

DOI:10.19651/j.cnki.emt.2210808

基于 Spark 的并行 k 均值聚类模拟退火算法求解 MMTSP *

孙 鉴^{1,2} 李 昊¹ 刘松佐¹ 刘 品¹ 陈 攀¹ 雷冰冰^{1,2}

(1. 北方民族大学计算机科学与工程学院 银川 750021; 2. 图像图形智能处理国家民委重点实验室 银川 750021)

摘要: 多起点闭环回路多旅行商问题是旅行商问题的扩展。针对这个问题文中提出了一种基于 Spark 框架的并行 k 均值聚类模拟退火算法。该算法首先采用 k 均值聚类算法将所有城市分类,然后对应每个类建立一个旅行商问题,并通过一种改进的模拟退火算法对旅行商问题求解,MMTSP 的解由这些类的最短路径之和计算得出。所提算法采用先聚类再执行模拟退火算法的求解策略可以极大的缩减模拟退火的搜索空间,并且由于 Spark 框架可以将聚类算法分好的若干类并行求解,从而更快的得到 MMTSP 问题的最优解。选取 TSPLIB 数据库中若干测试实例进行仿真实验,对求解精度和运行时间两个方面进行测试,与其他几种相关算法进行对比实验。实验结果表明,与目前 FCMPGA、IPGA、IWO 等算法相比,求解精度提高了 5%~40%,求解效率上对比其他算法提升 1~5 倍,尤其在 K 值较大时表现更优。

关键词: 多旅行商问题;旅行商问题;k 均值聚类;模拟退火算法

中图分类号: TP301.6 **文献标识码:** A **国家标准学科分类代码:** 520.3020

Spark-based parallel k-means clustering simulated annealing algorithm to solve MMTSP

Sun Jian^{1,2} Li Hao¹ Liu Songzuo¹ Liu Pin¹ Chen Pan¹ Lei Bingbing^{1,2}

(1. College of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China;

2. Key Laboratory of Image and Graphics Intelligent Processing State Civil Affairs Commission, Yinchuan 750021, China)

Abstract: Multiple Depots Multiple Traveling Salesman Problem is an extension of Traveling Salesman Problem. To solve this problem, a parallel K-means clustering simulated annealing algorithm based on Spark platform is proposed. The algorithm firstly classifies all the cities by K-means clustering algorithm, then establishes a traveling salesman problem for each class, and solves the traveling salesman problem by an improved simulated annealing algorithm. The MMTSP solution is calculated from the sum of the shortest paths of these classes. The proposed algorithm adopts the solution strategy of clustering first and then simulated annealing algorithm, which can greatly reduce the simulated annealing search space. Moreover, Spark platform can divide the clustering algorithm into several classes for parallel solution, so as to obtain the optimal solution of MMTSP problem faster. Several test instances in TSPLIB database are selected for simulation experiments, and the solution accuracy and running time are tested. The experiments are compared with other related algorithms. The experimental results show that, compared with the current FCMPGA, IPGA, IWO and other algorithms, the solution accuracy is improved by 5%~40%, and the solution efficiency is improved by 1~5 times compared with other algorithms, especially when the value of K is large.

Keywords: multiple traveling salesman problem; traveling salesman problem; k-means clustering; simulated annealing algorithm

0 引 言

旅行商问题(traveling salesman problem, TSP)是组合

优化领域中最著名的问题之一^[1]。它的目标是在已知点之间所有距离的情况下,搜索给定点数的最短路径或成本最小的路线。TSP 的一种推广是多旅行推销员问题

收稿日期:2022-07-25

* 基金项目:国家自然科学基金(62062002,62102201)、宁夏自然科学基金(2022AAC03289,2022AAC03245,2021AAC03034)、北方民族大学中央高校基本科研业务费专项资金(FWNX09)、北方民族大学校级一般项目(2021XYZJK01)资助

(multiple traveling salesman problem, MTSP)^[2], 其中有多
个推销员, 他们的路线都是不相交的, 并且必须在同一个节
点开始和结束。MTSP 是更加符合现实生活的一类问题,
广泛存在于车辆路由优化^[3]、应急设施服务区划分^[4]、灾情
巡视路线^[5]等实际问题中。如果有多个初始城市, 该问题
将成为多起点闭回路多旅行商问题 (multiple depots
multiple traveling salesman problem, MMTSP)^[6]。

