计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(10):268−272 [doi: 10.15888/j.cnki.csa.006593]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 Hadoop 的车辆调度算法优化及应用①

陈　燕[1], 于　放[2], 田　月[2], 刘　璐[2]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 沈阳计算技术研究所, 沈阳 110168)
通讯作者: 陈　燕, E-mail: 1728731328@qq.com

摘　要: 随着互联网技术的快速发展, 各行各业所产生的信息数据也在以指数级的速度增长. 传统的车辆调度算法已经不能够很好地解决车辆调度问题中出现的实时性, 大规模等问题. 因此, 本文构建了一种基于 Hadoop 的动态车辆调度并行智能优化算法. 该算法以传统遗传算法为基础, 通过改善遗传算法全局优化能力弱和收敛于局部次优解的问题, 并利用 Hadoop 平台的并行计算机制对传统遗传算法进行改进, 使其能够有效应对大规模、快速响应的车辆调度. 数值计算结果表明: 基于 Hadoop 的车辆调度算法能够有效提升传统调度算法的优化性能, 在处理大规模车辆调度问题时具有良好的加速比.

关键词: 智能调度; Hadoop; 车辆调度算法; 算法优化; 启发式算法

引用格式: 陈燕,于放,田月,刘璐.基于 Hadoop 的车辆调度算法优化及应用.计算机系统应用,2018,27(10):268–272. http://www.c-s-a.org.cn/1003-3254/6593.html

Optimization and Application of Vehicle Scheduling Algorithm Based on Hadoop

CHEN Yan[1], YU Fang[2], TIAN Yue[2], LIU Lu[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: With the rapid development of Internet technology, the information data generated by all industries and professions is growing at an exponential rate. The traditional vehicle scheduling algorithm in dealing with dynamic vehicle scheduling problem, already cannot satisfy real-time and large-scale scenario, while big data in Hadoop technology can be a good solution. Therefore, this study constructs a dynamic vehicle scheduling parallel intelligent optimization algorithm based on Hadoop. Based on traditional genetic algorithm, the Hadoop platform parallel computing mechanism is used to improve the weak global optimization ability and converging to local optimal solution of the algorithm. The improved algorithm can effectively cope with massive and rapid response of the vehicle scheduling. The result of numerical calculation shows that the algorithm of vehicle scheduling based on Hadoop can effectively improve the optimization performance of traditional scheduling algorithm and has a good acceleration ratio when dealing with large-scale vehicle scheduling problems.

Key words: intelligent scheduling; Hadoop; vehicle scheduling algorithm; algorithm to optimize; heuristic algorithm

移动互联网、物联网等技术快速走向应用, 现代物流技术正在实现跨越式发展. 这种进步正在深刻影响着社会信息化进程, 以物流技术的发展牵引着人类社会的不断进步. 正是现代物流技术的进步使得人们能够足不出户的实现商品的消费, 交易便捷化也进一步要求现代物流技术能够更为快速地实现商品的配送.

① 收稿时间: 2018-03-12; 修改时间: 2018-03-28; 采用时间: 2018-04-12; csa 在线出版时间: 2018-09-28

车辆调度问题主要分为静态和动态两类. 静态车辆无法适应动态的运营环境, 现代车辆调度问题研究热点集中于动态车辆调度以及结合数字地图导航技术的动态路径规划等方面. 如冯亮等针对客户需求、配送车辆实时匹配需求, 以 GPS/GIS 地图数据为基础, 应用混合遗传算法实现了动态车辆调度及路径规划, 能够有效提升物流配送效率、降低物流企业成本以及改善物流服务质量[1]. 并提出以客户需求、配送车辆/节点以及路网状态等为约束条件, 构建了混合整数线性规划模型, 同样实现了动态快速的车辆动态调度及路径规划[2]. 对于能够获取公交车辆 GPS 数据的应用场景, 针对公交车辆动态调度问题, 可利用大数据实现公交车辆相关参数的预测, 并利用改进遗传算法实现车辆的高效调度[3]. 针对实时性要求更高的应急车辆调度及配置问题, 研究学者利用城市快速路网相对固定的特征, 通过构建路网模型, 并利用混合蛙跳算法实现了快速准确的路径实时动态规划[4]. 而对于网购配送过程中, 末端配送存在的需求随机性强、配送服务时效性要求高等问题, 提出了移动配送模式, 并以配送与惩罚成本为目标函数, 对移动配送模式的实时动态求解进行了验证, 从研究成果看该模型能够有效提升配送反应效率, 实现了高效车辆调度[5]. 显然针对不同的应用场景, 现有研究的算法难以通用化, 特别是大规模、高实时性的车辆调度问题, 传统启发式算法是难以在实际应用场景中确保实时性的.本文主要实现的就是现有车辆调度算法的高效并行处理,以提高处理大规模、高实时性车辆调度问题的能力.

