

基于有序 BDL 树集的船舶电网潮流算法及执行时间分析

侯 玉^{1,2}, 吴旭生¹, 杨 波¹, 冯 源¹

(1. 海军工程大学电气工程学院, 湖北 武汉 430033; 2. 海军驻某厂军事代表室, 辽宁 大连 116005)

摘 要: 针对嵌入式船舶电力模拟训练系统在频繁模拟操作指令下对潮流计算的快速性需求以及嵌入式模拟训练系统极其有限的运算资源之间的矛盾, 提出一种基于有序 BDL 树集的船舶电网潮流算法。该算法在常规回推前推法基础上对节点编号进行优化, 将多电站船舶电力系统在实际开环运行模式下等效为由多个辐射状电网构成的有序 BDL 树集, 树集中的每个有序 BDL 树包含单个电站中的主配电板层、配电层和负载层, 并把发电机潮流整合入主配电板层, 避免了常规潮流算法中节点编号突变问题。对该算法的时间复杂度分析结果及实际运行结果进行比较分析, 证明该算法比常规船舶电网潮流算法更优。

关键词: 船舶电网; 潮流计算; 有序 BDL 树集; 节点编号

中图分类号: TM 744 文献标志码: A 文章编号: 1673-6540(2018)10-0041-05

Ship Power Flow Algorithm and Its Time Complexity Analysis Based on Ordered BDL Tree Sets

HOU Yu^{1,2}, WU Xusheng¹, YANG Bo¹, FENG Yuan¹

(1. College of Electric Engineering, Naval Engineering University, Wuhan 430033, China;

2. Naval Representative Office, Dalian 116005, China)

Abstract: Aiming at the contradiction between the speed requirement of power flow calculation under frequent simulation operation instructions and the extremely limited computing resources of the embedded marine power simulation training system, a power flow calculation algorithm for marine power grid based on ordered BDL tree set was proposed. The algorithm optimized the node numbering method on the basis of common methods, and regarded the multi-station marine power system in the actual open-loop operation mode as the ordered BDL tree composed of multiple radial power grids. Each ordered BDL tree contained the main power distribution board, distribution layer and load layer in a single power station, and integrated the generator flow into the main power distribution board layer, avoiding the problem of node numbering mutation in the common power flow algorithms. The time complexity analysis results and the actual operation results of the algorithm were analyzed and compared, which showed that the proposed algorithm was better than the common marine power flow algorithms.

Key words: marine power grid; power flow calculation; ordered BDL tree set; node numbering

0 引 言

嵌入式船舶电力模拟训练系统能够在船舶运行模式下供操作人员进行培训、练习和考核, 可在实际环境中快速提高操作人员的操作技能, 节省模拟器开发成本及训练用时。因此, 嵌入式船舶

电力模拟训练系统近年来得到了迅速的发展^[1-4]。

在嵌入式船舶电力模拟训练系统运行过程中, 要对各种正确或错误的操作指令给出准确的响应, 不仅要有准确的操作逻辑响应, 还必须对电网拓扑结构变化进行潮流计算, 以尽可能模拟出

逼近实际情况的潮流分布。船舶电网由于特殊的结构形式,常使用与陆地电网不同的专用潮流算法,通常有基于广度优先编号算法的回推前推法、节点电势法等^[5-8]。

这些算法可用于电力系统仿真运行、分析研究及实时监控中,但在嵌入式模拟训练系统中应用时遇到以下问题:一是模拟训练系统需要频繁响应用户的操作,导致模拟电力系统网络拓扑结构不断发生改变,从而对船舶电力模拟训练系统电网拓扑分析和潮流计算的实时性提出较高要求;二是嵌入式模拟训练系统仅可使用运算性能非常有限的舰载电力监控系统的运算资源,不能影响电力监控系统本身的正常运行。这要求嵌入式模拟训练系统的潮流算法必须尽可能少地占用运算资源,同时有较好的实时性。