本文针对 MMTSP 进行研究, 此问题在国内外的研究
相对于 MTSP 来说较少。蚁群优化算法已广泛应用于
MMTSP 研究领域, 并取得了良好的效果。孙魏等^[7]将多
无人机协同侦察建模为 MMTSP, 并利用蚁群算法进行求
解。Seven 等^[8]使用聚类算法将各城市分类, 然后再利用
蚁群算法求最短路径。Liu 等^[9]提出将蚁群算法和遗传算
法相结合的算法, 该算法通过积累最有效的子解, 收敛到最
优的最终解。然而, 由于蚁群优化系统具有许多固有参数,
参数优化困难。这导致了其计算速度慢和局部收敛等缺
点。遗传算法已被许多研究人员应用于 MMTSP。Zhou
等^[10]提出遗传算法和粒子群算法对 MMTSP 进行求解。
胡士娟等^[11]先使用模糊 C 均值聚类方法将所有城市分为
若干类, 然后对每个类采用遗传算法求解。Singh 等^[12]
提出的稳态分组遗传算法并设计了一种新的遗传算子对
MMTSP 问题进行求解。欧阳杰平^[13]提出的新的两段式
染色体设计, 该算法具有更少的解空间, 并且能产生更少的
解。Brown 等^[14]提出一种新的染色体表示方法并与分组
遗传算法相结合对 MMTSP 进行求解。Yuan 等^[15]提出
的一种新的交叉算子, 该算法可以缩小问题搜索空间的大小
以提高解的精度。当遗传算法使用染色体技术来表示问题
的解时, 容易出现解的冗余问题, 最终导致搜索无效和解较
差。Venkatesh 等^[16]提出使用人工蜂群算法和杂草入侵优
化算法对 MMTSP 进行求解, 并通过使用局部搜索算法进
一步改进了人工蜂群算法和杂草入侵优化算法的最优解。
在人工蜂群算法和杂草入侵优化算法的优化过程中, 特别
是在解决高维 MTSP 问题时, 全局搜索能力和本地搜索能
力之间的平衡存在一些限制。

MMTSP 问题提出初期众多学者大多使用精确算法求
解, 尽管该问题的定义简单, 但它的求解却相当困难。随着
点数的增加, 解决问题的时间也相应增加; 传统的启发式算
法和元启发式算法都存在易陷入局部最优和收敛速度慢等
问题, 为克服上述缺点, 本文尝试运用并行处理框架以及混
合多种算法进行优化求解; 通过使用聚类算法将问题简化
为并发处理的 TSP 求解算法, 对于每个类, 将使用模拟退
火算法 (simulated annealing, SA) 确定最短距离; 依托并行
处理框架并发完成 TSP 问题的求解, 将每个类的结果相
加, 以得到 MMTSP 的解。

本文提出的基于 Spark 的并行 k 均值聚类模拟退火算
法 (spark-based parallel K-means clustering simulated
annealing algorithm, SPKSA) 使用 TSPLIB 中的标准数据集

作为实验数据, 通过实验对比 SPKSA 和其他算法的求解质
量和运行时间。实验表明, 与目前较为优秀的算法相比,
SPKSA 对于提升求解精度以及求解效率都有较大提升。

本文贡献:

- 采用分类+并行化思想的 SPKSA 算法可以直接部
署在 Spark 中, 保证该算法面对大规模 MMTSP 问题依然
能够提供较高的求解精度和效率。
- 采用大规模邻域+3-opt 改进的局部搜索算法保证
并行处理的求解精度。
- 针对 MMTSP 的特性, 在模拟退火算法的基础上提
出一种新的退温函数, 改进后的退温函数可以很好的跳出
局部最优解。

1 实验基础

1.1 MMTSP 问题描述

本文研究的是 MMTSP 问题, 该问题可以简单的描述
为: 现有 m 个城市 z_1, z_2, \dots, z_m (每个城市的坐标为 $(a_i,$
 $b_i)$), 这 m 个城市由 k 个旅行商遍历, 并且这 k 个旅行商分
别遍历不同的城市, 最后使所有旅行商总路径之和最短。
即将 $Z = \{z_1, z_2, \dots, z_m\}$ 划分为 k 个非空子集 $\{R_i\}_{i=1}^k$, 需
要找到经过每个子集经过的最短路径, 其中 k 个旅行商要
把剩余的 $m-k$ 个城市完全遍历, 每个城市都只能被一个
旅行商访问, 并且最终都回到自己的出发城市。由于 TSP
已经证明属于 NP-hard 问题^[17], 而 MMTSP 既需要分组优
化, 也需要优化组内的遍历次序, 因此 MMTSP 也属于 NP-
hard 问题。

