))，即

.

**效果分析:**

1. 从计算时间成本看, 本文的两个ADMM-SQP算法要远优于OPTI. 首先, 从表5-4可知, 随着问题规模增加, OPTI的计算时间快速增加. 特别地, 当发电机组数超过80时, OPTI的计算时间均超过1000秒. 而本文的ADMM-SQP算法则从20秒左右缓慢增加,直至机组数达到250时,计算时间依旧不超过225秒. 其次, 从20个实例总的计算时间看, 两个ADMM-SQP算法的计算时间均仅为OPTI的9%左右.
2. 在大量节省计算时间成本的前提下,本文ADMM-SQP算法的求解精度(近似最优值)依旧良好, 其与OPTI的相对误差为万分之一到万分之二. 导致这些细小误差的原因有二. 一是传统ADMM类算法求解精度不高是其固有缺陷. 二是算法迭代中MTALB计算误差或子问题求解精度误差也会加剧这种误差。
3. 随着问题规模的增加, ADMM-SQP算法迭代次数是相当稳定的，这充分体现算法良好的鲁棒性。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | NLP-OPTI | | ADMM-SQP-B | | | | ADMM-SQP-C | | | |
| Total cost | *C*time (s) | Total cost |  | *N*it | *C*time(s) | Total cost |  | *N*it | *C*time(s) |
| 1 | 1243485.20 | 1.76 | 1243611.17 | 0.0101% | 12 | 2.48 | 1243611.28 | 0.0101% | 12 | 2.47 |
| 2 | 1833617.31 | 3.33 | 1833808.32 | 0.0104% | 12 | 2.66 | 1833808.47 | 0.0104% | 12 | 2.66 |
| 3 | 2444823.08 | 8.18 | 2445089.70 | 0.0109% | 12 | 2.94 | 2445089.93 | 0.0109% | 12 | 2.93 |
| 4 | 3650316.85 | 27.36 | 3650940.12 | 0.0171% | 13 | 3.94 | 3650940.12 | 0.0171% | 13 | 3.92 |
| 5 | 5083735.09 | 134.20 | 5084391.63 | 0.0129% | 12 | 4.69 | 5084392.09 | 0.0129% | 12 | 4.67 |
| 6 | 6192152.16 | 98.26 | 6193185.51 | 0.0167% | 13 | 6.86 | 6193186.04 | 0.0167% | 13 | 6.83 |
| 7 | 8636967.25 | 111.54 | 8638532.86 | 0.0181% | 13 | 17.11 | 8638533.58 | 0.0181% | 13 | 17.90 |
| 8 | 9973328.82 | 1004.90 | 9974943.64 | 0.0162% | 13 | 20.99 | 9974944.50 | 0.0162% | 13 | 19.74 |
| 9 | 11035233.41 | 1005.22 | 11037322.45 | 0.0189% | 13 | 25.50 | 11037323.38 | 0.0189% | 13 | 25.53 |
| 10 | 12291433.08 | 1006.47 | 12293741.41 | 0.0188% | 13 | 30.03 | 12293742.46 | 0.0188% | 13 | 29.30 |
| 11 | 13513839.10 | 1004.68 | 13516385.17 | 0.0188% | 13 | 36.58 | 13516386.32 | 0.0188% | 13 | 36.12 |
| 12 | 14736246.11 | 875.78 | 14739033.12 | 0.0189% | 13 | 43.06 | 14739034.37 | 0.0189% | 13 | 42.97 |
| 13 | 15975567.93 | 893.45 | 15978719.24 | 0.0197% | 13 | 50.93 | 15978720.59 | 0.0197% | 13 | 50.46 |
| 14 | 18492204.68 | 1014.45 | 18495641.46 | 0.0186% | 13 | 65.53 | 18495643.01 | 0.0186% | 13 | 64.40 |
| 15 | 20937025.33 | 1017.75 | 20940974.18 | 0.0189% | 13 | 90.34 | 20940975.94 | 0.0189% | 13 | 89.85 |
| 16 | 22197414.84 | 1020.97 | 22201586.65 | 0.0188% | 13 | 99.25 | 22201588.51 | 0.0188% | 13 | 99.08 |
| 17 | 24659160.06 | 1024.30 | 24663850.50 | 0.0190% | 13 | 126.55 | 24663852.57 | 0.0190% | 13 | 125.96 |
| 18 | 27103981.13 | 1030.96 | 27109222.63 | 0.0193% | 13 | 165.28 | 27109224.91 | 0.0193% | 13 | 162.26 |
| 19 | 29641688.17 | 1026.52 | 29647408.79 | 0.0193% | 13 | 202.02 | 29647411.31 | 0.0193% | 13 | 201.57 |
| 20 | 30864098.29 | 1023.08 | 30870121.91 | 0.0195% | 13 | 223.97 | 30870124.54 | 0.0195% | 13 | 223.85 |
| Sum |  | 13333.16 |  |  | 256 | 1220.71 |  |  | 256 | 1212.47 |