在广度优先编号的回推前推潮流算法基础上,已提出了一种基于有序 GBDL 树集的节点编号算法,将单个电站构建成一个有序 GBDL 树,多个电站构成有序 GBDL 树集。每个有序 GBDL 树包含发电机层 (Generation, G)、主配电板层 (Bridge, B)、配电层 (Distribution, D) 和负载层 (Load, L),该编号法根据每个主配电板节点的入度大小对树集中 B 层再进行细化分层,之后对树集用广度优先搜索进行编号。该算法可用于嵌入式船舶电力模拟训练系统。

本文提出了一种基于有序 BDL 树集的节点编号算法。该节点编号算法把发电机整合到主配电板层中,把船舶电网拓扑结构等效为 BDL 树集,再对树集采用广度优先搜索编号,可避免运算过程中编号突变情况。

本文对算法的时间复杂度进行了分析,证明了该算法的求解总时间优于采用广度优先编号以及采用其他诸如树的前序、中序、后序编号,深度优先编号的潮流算法。最后针对某实际算例进行了对比验证。

1 有序 BDL 树集理论

有序 GBDL 树集在逆序递推节点注入潮流和支路潮流时,在到达 B 层的节点即会终止,然后需要从 G 层节点开始,递推至 B 层的节点。因此,为了使树集结构中潮流计算递推顺序一致,现提出一种有序 BDL 树集,其核心思想是将 G 层节

点及其相关设备合并至 B 层节点中,得到有序 BDL 树集。

定义 1:定义任意 BDL_i 为船舶电网中的某个分层有向树,定义任意的 $S_{BDL} = \{BDL_1, BDL_2, \dots\}$ 为某个连通的 BDL_i 的集合。定义任意的 S_n 为 S_{BDL} 中所有节点集合,任意的 S_{Br} 为 S_{BDL} 中所有支路集合,任意的 S_{BBr} 为 S_{BDL} 中所有跨接支路集合,电网拓扑中所有电源均通过 BBr 与 S_{BDL} 相连接。任意的 S_{CBr} 为 S_{BDL} 中所有普通支路集合。定义任意 BDL 树集:BDL 有向树相互间仅通过一条连通的跨接支路相连的集合。

定义 2(有序 BDL 树):设船舶电网中任意的 BDL 树为 BDL_i ,按广度优先搜索,对 BDL_i 树中的各个节点按等级的顺序依次排序编号,等级小的节点优先编号。同一等级的节点内部,度小的节点优先编号。树中支路的编号顺序取决于节点的编号顺序。定义首端节点 FN 为支路的两端节点中编号较小的节点,末端节点 TN 为编号较大的节点。首先按 FN 的大小进行支路的排序,在 FN 相同的前提下,按 TN 的编号顺序进行支路排序。通过上述编号方法形成的 BDL 树定义为有序 BDL 树。

BDL 树对应的有序 BDL 树如图 1 所示。其中,B 表示支路编号,N 表示节点编号,L 表示负载编号。

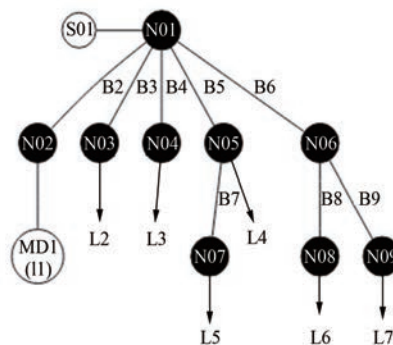


图 1 BDL 树对应的有序 BDL 树

定义 3(跨接支路集合):船舶电网的拓扑结构中,连接有序 BDL 树的支路集合称为跨接支路集,记为 $S_{Bridge} = \{Bridge_1, Bridge_2, \dots, Bridge_n\}$,跨接支路 $Bridge_k$ 的等级高于其他所有支路。

定义 4(S_{BN} 内部子等级及编号规则):设某 S_{BDL} 对应的配电及跨接中心节点集合为 S_{BN} ,以

S_{BN} 中度最大的节点为树根可形成 Bridge 子树, 则 S_{BN} 中任意一个 BN_k , 其在 S_{BN} 内部的子等级定义为 Bridge 子树中所在的层。Bridge 子树与有序 BDL 树的编号规则相同。