为此本文针对动态车辆调度过程中对算法实时性要求高, 传统启发式算法性能难以满足不确定环境下多约束的大规模车辆调度的现状, 提出基于 Hadoop 的车辆调度算法优化, 实现传统启发式算法的高效并行计算, 提升算法的加速比及扩展性, 能够有效大规模动态车辆实时调度问题.

1 传统启发式算法

1.1 遗传算法基本原理

遗传算法 (Genetic Algorithm, 简称 GA) 是应用最早的现代启发式优化算法之一, 其基本原理是借鉴自然界生物“优胜劣汰、适者生存”的进化机制, 以遗传变异理论为基础, 实现代际间的迭代搜索, 从而实现随机全局搜索以及优化.

遗传算法通过设计一种代表生物进化基因的编码来实现代际间的迭代搜索, 其基本要素包括: 编码、种群、适应度评估、选择、交叉、变异等. 通常包含以下步骤: (1) 将问题的参数进行编码; (2) 生成初始群体; (3) 计算群体中各个体的适应度函数值; (4) 按选择策略选择将要进入下一代的个体; (5) 按交叉概率 Pc 进行交叉操作; (6) 按变异概率 Pm 进行变异操作; (7) 如果不满足终止准则, 则转到 (3), 否则转入下一步; (8) 将适应度函数值最优的个体作为该问题的最优解, 输出[6].

1.2 粒子群算法基本原理

粒子群算法 (PSO) 属于群智能算法的一种, 是通过模拟鸟群捕食行为设计的. 假设区域里就只有一块食物 (即通常优化问题中所讲的最优解), 鸟群的任务是找到这个食物源. 鸟群在整个搜寻的过程中, 通过相互传递各自的信息, 让其他的鸟知道自己的位置, 通过这样的协作, 来判断自己找到的是不是最优解, 同时也将最优解的信息传递给整个鸟群, 最终, 整个鸟群都能聚集在食物源周围, 即得到问题最优解.

粒子群算法通过设计一种无质量的粒子来模拟鸟群中的鸟, 粒子仅具有两个属性: 速度V和位置X, 速度代表移动的快慢, 位置代表移动的方向. 每个粒子在搜索空间中单独的搜寻最优解, 并将其记为当前个体极值P_{best}, 并将个体极值与整个粒子群里的其他粒子共享, 找到最优的那个个体极值作为整个粒子群的当前全局最优解G_{best}, 粒子群中的所有粒子根据自己找到的当前个体极值P_{best}和整个粒子群共享的当前全局最优解G_{best}来调整自己的速度和位置. 粒子群算法的思想相对比较简单, 主要分为: (1) 初始化; (2) 评价粒子, 即计算适应值; (3) 寻找个体极值P_{best}; (4) 寻找全局最优解G_{best}; (5) 修改粒子的速度和位置.

1.3 并行启发式算法原理

并行启发式算法的基本原理是基于现代并行计算的概念和原理, 同时融合现代启发式算法的全局随机搜索优化能力, 使算法能够更好的适用于大规模种群、动态多样性、更快的收敛速度和全局寻优能力. 目前, 并行启发式算法已经是大数据计算领域的热点研究领域. 其主要优点包括: (1) 并行启发式算法染色体的并行编码形式, 使初始种群具有很好的多样性; (2) 由单机全局搜索到多机局部搜索, 使算法实时性得以提高; (3) 并行启发式算法通过海量数据的预处理, 能够得到 K 个局部优化方案集, 提高了算法的求解质量.

2 基于 Hadoop 的并行启发式车辆调度算法

2.1 算法流程设计

基于 Hadoop 的车辆调度并行启发式算法优化主要分两个阶段: (1) 利用并行遗传算法算法对海量的车辆调度相关数据 (路况的实时监控视频、车辆基本运行状况等) 进行预处理, 得到 K 个局部优化的调度及路径规划方案集 (对创建的方案集进行源节点和方案集是否为空判断), 为下一步启发式算法实现高效全局优化奠定基础. (2) 利用并行的启发式算法 (遗传算法、粒子群算法等) 对局部优化的调度及路径规划方案集进行全局优化搜索, 得到最优的方案.基于 Hadoop 的并行遗传算法充分利用 Hadoop 的主从计算模式, 通过数据并行方式, 实现大规模、高实时的车辆调度优化. 对某一区域内的车辆调度数据进行最短路径分析, 得到局部的最短路径方案集, 根据方案集生成向量列表, 此时启动并行启发式算法, 对车辆调度及路径规划方案集进行全局优化. 主要的 Map Reduce 编程模型包括了 Map、Combine 和 Reduce 3 个阶段.