表5-4 ADMM-SQP-B, ADMM-SQP-C及OPTI求解中大规模实例数值结果

**c. 与同类算法纵向比较**

由于本文所提的两个ADMM-SQP算法思想源于传统SQP算法和ADMM算法, 对算法数值效果进行纵向比较是必要的. 基于对表5-3的20个机组调度实例的求解，下面我们进一步比较本文的两个ADMM-SQP算法与传统SQP算法(直接求解未分块的QP子问题）及文献[34]的BADMM算法. 在本文的实验平台及软件运行环境下, SQP算法(作用于等价模型(5.5))及BADMM算法(作用于原模型(5.1)-(5.4))求解表5-3的20个实例的结果如下表5-5所示。

**效果分析:** 基于表5-4和表5-5的数值报告，从计算时间、与OPT的相对误差及迭代次数等3个维度综合分析，所比较的几个算法的数值效果由优到劣排列依次为：本文的两个ADMM-SQP算法、POTI求解器、SQP算法、BADMM算法。尤其是BADMM算法，其计算时间及相对误差都有些难以接受。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | SQP | | | | BADMM | | | |
| Total cost | *C*time (s) | *N*it |  | Total cost | *C*time (s) | *N*it |  |
| 1 | 1243621.12 | 1.91 | 9 | 0.0109% | 1246279.53 | 254.97 | 14 | 0.2247% |
| 2 | 1833917.14 | 2.18 | 9 | 0.0164% | 1837568.26 | 233.53 | 11 | 0.2155% |
| 3 | 2445257.57 | 2.59 | 9 | 0.0178% | 2449871.75 | 391.13 | 10 | 0.2065% |
| 4 | 3651003.34 | 4.41 | 9 | 0.0188% | 3658906.61 | 144.10 | 10 | 0.2353% |
| 5 | 5084413.33 | 8.33 | 9 | 0.0133% | 5096007.23 | 203.09 | 10 | 0.2414% |
| 6 | 6193214.44 | 11.77 | 9 | 0.0172% | 6207111.21 | 359.62 | 8 | 0.2416% |
| 7 | 8638525.13 | 20.75 | 9 | 0.0180% | 8660297.92 | 595.75 | 10 | 0.2701% |
| 8 | 9975027.62 | 28.21 | 9 | 0.0170% | 9998274.80 | 842.11 | 10 | 0.2501% |
| 9 | 11037421.07 | 35.46 | 9 | 0.0198% | 11065459.46 | 845.75 | 18 | 0.2739% |
| 10 | 12293845.76 | 45.20 | 9 | 0.0196% | 12321970.06 | 889.80 | 24 | 0.2484% |
| 11 | 13516535.03 | 55.23 | 9 | 0.0199% | 13547749.30 | 821.01 | 25 | 0.2509% |
| 12 | 14739230.35 | 69.35 | 9 | 0.0203% | 14773600.31 | 1069.94 | 29 | 0.2535% |
| 13 | 15978839.21 | 86.94 | 9 | 0.0205% | 16013723.12 | 1072.55 | 35 | 0.2388% |
| 14 | 18496037.13 | 125.37 | 9 | 0.0207% | 18537685.25 | 1155.91 | 38 | 0.2459% |
| 15 | 20941642.18 | 168.45 | 9 | 0.0221% | 20989117.36 | 1766.46 | 39 | 0.2488% |
| 16 | 22202422.89 | 192.17 | 9 | 0.0226% | 22252897.80 | 1233.14 | 50 | 0.2500% |
| 17 | 24665023.43 | 252.63 | 9 | 0.0238% | 24720955.29 | 3490.16 | 52 | 0.2506% |
| 18 | 27110749.50 | 75.01 | 9 | 0.0250% | 27172517.96 | 3193.77 | 55 | 0.2529% |
| 19 | 29649167.31 | 92.22 | 9 | 0.0252% | 29717165.57 | 3886.75 | 54 | 0.2546% |
| 20 | 30872031.23 | 98.05 | 9 | 0.0257% | 30947624.26 | 3923.02 | 58 | 0.2706% |
| Sum |  | 1376.23 | 180 |  |  | 26372.6 | 560 |  |

表5-5 SQP算法与BADMM算法求解大规模实例结果

1. Theerthamalai A, Maheswarapu S. An effective non-iterative “-logic based” algorithm for economic dispatch of generators with cubic fuel cost function[J]. International Journal of Electrical Power & Energy Systems, 2010, 32(5):539-542.
2. Ziane I, Benhamida F, Graa A. Simulated annealing algorithm for combined economic and emission power dispatch using max/max price penalty factor[J]. Neural Computing & Applications, 2016:1-9.
3. [Online]. Available: <https://www.inverseproblem.co.nz/OPTI/index.php/Main/HomePage>.
4. Wang F, Xu Z, Xu H K. Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems[J]. Eprint Arxiv, 2014.