定义 5 (有序 BDL 树集): 对于 CJ 电网的任意 BDL 树集。按照其中 S_{BN} 的子等级对此树集中的树进行排序。然后按照树的先后顺序, 对所有

树的节点按照各树和节点的等级统一进行编号。随后, 按照有序 BDL 树支路编号规则以及有序 BDL 树集 S_{BN} 内部子等级及编号规则, 统一对此树集中的所有支路进行编号。通过此方法形成的 BDL 树集定义为有序 BDL 树集 OS_{BDL} 。

例如, 有序 BDL 树的编号方法如图 1 所示, 有序 BDL 树集的编号方法如图 2 所示。

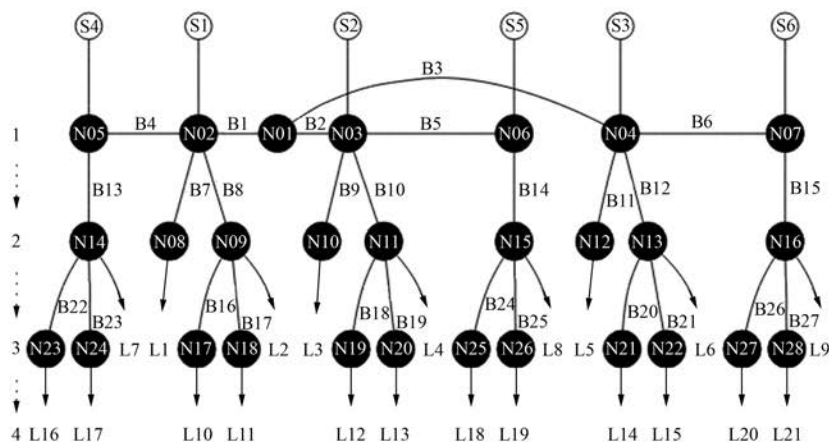


图2 BDL 树集对应的有序 BDL 树集

定理 1: 对于任意有序 BDL 树集中的任意路径, 其对应的节点编号按 $B \rightarrow D \rightarrow L$ 的顺序由小到大, 且是连续的。其对应的支路编号按 $B \rightarrow D \rightarrow L$ 的顺序由小到大, 且是连续的。定理 1 保证了基于有序 BDL 树集的船舶电网拓扑分析与潮流计算可参考树的性质。

定理 2: 基于有序 BDL 树的编号顺序, 采用回推前推法或节点电势法的船舶电网潮流计算每一次迭代始终可解。

定理 3: 基于有序 BDL 树集的编号顺序, 采用回推前推法或节点电势法的船舶电网潮流计算每一次迭代始终可解。

2 基于有序 BDL 树集的开断潮流算法时间复杂度分析

由定理 2 和定理 3 描述的船舶电网潮流计算方法本身是通用的, 即使船舶电网编号不采用有序 BDL 树集的方式, 例如仍采用传统的回推前推法或节点电势法进行编号, 利用上述算法求解船舶电网潮流仍然可行。但问题在于按照传统的回推前推法或节点电势法对船舶电网进行迭代遍历

时, 由于各个 BDL 树之间没有进行排序, 因此在逆序或顺序求解至 BN 和 Bridge 支路潮流时可能不满足求解条件, 需要搜索满足条件的 BN 和相应的 Bridge 支路算法才能继续。那么, 在使用基于有序 BDL 树集的电网潮流算法过程中, 逆序递推节点注入潮流和支路潮流时, 递推顺序不会发生突变。

定理 4: 对于任意相同的 CJ 可解电网运行拓扑结构, 基于有序 BDL 树(集)的 CJ 电网潮流计算, 其算法求解总时间优于采用基于有序 GBDL 树(集)的算法。基于有序 GBDL 树集的船舶电网潮流计算, 其算法求解总时间优于采用同样潮流算法的广度优先编号、树的前序或后序编号、深度优先编号以及树的中序编号。

证明: 对于任意一个 CJ 电网, 潮流算法的计算机可执行程序的单次迭代, 软件运行求解的总时间 t_{solve} 均可分解为搜索节点的总时间 t_{sum_nfa} 与所有节点潮流计算的总时间 t_{sum_nc} 之和:

$$t_{solve} = t_{sum_nfa} + t_{sum_nc} \quad (1)$$

设潮流算法计算完成需要迭代的次数为 n_{iter} , 潮流计算总时间为

$$t_{\text{sum_solve}} = n_{\text{iter}}(t_{\text{sum_nfa}} + t_{\text{sum_nc}}) \quad (2)$$