在执行此算法前, 集群主节点首先将已得到的簇类中心向量列表和簇中心数目广播到各个子节点上. Map 函数的输入依然是各个数据块集合, Map 的输入格式为<key1, value1>形式, key1 为对象 id, valuel 为数据对象向量, Map 函数的逻辑就是将本节点上的数据对象划到离其最近的簇向量中去, 输出格式也是<key2, value2>, key2 为簇向量标识符, value2 为数据对象向量. Map 过程就将集群中的所有点都划入到离其最近的簇中心向量中去, 这样可能会使簇中心向量发生变化, 因此需要再次计算簇中心向量.

Combine 函数的输入是 Map 的输出<key2, value2>, 而此函数的输出依然是键值对可以表示为<key3, value3>, key3 依然还是簇类向量标识符, value3 为相同 key3 的所有向量组合和这些向量的数目.

Reduce 函数处理属于同一簇的所有数据对象向量, 并重新生成新的簇类中心向量, 其输入输出均是键值对形式. 此时调用传统启发式算法, 实现输入数据对象的随机快速搜索.

输入信息是各个子节点的 combine 结果, 输出信息是簇类标识符和新的簇类中<key4, value4>, 在每一次 Reduce 执行完成后, 都需要将新的簇类中心向量与前一次的簇类中心向量进行比较如果满足上文给出的聚类准则函数就可结束迭代, 即认可聚类结果已收敛. 具体流程如图 1 所示.

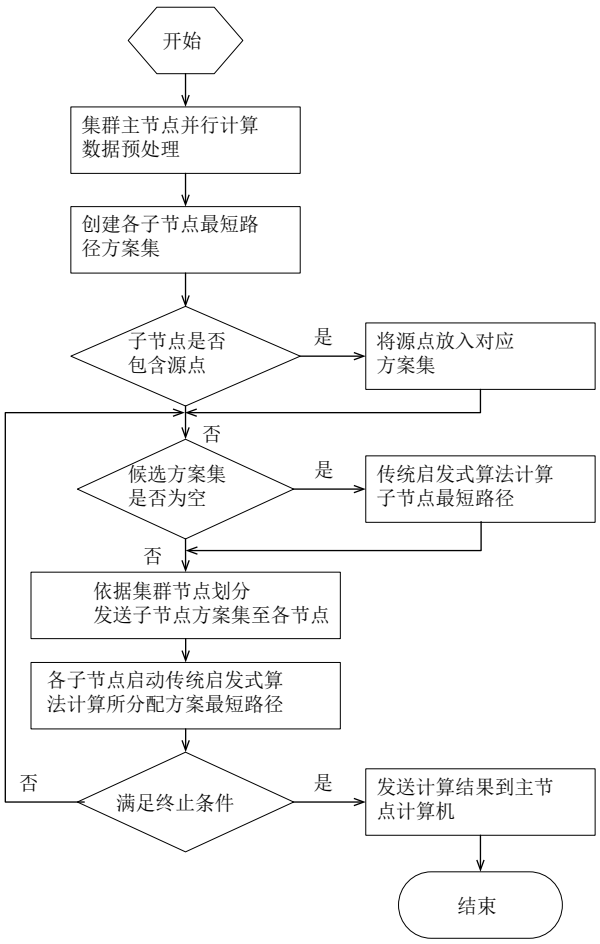

图 1 基于 Hadoop 的并行启发式算法流程

2.2 遗传算法的 MapReduce 并行过程

(1) Map 函数

种群适应值函数的计算 (步骤 (3) 和 (7)) 匹配 Map 函数, 它必须独立于其他实例来计算. 利用 Map 函数基本定义, 计算给定个体的适应值. 并且将最好的个体的路径保存下来最终写到分布式文件系统 HDFS 的全局文件中. 初始化任务的客户端, 在 MapReduce 结束时从 mapper 中读取这些值并检查收敛条件是否满足. GA 每次迭代的 Map 过程伪代码:

```
Map(key, value):
Individual ← INDIVIDUAL REPRESENTATION(key)
fitness ← CALCULATEFITNESS(individual)
EMIT (individual, fitness)
{Keep track of the current best}
if fitness > max then
max ← fitness
```

maxInd ← individual
end if
if all individuals have been processed then
Write best individual to global file in DFS
end if