1.2 Spark 框架

Spark^[18] 是一个基于内存的分布式处理框架。该框架基
于内存计算, 提高了大数据环境下数据处理的实时性, 保证
了高容错性和高扩展性。它允许用户将 Spark 部署在许多
普通的硬件上, 形成集群。Spark 中数据的基本组织形式是
弹性分布式数据集 (resilient distributed dataset, RDD), 采用
RDD 的依赖关系构建有向无环图 (directed acyclic graph,
DAG), 将 DAG 分解成多个阶段进行调度处理。

Spark 集群分为 Master 节点和 Worker 节点, Master
节点协调和调度集群中的所有 Worker 节点; 在每个
Worker 节点中, 都有一些执行器, 它们用于管理工作线程。
当一个算法在 Spark 上运行时, 会在所有 Worker 节点上形
成一个由这些工作线程组成的线程池, 并对算法的输入进
行分区并分配到线程中。最后, Spark 中的 Master 节点在
线程池中分配线程数来并行执行算法, 每个线程都带着部
分输入执行算法。像模拟退火算法这种存在许多次迭代的
算法, 使用 Spark 将迭代中间结果保存在内存中, 大大降低
磁盘 I/O 处理, 提升迭代效率。图 1 即为 Spark 架构图,
Cluster Manager 控制整个集群, 监控 Worker, Driver 运行
Application 的 Main() 函数, Executor 执行器, 是为某个
Application 运行在 Worker Node 上的一个进程。

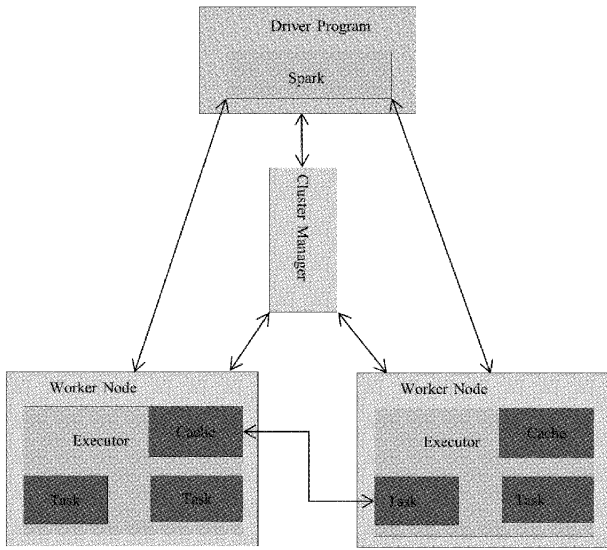


图 1 Spark 架构图

2 SPKSA 算法

本文提出一种将基于 Spark 的 K 均值聚类和模拟退火相结合的智能优化算法。该算法在第一阶段利用基于 Spark 的 K 均值聚类算法并行化处理数据集,根据彼此之间的欧式距离进行分类,将距离近的划分为一类。第二阶段利用模拟退火算法对划分好的每个类进行优化,从而加快求解速度。

2.1 基于 Spark 的 K 均值聚类算法

K-means^[19]是一种无监督学习算法。K-means 的思想很简单。对于给定的数据集,根据样本间的距离将数据集划分为 K 类。让类内的点尽量紧密的连在一起,而让类间的距离尽量的大。基于 Spark 的 k 均值聚类算法将每个样本到中心点距离的计算进行并行化。具体步骤如下:

步骤 1 根据给定的 k 值,然后随机初始化一组聚类中心。

步骤 2 根据欧式距离计算所有样本点到每一个划分中心的距离,并将所有样本点划分到距离最近的划分中心。

步骤 3 计算每个划分中样本点的平均值,并将其作为新的中心。

步骤 4 循环进行步骤一和二直至最大迭代次数,或划分中心的变化小于某一预定阈值,完成 k 均值划分。

例如有 13 个城市需要 4 个旅行商遍历,将 13 个城市按照 {1, 2, ..., 13} 编号,按照上述聚类之后会得到 4 个类,可以设为 {{1, 8, 6}, {4, 9, 10, 13}, {2, 7, 5}, {12, 3, 11}}, 第二阶段只需将各个类进行排序优化即可。