对于任意相同的 CJ 可解电网运行拓扑结构, 基于有序 BDL 树(集)的 CJ 电网潮流算法和基于有序 GBDL 树(集)的算法, 其需要完成的迭代次数 n_{iter} 均相同, 即:

$$n_{\text{iter}}^{\text{GBDL}} = n_{\text{iter}}^{\text{BDL}} = n_{\text{iter}} \quad (3)$$

设任意 CJ 可解电网运行拓扑结构为 S , 对应的有序 GBDL 树(集)为 OS_{GBDL} , 对应的有序 BDL 树(集)为 OS_{BDL} , 其中, 节点个数为 $|S_{\text{N}}^{\text{GBDL}}|$ 和 $|S_{\text{N}}^{\text{BDL}}|$, GN 节点个数为 $|S_{\text{GN}}^{\text{GBDL}}|$, 支路个数为 $|S_{\text{Br}}^{\text{GBDL}}|$ 和 $|S_{\text{Br}}^{\text{BDL}}|$, 发电机组个数为 $|S_{\text{G}}|$, 负荷个数为 $|S_{\text{L}}^{\text{GBDL}}|$ 和 $|S_{\text{L}}^{\text{BDL}}|$, 于是根据定理 3, 有:

$$|S_{\text{N}}^{\text{GBDL}}| - |S_{\text{Br}}^{\text{GBDL}}| = 1 \quad (4)$$

$$|S_{\text{N}}^{\text{BDL}}| - |S_{\text{Br}}^{\text{BDL}}| = 1 \quad (5)$$

$$|S_{\text{N}}^{\text{GBDL}}| - |S_{\text{N}}^{\text{BDL}}| = |S_{\text{Br}}^{\text{GBDL}}| - |S_{\text{Br}}^{\text{BDL}}| =$$

$$|S_{\text{GN}}^{\text{GBDL}}| = |S_{\text{G}}| \geq 1 \quad (6)$$

$$|S_{\text{L}}^{\text{GBDL}}| - |S_{\text{L}}^{\text{BDL}}| = 0 \quad (7)$$

由两种潮流算法的流程可知, 对于 $\forall S$, 两种算法的单个节点的平均访问次数 $\overline{n_{\text{N}}^{\text{fa}}}$, 单个支路的平均访问次数 $\overline{n_{\text{Br}}^{\text{fa}}}$, 单个发电机组的平均访问次数 $\overline{n_{\text{G}}^{\text{fa}}}$, 单个支路的平均访问次数 $\overline{n_{\text{L}}^{\text{fa}}}$ 均相同。

根据计算机内存寻址的原理, 从当前节点的地址跳转到下一个待访问节点的地址, 其寻址时间是一个常量, 设为 T_{fa} 。

由于静态内存分配的地址是连续的, 因此采用静态内存分配的节点集, 其所有元素的地址已知。从而, 从当前节点跳转到任意待访问节点的寻址时间均为 T_{fa} 。

由于动态内存分配的地址由堆进行分配, 是随机的, 因此采用动态内存分配的节点集, 其所有节点的地址未知。但是如果顺序递推或逆序递推, 即按堆所建立的完全二叉树对当前节点的下一个元素或前一个元素进行查找, 则根据堆的性质, 寻址时间均为 T_{fa} 。

于是, 有:

$$t_{\text{sum_nfa}}^{\text{GBDL}} = |S_{\text{N}}^{\text{GBDL}}| \cdot \overline{n_{\text{N}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{Br}}^{\text{GBDL}}| \cdot \overline{n_{\text{Br}}^{\text{fa}}} \cdot T_{\text{fa}} +$$

$$|S_{\text{G}}^{\text{GBDL}}| \cdot \overline{n_{\text{G}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{L}}^{\text{GBDL}}| \cdot \overline{n_{\text{L}}^{\text{fa}}} \cdot T_{\text{fa}} \quad (8)$$

$$t_{\text{sum_nfa}}^{\text{BDL}} = |S_{\text{N}}^{\text{BDL}}| \cdot \overline{n_{\text{N}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{Br}}^{\text{BDL}}| \cdot \overline{n_{\text{Br}}^{\text{fa}}} \cdot T_{\text{fa}} +$$

$$|S_{\text{G}}^{\text{BDL}}| \cdot \overline{n_{\text{G}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{L}}^{\text{BDL}}| \cdot \overline{n_{\text{L}}^{\text{fa}}} \cdot T_{\text{fa}} \quad (9)$$