(2) C. Partitioner

若 GA 的选择算子 (步骤 (4)) 在每个节点本地执行, 空间约束将会人为引入并降低选择压力, 并可能导致收敛时间增加. 因此, 去中心化的、分布式的选择算法成为首选. MapReduce 模型唯一的全局通信点是 Map 和 Reduce 之间的 Shuffle. 在 Map 阶段的最后, MapReduce 框架使用分区将 key/value 对输出给 Reduce 过程. 分区会将中间的 key/value 对在各个 Reduce 之间进行拆分. 函数 GetPartition() 返回给定的 (key, value) 应该发送给的 Reducer. 在默认应用中, 它使用 Hash(key)%numReducers, 以便具有相同 key 的值分配给同一个 reducer, 从而可以应用 Reduce 函数. 通过构建适宜的分区工具, 随机地把个体分配给不同的 Reducer. GA 的随机分割过程伪代码:

int GETPARTITION(key, value, numReducers):
return RANDOMINT(0, numReducers - 1)

(3) Reduce 过程

采用无替换竞争选择算子. 一场竞争在 S 个随机选择的个体直接进行, 选出其中的赢家. 重复这个过程, 重复次数为种群大小. 由于随机选择个体相当于随机对所有个体洗牌并按顺序处理它们, Reduce 函数也按顺序处理每个个体. 最初的个体为最后一轮竞赛而缓存, 当比赛窗口满了, 执行 SelectionAndCrossover 算法. 当交叉窗口满了, 我们使用随机交叉算子.在实际应用中, 可将 S 设为 5, 交叉算子采用选出的两个连续的父代个体进行. GA 每次迭代的 Reduce 过程伪代码:

Initialize processed ←0, tournArray [2· tSize],
crossArray [cSize]
REDUCE(key, values):
while values.hasNext() do
individual ← INDIVIDUALREPRESENTATION(key)
fitness ← values.getValue()
if processed < tSize then
{Wait for individuals to join in the tournament and put them for the last rounds}
tournArray [tSize + processed%tSize] ← individual
else
{Conduct tournament over past window}
SELECTIONANDCROSSOVER()
end if
processed ← processed + 1
if all individuals have been processed then
{Clean up for the last tournament windows}
for k=1 to tSize do
SELECTIONANDCROSSOVER()
processed ← processed + 1
end for
end if
end while
SELECTIONANDCROSSOVER:
crossArray[processed%cSize] ← TOURN (tournArray)
if (processed - tSize) % cSize = cSize - 1 then
newIndividuals ← CROSSOVER(crossArray)
for individual in newIndividuals do
EMIT (individual, dummyFitness)
end for
end if

3 实验及结果分析

3.1 Hadoop 环境构建

环境构建分为硬件环境和软件环境, 其中 Hadoop 计算节点 4 个, 操作系统为 Cent OS, 具体参数如下:

硬件环境: 4 台 PC 机;

软件环境: 操作系统: Cent OS 6.5 版本; JDK: 1.8 版本; Hadoop: 3.0 版本; Mahout: 0.12.1 版本.

3.2 计算实例及分析

本文通过 Maple 软件实现配送网络和验证数据的随机产生, 采用少量数据就能够发现基于 Hadoop 的并行启发式遗传算法的高效性, 而 Hadoop 本身具有处理大数据的能力. 为在论文中体现数据真实性, 采用随机产生的方式以少量数据进行示例性说明 (大量数据无法在文中展示, 也不利于论文结果重现). 假设配送区域限定为 100×100 km^2 的正方形区域, 利用 rand 函数随机获得 20 个静态需求位置数据和 10 个动态需求位置数据, 配送车辆最大配送体积为 9 m^3, 单个需求的体

积最大为 3 m^3, 车辆最大配送距离为 100 km, 配送中心的位置设定为 O(50 km, 50 km). 随机产生的静态需求位置数据如表 1 所示, 动态需求位置数据如表 2 所示.