2.2 改进的模拟退火算法

模拟退火^[20]算法的灵感借鉴于固体退火原理,当固体的温度很高的时候,内能相对较大,固体的内部粒子处于快速无序运动,在缓慢降低温度的过程中,固体的内能减小,

粒子慢慢渐趋有序,最终,当固体处于常温时,内能达到最小。传统模拟退火算法存在求解质量低和收敛速度慢等问题并且受温度冷却速率的影响,模拟退火算法中退温函数影响算法的收缩空间,温度高时算法进行全局搜索,温度低时,算法进行局域搜索。温度降温较快算法全局搜索时间过短而无法得到全局最优解,降温过慢算法局域搜索能力较弱影响解的质量,常用的标准退温方式如式(1)所示。

$$T_0 = T_0 \times \alpha \quad (1)$$

式中: T_0 表示初始温度, α 为退温系数, α 的大小决定温度下降的速度。基于这种退温方式,本文针对 MMTSP 的特性,对算法的退温函数进行改进,改进的退温函数如式(2)所示。

$$T = \pi + (T_0 - \pi) / (1 + \exp((t - gapIter) / 8)) \quad (2)$$

式中: T_0 表示初始温度, T 为变化后的温度, t 为迭代次数, $gapIter$ 表示浮动的间隔。选择标准退温和本文退温函数进行效果对比,退温曲线如图 2 所示。

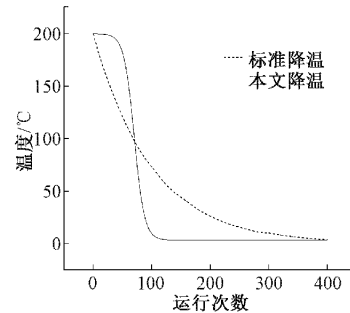


图 2 退温曲线对比图

由图 2 可知,标准降温到达低温状态迭代次数过多,结合 SPKSA 的特点,在高温时使温度下降缓慢以增强全局搜索能力,在低温时为增强模拟退火算法的局部搜索能力加入大规模邻域搜索算法和 3-opt 算子,解的精度得到提高。

1) 大规模邻域搜索

在模拟退火算法产生新解的过程引入大规模邻域搜索算法^[21](large neighborhood search, LNS)。LNS 的思想为:从当前最优解中随机破坏多个节点;根据约束条件将破坏的节点重新插入得到新的解。LNS 实质上是通过交替使用 Destroy 和 Repair 两个方法逐步改善初始解。

Destroy 顾名思义是“毁坏”的意思,Repair 是“修复”的意思。初始解空间是 1-2-3-4-5-6。首先随机选取 $N/2$ 个节点去移除初始解(N 为节点总个数),破坏的城市为 1、4 和 6,剩下的城市按照初始顺序依次排列,即生成的破坏解空间为 2-3-5。然后使用 Repair 方法对 Destroy 后的解进行修复,即对 2-3-5 这个解进行修复,也就是将 1、4 和 6 这 3 个城市重新安排进 2-3-5 当中,这里可以从 1、4 和 6 中随机选择一个城市插入到 2-3-5 中,比如说选择 1 插入到 2-3-5 中,一共有 3 种可能:1-2-3-5、2-1-3-5、2-3-1-5(1-2-3-5 和 2-3-5-1 效果相同,故不重复),从这 3 个可能解中选择一个

最好的,比如说 1-2-3-5 最好,那就选择 1-2-3-5,剩下的 4 和 6 同理,分别插入不同的位置计算适应度值,选取最小的值,即为当前最优解。

2)局部搜索算法 3-opt

3-opt 算法^[22]是一种针对 TSP 问题的局部搜索算法。如图 3 所示,3-opt 通过选取路径中不相邻的 3 个节点之间的连接删除,然后尝试其他 7 种不同连接方式,并计算不同连接方式之后的路径长度,选取路径长度最短的连接方式作为新的连接方式。对于路径中不同的 3 个连接重复多次连接过程,直到所有的 3 个连接都没有新的连接方式。