所以,

$$t_{\text{sum_nfa}}^{\text{GBDL}} - t_{\text{sum_nfa}}^{\text{BDL}} =$$

$$|S_{\text{N}}^{\text{GBDL}}| \cdot \overline{n_{\text{N}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{Br}}^{\text{GBDL}}| \cdot \overline{n_{\text{Br}}^{\text{fa}}} \cdot T_{\text{fa}} +$$

$$|S_{\text{G}}^{\text{GBDL}}| \cdot \overline{n_{\text{G}}^{\text{fa}}} \cdot T_{\text{fa}} + |S_{\text{L}}^{\text{GBDL}}| \cdot \overline{n_{\text{L}}^{\text{fa}}} \cdot T_{\text{fa}} -$$

$$|S_{\text{N}}^{\text{BDL}}| \cdot \overline{n_{\text{N}}^{\text{fa}}} \cdot T_{\text{fa}} - |S_{\text{Br}}^{\text{BDL}}| \cdot \overline{n_{\text{Br}}^{\text{fa}}} \cdot T_{\text{fa}} -$$

$$|S_{\text{G}}^{\text{BDL}}| \cdot \overline{n_{\text{G}}^{\text{fa}}} \cdot T_{\text{fa}} - |S_{\text{L}}^{\text{BDL}}| \cdot \overline{n_{\text{L}}^{\text{fa}}} \cdot T_{\text{fa}} =$$

$$T_{\text{fa}} \cdot [\overline{n_{\text{N}}^{\text{fa}}} \cdot (|S_{\text{N}}^{\text{GBDL}}| - |S_{\text{N}}^{\text{BDL}}|) +$$

$$\overline{n_{\text{Br}}^{\text{fa}}} \cdot (|S_{\text{Br}}^{\text{GBDL}}| - |S_{\text{Br}}^{\text{BDL}}|) +$$

$$\overline{n_{\text{G}}^{\text{fa}}} \cdot (|S_{\text{G}}^{\text{GBDL}}| - |S_{\text{G}}^{\text{BDL}}|) +$$

$$\overline{n_{\text{L}}^{\text{fa}}} \cdot (|S_{\text{L}}^{\text{GBDL}}| - |S_{\text{L}}^{\text{BDL}}|)] =$$

$$T_{\text{fa}} \cdot (\overline{n_{\text{N}}^{\text{fa}}} \cdot |S_{\text{G}}| + \overline{n_{\text{Br}}^{\text{fa}}} \cdot |S_{\text{G}}| + \overline{n_{\text{G}}^{\text{fa}}} \cdot |S_{\text{G}}|) =$$

$$T_{\text{fa}} \cdot |S_{\text{G}}| \cdot (\overline{n_{\text{N}}^{\text{fa}}} + \overline{n_{\text{Br}}^{\text{fa}}} + \overline{n_{\text{G}}^{\text{fa}}}) > 0 \quad (10)$$

由两种潮流算法的流程可知, 对于 $\forall S$, 两种算法的单个节点的平均求解时间 $\overline{t_{\text{N}}^{\text{fa}}}$, 单个支路的平均求解时间 $\overline{t_{\text{Br}}^{\text{fa}}}$, 单个发电机组的平均求解时间 $\overline{t_{\text{G}}^{\text{fa}}}$, 单个支路的平均求解时间 $\overline{t_{\text{L}}^{\text{fa}}}$ 均相同。