表 1 静态需求位置数据

序号	需求	位置	序号	需求	位置
1	0.8	[29, 44]	11	0.2	[47, 9]
2	0.5	[26, 12]	12	1.4	[34, 19]
3	0.3	[45, 13]	13	0.2	[44, 13]
4	1.4	[18, 40]	14	0.9	[26, 49]
5	1.7	[32, 16]	15	0.8	[36, 46]
6	0.7	[43, 26]	16	1.7	[29, 3]
7	0.2	[6, 1]	17	0.3	[38, 39]
8	0.2	[19, 3]	18	0.9	[29, 37]
9	1.7	[34, 6]	19	1.2	[5, 28]
10	0.4	[25, 49]	20	0.5	[14, 37]

表 2 动态需求位置数据

序号	需求	时间窗	位置	序号	需求	时间窗	位置
1	1.3	1.5	[41, 43]	6	0.8	1	[8, 19]
2	0.9	0.9	[39, 25]	7	0.7	1.2	[47, 10]
3	1.6	1	[44, 41]	8	0.5	1.7	[13, 49]
4	0.1	1	[2, 15]	9	0.7	1.2	[4, 8]
5	0.8	1.6	[21, 12]	10	0.4	0.9	[9, 9]

为了验证基于 Hadoop 的车辆调度算法改进性能, 对上述车辆调度问题进行求解, 得到基于遗传算法、基于粒子群算法以及基于 Hadoop 的车辆调度算法的基本收敛性能, 如图 2 所示.

由图 2 可知, 三种算法中基于 Hadoop 的车辆调度算法能够在 20 代后实现快速收敛, 这得益于 Hadoop 并行遗传算法的局部可行解集的提前获取和全局并行计算的收敛性.

其次对比分析三者的搜索成功率、优化调度时间等基本性能, 如表 3 所示, 可以看出三种算法中基于 Hadoop 的车辆调度改进算法能够实现全方位的性能提升, 特别是在收敛时间和平均迭代次数上. 在车辆调度节点数量为静态 20, 动态数量为 10 的情况下, 利用 4 台 PC 机, 计算时间由传统启发式算法大约 17.5 s 左右降低到 3.1 s. 对于更大规模的多约束高实时车辆动态调度问题而言, 基于 Hadoop 的并行遗传算法能够更快响应需求.

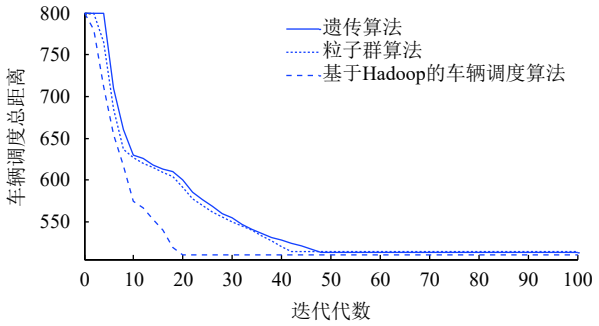

图 2 算法收敛性能

表 3 算法计算结果

算法	最优值	最劣值	平均值	成功率 (%)	平均迭代次数	收敛时间	加速比
遗传算法	563.87	1369	947	15	107.41	15.9	--
粒子群算法	563.87	1573	1056	13	112.12	19.4	--
并行遗传算法	563.87	647	601	51	31.06	3.1	3.7

注: 加速比表示单 PC 机运行时间与 Hadoop 多节点运行时间的比值

4 结论

本文针对传统车辆调度算法在处理动态、实时、大规模车辆调度存在的不足, 特别是启发式算法对初始种群敏感, 充分利用 Hadoop 在处理海量数据的快速性、计算效率等特性, 提出了一种基于 Hadoop 的车辆调度并行优化算法, 快速获取可行的区域最短路径规划方案集, 并通过 Map Reduce 并行编程框架实现该算法来适应大规模车辆调度问题, 有效提高车辆调度算法的计算实时性及优化性能. 仿真结果验证了该算法在处理大规模车辆调度问题时具有良好的全局优化能力、计算加速比以及模型扩展能力.

参考文献

1 冯亮, 梁工谦. 基于 GPS/GIS 协同的动态车辆调度和路径规划问题研究. 计算机科学, 2017, 44(9): 272–276, 285.
2 冯亮, 梁工谦. 联网中物流配送车辆调度目标定位设计与仿真. 计算机仿真, 2017, 34(4): 377–381, 405. [doi: 10.3969/j.issn.1006-9348.2017.04.080]
3 唐德权, 黄金贵, 史伟奇. 基于大数据平台的动态车辆路径调度算法. 计算机工程, 2018, 44(1): 74–78. [doi: 10.3969/j.issn.1000-3428.2018.01.012]
4 张丽平. 基于 GPS 数据的公交车辆动态调度研究[硕士学位论文]. 杭州: 浙江工业大学, 2016.
5 段晓红. 城市快速路网应急车辆动态调度与再配置研究[硕士学位论文]. 北京: 北京交通大学, 2016.