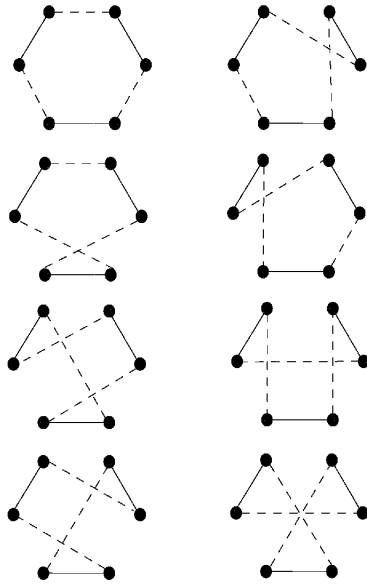


图 3 3-opt 八种连接方式

2.3 SPKSA 算法流程

本文提出一种基于 Spark 的独立试验并行策略,将 K-means 产生的类分配给每个 Slave 节点,由 Slave 节点独立完成算法并得到结果,最后将每个 Slave 的结果相加即为最终结果。此策略依赖用时最长的 Slave 节点,且由于模拟退火算法是一种随机寻优算法,时间复杂度无法估计。此算法和传统算法相比,缩减了算法的搜索空间,并且降低了运行时间。将 K-means 分好的类存入 RDD 中,并运行产生初始解,选取当前最优解使用 LNS 进行扰动生成新的解,最后使用 3-opt 进行优化。

SPKSA 步骤如下:

步骤 1 将城市按照 K 均值聚类算法进行分类;

步骤 2 实例化 RDD 将城市并行传到模拟退火算法;

步骤 3 每个组的城市按照 TSP 根据贪婪策略进行初始化并计算适应度,记录最优距离 Lbest 和最优路径 Pbest;

步骤 4 使用 LNS 和 3-opt 对当前最优路径进行优化,计算适应度值,选取最优路径 Pnew,比较 Pnew 路径和 Pbest 路径适应度值,若 $f(Pbest) \geq f(Pnew)$,则 $Pbest = Pnew$;否则,按照 Metropolis 准则进行更新,并根据退温函

数更新温度;

步骤 5 判断是否到达结束条件,没有达到则循环执行步骤 4,直至到达迭代次数;

步骤 6 输出各个组的最优解;

步骤 7 将各个组的最优解相加作为 MMTSP 的最优解。

3 算法性能测试

本文实验数据来自 TSPLIB(<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>),TSPLIB 是 TSP 和相关问题的示例实例库。为了测试 SPKSA 算法的性能,本文采用 Spark2.4.0、Intel Core i7-9700 CPU@3.00 GHz×5 以及 Python 对 TSPLIB 数据库中标准数据集作为实验数据进行了仿真实验。表中最优解、平均解和平均时间分别用 BEST、AVG 和 Time(s)表示且同一 K 值下的最优解已加粗。

3.1 求解质量对比

选取 TSPLIB 数据集 kroA100、pr299、pcb299、rRat575 分别在 K 等于 5 和 10 时进行实验对比,对 SPKSA 和 IPGA、FCMPGA 进行对比。3 种算法对 4 个实例独立运行 30 次,求解质量结果如表 1 所示。

表 1 3 种算法实验结果对比

数据集	算法	AVG	BEST	
kroa100	K=5	SPKSA	23 005	22 245
		FCMPGA	24 369	23 677
	K=10	IPGA	37 051	30 933
		SPKSA	22 863	22 143
		FCMPGA	23 306	22 859
		IPGA	39 159	33 535
pr299	K=5	SPKSA	51 883	49 669
		FCMPGA	54 644	54 016
	K=10	IPGA	164 544	154 819
		SPKSA	52 887	51 730
		FCMPGA	55 240	54 080
		IPGA	190 220	180 920
Pcb442	K=5	SPKSA	53 169	52 925
		FCMPGA	57 855	57 034
	K=10	IPGA	222 880	211 284
		SPKSA	54 936	54 809
		FCMPGA	56 254	55 800
		IPGA	247 448	240 635
rRat575	K=5	SPKSA	7 021	6 959
		FCMPGA	8 057	7 852
	K=10	IPGA	34 712	33 049
		SPKSA	7 121	7 080
		FCMPGA	7 448	7 393
		IPGA	36 555	36 060

表 1 列出来 3 种算法对 4 个实例独立运行 30 次的测试结果。由表 1 的实验结果中,在各个规模的问题中,SPKSA 的平均解都是相对较好的,说明 SPKSA 将各个类作为 TSP 求解后,能对各个组的搜索空间充分探索,从而

求得满足目标函数的最优解。

选取 TSPLIB 数据集 Eil51、Kroa100、Kroa150 分别在不同 K 值时进行实验对比,对于基准测试,将 SPKSA 与各种以前的提出的算法进行了比较,求解质量结果如表 2 所示。