于是, 有:

$$t_{\text{sum_nc}}^{\text{GBDL}} = |S_{\text{N}}^{\text{GBDL}}| \cdot \overline{t_{\text{N}}^{\text{fa}}} + |S_{\text{Br}}^{\text{GBDL}}| \cdot \overline{t_{\text{Br}}^{\text{fa}}} +$$

$$|S_{\text{G}}^{\text{GBDL}}| \cdot \overline{t_{\text{G}}^{\text{fa}}} + |S_{\text{L}}^{\text{GBDL}}| \cdot \overline{t_{\text{L}}^{\text{fa}}} \quad (11)$$

$$t_{\text{sum_nc}}^{\text{BDL}} = |S_{\text{N}}^{\text{BDL}}| \cdot \overline{t_{\text{N}}^{\text{fa}}} + |S_{\text{Br}}^{\text{BDL}}| \cdot \overline{t_{\text{Br}}^{\text{fa}}} +$$

$$|S_{\text{G}}^{\text{BDL}}| \cdot \overline{t_{\text{G}}^{\text{fa}}} + |S_{\text{L}}^{\text{BDL}}| \cdot \overline{t_{\text{L}}^{\text{fa}}} \quad (12)$$

所以,

$$t_{\text{sum_nc}}^{\text{GBDL}} - t_{\text{sum_nc}}^{\text{BDL}} = |S_{\text{N}}^{\text{GBDL}}| \cdot \overline{t_{\text{N}}^{\text{fa}}} + |S_{\text{Br}}^{\text{GBDL}}| \cdot \overline{t_{\text{Br}}^{\text{fa}}} +$$

$$|S_{\text{G}}^{\text{GBDL}}| \cdot \overline{t_{\text{G}}^{\text{fa}}} + |S_{\text{L}}^{\text{GBDL}}| \cdot \overline{t_{\text{L}}^{\text{fa}}} -$$

$$|S_{\text{N}}^{\text{BDL}}| \cdot \overline{t_{\text{N}}^{\text{fa}}} - |S_{\text{Br}}^{\text{BDL}}| \cdot \overline{t_{\text{Br}}^{\text{fa}}} -$$

$$|S_{\text{G}}^{\text{BDL}}| \cdot \overline{t_{\text{G}}^{\text{fa}}} - |S_{\text{L}}^{\text{BDL}}| \cdot \overline{t_{\text{L}}^{\text{fa}}} =$$

$$\overline{t_{\text{N}}^{\text{fa}}} \cdot (|S_{\text{N}}^{\text{GBDL}}| - |S_{\text{N}}^{\text{BDL}}|) +$$

$$\overline{t_{\text{Br}}^{\text{fa}}} \cdot (|S_{\text{Br}}^{\text{GBDL}}| - |S_{\text{Br}}^{\text{BDL}}|) +$$

$$\overline{t_{\text{G}}^{\text{fa}}} \cdot (|S_{\text{G}}^{\text{GBDL}}| - |S_{\text{G}}^{\text{BDL}}|) +$$

$$\overline{t_{\text{L}}^{\text{fa}}} \cdot (|S_{\text{L}}^{\text{GBDL}}| - |S_{\text{L}}^{\text{BDL}}|) =$$

$$|S_{\text{G}}| \cdot (\overline{t_{\text{N}}^{\text{fa}}} + \overline{t_{\text{Br}}^{\text{fa}}} + \overline{t_{\text{G}}^{\text{fa}}}) > 0 \quad (13)$$

所以,

$$t_{\text{sum_solve}}^{\text{GBDL}} - t_{\text{sum_solve}}^{\text{BDL}} = n_{\text{iter}}^{\text{GBDL}} \cdot (t_{\text{sum_nfa}}^{\text{GBDL}} + t_{\text{sum_nc}}^{\text{GBDL}}) -$$

$$n_{\text{iter}}^{\text{BDL}} \cdot (t_{\text{sum_nfa}}^{\text{BDL}} + t_{\text{sum_nc}}^{\text{BDL}}) =$$

$$\begin{aligned} n_{iter} \cdot (t_{sum_nfa}^{GBDL} - t_{sum_nfa}^{BDL} + t_{sum_nc}^{GBDL} - t_{sum_nc}^{BDL}) = \\ n_{iter} \cdot |S_G| \cdot [T_{fa} \cdot (\overline{n_N^{fa}} + \overline{n_{Br}^{fa}} + \overline{n_G^{fa}}) + \\ (\overline{t_N^{fa}} + \overline{t_{Br}^{fa}} + \overline{t_G^{fa}})] > 0 \end{aligned} \quad (14)$$

