

船舶 BOM 数据遍历搜索算法模块-技术要求

1.总体要求

针对 BOM(物料清单)数据的遍历与搜索需求,设计并实现一套精确的算法模块。BOM 数据呈树状或图状结构,包含父子关系及部件属性。算法应支持高效查询、统计分析及可视化展示,并具备良好的扩展性与维护性。实现方案需充分考虑工业制造场景的实际需求,包括对大型复杂产品 BOM 的处理能力支持多层级嵌套结构、循环引用检测、异构数据源整合。算法模块需要提供标准 API 接口,便于与现有 PLM、ERP、MES 等系统集成,确保在分布式环境下的可靠性和可扩展性。投标人需提供完整的技术架构设计文档,说明算法选型依据、性能优化策略及容错处理机制。

2.算法功能要求

投标人须实现以下核心算法,并提供完整源代码及算法说明文档:

2.1 全序化 BOM 构型的深度优先拓扑规约算法

功能要求:

- ① 实现基于栈的深度优先遍历算法,支持递归和迭代两种实现方式,按先子后兄弟"顺序访问所有节点。
- ② 算法需处理多种 BOM 结构变体,包括多叉树、带权树、带循环检测的图结构。
- ③ 提供前序、中序、后序三种遍历模式,支持自定义节点访问回调函数,允许在遍历过程中修改节点属性或结构。
- ④ 算法应具备中断恢复机制,支持从任意节点开始遍历或指定遍历深度限制。

2.2 BOM 广度优先层级规约算法

功能要求:

- ① 实现基于队列的广度优先遍历算法,按层级逐层访问节点,支持层级感知的节点处理。
- ② 算法需实现高效的双端队列管理,支持动态队列大小调整,避免频繁内存分配。
- ③ 提供层级统计功能,自动计算每层节点数、属性汇总值。
- ④ 支持层级跳跃遍历,可跳过指定层级或仅遍历特定深度范围。
- ⑤ 实现基于优先级的广度优先遍历,允许按节点属性(如关键度、成本等)调整访问顺序。

2.3 多尺度层级索引的递归分治构建算法

功能要求:

- ① 在遍历过程中为节点构建多层次索引结构,包括层级索引、路径编码索引、属性值索引。
- ② 层级索引采用倒排表结构,建立层级号到节点列表的快速映射。
- ③ 路径编码采用紧凑的字符串或数值编码,支持快速父子关系查询和路径匹配。
- ④ 属性值索引针对常用查询属性(如物料编码、版本号)建立 B+树或哈希索引。
- ⑤ 支持增量索引更新,当 BOM 局部变更时仅重建受影响部分的索引。

2.4 双模态融合遍历的启发式择优遍历算法

功能要求:

- ① 结合 DFS 与 BFS 优点,实现自适应混合遍历策略。
- ② 算法需实时分析 BOM 结构特征(深度、宽度、分支因子、子树大小分布),动态选择最优遍历策略。
- ③ 实现基于机器学习的策略选择模型,根据历史遍历性能数据优化决策。

- ④ 提供混合遍历的进度监控和策略切换日志。

2.5 基于父子锚定关系的递推式树形拓扑聚簇算法

功能要求:

- ① 从多种数据源（邻接表、边列表、父子关系表）构建完整的 BOM 树结构。
- ② 算法需支持循环引用检测和自动修复，提供循环路径报告和修复建议。
- ③ 处理多根节点森林，为每个连通分量构建独立子树。
- ④ 支持异构节点类型（如设计件、工艺件、制造件等）的统一建模。
- ⑤ 实现结构规范化，自动检测并修复常见结构问题（如孤立节点、重复边、不一致的父子关系等）。

2.6 变更驱动的递进式增量拓扑枚举算法

功能要求:

- ① 当 BOM 发生局部变更时，智能识别受影响范围，仅对变更子树进行重遍历。
- ② 实现变更影响分析，基于依赖图确定需要重遍历的节点集合。
- ③ 支持多种变更类型：节点增删、属性修改、结构调整、关系变化。
- ④ 提供增量结果与原有结果的智能合并策略，处理合并冲突。
- ⑤ 实现遍历状态持久化，支持从上次中断点继续增量遍历。

2.7 基于拓扑不变量挖掘的图结构归约与增强算法

功能要求:

- ① 对 BOM 图结构进行深度分析，识别关键路径、瓶颈节点、冗余结构、循环依赖等特征。
- ② 实现图聚类算法，将大型 BOM 分解为逻辑模块，便于分治处理。
- ③ 提供结构优化建议，如重构深度过大的子树、合并相似结构、消除冗余节点。
- ④ 实现图度量计算，包括连通性、直径、中心性、模块度等指标。
- ⑤ 支持可视化分析生成结构热力图、依赖关系图、聚类分布图。

3. 技术指标要求

- ① 数据结构：使用内存高效的数据结构表示 BOM 树，如压缩前缀树、位图索引、飞地数组等。数据结构需支持序列化和反序列化，便于持久化存储和网络传输。
- ② 算法复杂度：各算法时间/空间复杂度需严格满足上述指标，并提供详细的理论分析证明。需提供最坏情况、平均情况、最佳情况下的复杂度分析，以及实际性能测试数据对比。
- ③ 并发支持：关键算法需支持多线程安全访问，实现读写锁、无锁数据结构、线程池等并发控制机制。支持分布式环境下的算法执行。
- ④ 异常处理：完善的错误检测与处理机制，包括输入验证、边界检查、资源管理、异常恢复。提供详细的错误代码和诊断信息，支持错误分类和优先级处理。实现熔断机制和降级策略，确保系统在异常情况下的可用性。

4. 交付物要求

- ① 源代码：完整可编译/运行的算法实现，包含详细注释等。
- ② 代码需遵循所选语言的编码规范，通过静态代码分析工具检查。提供构建脚本和依赖管理文件，确保一键编译部署。
- ③ 算法说明文档：每个算法的详细设计文档。

5. 验收标准

- ① 功能验收：所有算法功能通过完整的测试用例验证，测试覆盖率不低于 95%。性能测试需在标准硬件配置下进行，验证各项性能指标达标。
- ② 文档验收：算法说明文档完整、清晰。文档需通过技术评审，确保内容准确表述清

晰、示例正确。

③ 代码质量：代码符合所选语言的通用规范，通过静态代码检查。注释覆盖率不低于30%，关键算法需有详细注释。