证明完毕。

3 结 语

以图 2 中的 28 个节点电力系统为例对所提算法进行验证。测试方法如下:对电网中的支路集按图 2 中的编号排序形成有序支路集,每次分别断开此有序支路集中的一条支路。然后对潮流计算结果进行测试,求取相应算法的平均执行时间。

上述开断潮流程序的测试结果平均执行时间以及主要硬件和软件测试环境如表 1 所示。

表 1 算法测试环境及平均执行时间

参数名称	参数值
基于有序 BDL 树集编号潮流算法	502 μs
基于有序 GBDL 树集编号潮流算法	502 μs
基于广度优先搜索编号潮流算法	598 μs
基于树的前序编号潮流算法	688 μs
基于树的后序编号潮流算法	686 μs
基于深度优先搜索编号潮流算法	825 μs
基于树的中序编号潮流算法	826 μs
CPU	Intel E4600+1.6 GHz
内存	DDR800, 4 G
编程语言	Rust 1.11

可见,在上述测试环境下,基于有序 BDL 树集编号潮流算法平均执行时间最短,仅为 502 μs。因此,其时间复杂度优于其他所有算法。

【参 考 文 献】

[1] 王悍泉,金炳哲,龚喜文.船舶电力系统模拟训练系统的设计与实现[J].上海船舶运输科学研究所学报,2016,39(1): 49-53.

[2] 谢辉,卜乐平,冯源.船舶电力模拟训练评估系统的设计和开发[J].舰船科学技术,2013,35(11): 45-48.

[3] 徐远新,林平,徐海刚.基于 HLA 的舰载嵌入式模拟训练系统设计方案[J].军事运筹与系统工程,2012,26(2): 51-54.

[4] 潘昕,侯新国,夏立,等.船舶电力系统的建模与仿真[J].电力科学与工程,2013,29(4): 49-53.

[5] 康军,马伟明,付立军,等.舰船电力系统潮流计算方法[J].武汉理工大学学报(交通科学与工程版),2008,32(1): 32-35.

[6] 冀欣,张晓峰.基于节点电势法的舰船电力系统潮流计算方法[J].船海工程,2007,36(5): 132-135.

[7] 马秀林,徐岩,王增平,等.静态优化法编号在潮流计算中的应用[J].电力科学与工程,2007,23(2): 12-14.

[8] 孙秋野,陈会敏,杨家农,等.牛顿类潮流计算方法的收敛性分析[J].中国电机工程学报,2014,34(13): 2196-2200.

收稿日期: 2018-04-25

(上接第 23 页)

[4] 闫鹤鸣,李相俊,麻秀范,等.基于超短期风电预测功率的储能系统跟踪风电计划出力控制方法[J].电网技术,2015,39(2): 432-439.

[5] JIN C L, LU N, LU S, et al.A coordinating algorithm for dispatching regulation services between slow and fast power regulating resources[J].IEEE Transactions on Smart Grid,2014,5(2): 1043-1050.

[6] 章竹耀,郭晓丽,张新松,等.储能电池平抑风功率波动策略[J].电力系统保护与控制,2017,45(3): 62-68.

[7] 赵兴勇,王帅,杨涛,等.平抑风电场功率波动的复合储能系统控制策略[J].电机与控制应用,2017,

44(7): 108-115.

[8] 薛亚林,周建萍,崔屹.基于直流微电网的混合储能协调控制策略仿真研究[J].电机与控制应用,2017,44(8): 19-26.

[9] 胡泽春,丁华杰,孔涛.风电-抽水蓄能联合日运行优化调度模型[J].电力系统自动化,2012,36(2): 36-41.

[10] 崔杨,彭龙,仲悟之.双馈型风电场群无功分层协调控制策略[J].中国电机工程学报,2017,37(17): 4300-4307.

[11] 贺军,张俊洪.风储微电网运行控制研究[J].电机与控制应用,2015,42(12): 50-54.

收稿日期: 2018-04-25