表 2 8 种算法在 3 个数据集上的统计结果

数据集	Num	K	GA1C ^[14]	GA2PC ^[13]	GGA-SS ^[12]	ACO ^[9]	TCX ^[15]	ABC ^[16]		IWO ^[16]		SPKSA	
								AVG	BEST	AVG	BEST	AVG	BEST
Eil51	1	3	529	543	449	448	492	448	446	448	446	445	435
	2	5	564	586	479	478	519	475	472	478	472	466	450
	3	10	801	723	584	584	670	581	580	583	581	455	435
Kroa100	4	3	27 036	26 653	22 051	22 619	26 130	21 814	21 798	21 941	21 798	21 773	21 426
	5	5	29 753	30 408	23 678	24 166	28 612	23 221	23 182	23 319	23 294	23 005	22 245
	6	10	36 890	31 227	28 488	27 890	30 988	27 004	26 961	27 072	26 961	22 840	22 143
	7	20	62 471	54 700	40 892	39 949	44 686	38 397	38 333	38 357	38 245	22 346	21 329
Kroa150	8	3	46 111	47 418	38 434	39 247	44 674	38 262	38 066	38 055	37 957	27 571	26 656
	9	5	49 443	49 947	39 962	40 647	47 811	39 202	38 979	38 881	38 714	28 246	27 471
	10	10	59 341	54 958	44 274	44 436	51 326	42 712	42 441	42 462	42 234	28 897	28 280
	11	20	94 291	73 934	56 412	55 980	62 400	53 877	53 603	53 612	53 475	29 081	28 033
	12	30	131 503	99 547	72 783	71 266	78 023	69 045	68 865	68 751	68 541	28 076	27 260

表 2 列出了 8 种算法在 3 个实例的测试结果,表中 AVG 代表平均求解结果,BEST 表示最优求解,表中在同一 K 值下的最优解以加粗。表 2 的实验结果显示,在各个规模不同 K 值的问题中,SPKSA 的平均解和最优解都是更好的。其他算法在同一规模下随着 K 值的增大求解精度大大降低,而 SPKSA 因为在模拟退火算法中加入局部搜索算法可以进一步求得最优解,并且使用 Spark 对算法进行并行化使得当 K 值越大时模拟退火算法的搜索空间反而更小,即使 SPKSA 在同一规模下不同 K 值的求解精度基本不变。

表 3 总结了表 2 中各数据集的统计结果,提供了 3 个效果较好的算法与其他算法的对比结果,<、=、>表示 3 个算法相比于其他算法在 12 个实例中表现更好、相同和更差的数量。表 2~3 清楚的表明,SPKSA 在 3 个数据集上都优于以前提出的方法。

表 3 实例汇总

算法	ABC			IWO			SPKSA		
	<	=	>	<	=	>	<	=	>
GA1C	12	0	0	12	0	0	12	0	0
GA2PC	12	0	0	12	0	0	12	0	0
GGA-SS	12	0	0	12	0	0	12	0	0
ACO	12	0	0	10	2	0	12	0	0
TCX	12	0	0	12	0	0	12	0	0
ABC	—	—	—	6	1	5	12	0	0
IWO	5	1	6	—	—	—	12	0	0
SPKSA	0	0	12	0	0	12	—	—	—

图 4~6 以图形方式比较了 12 个实例上 GA1C、GA2PC、GGA-SS、ACO、TCX、ABC、IWO 和 SPKSA 的总距离。这 3 张图也是利用表 3 的结果绘制的,这些数字明确地重申了 SPKSA 优于其他方法。从这些图中可以清楚地观察到,在比较的所有八种方法中,GA1C 和 GA2PC 表现最差,SPKSA 表现最好。

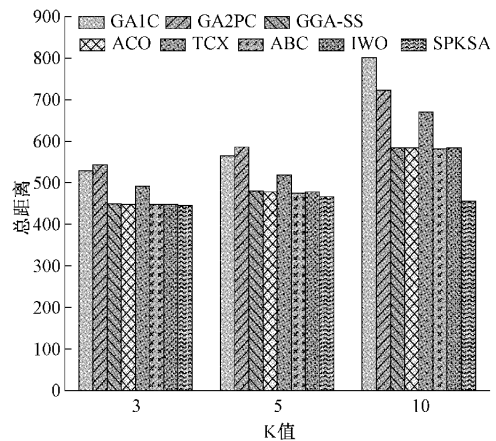


图 4 8 种算法在 Eil51 中的比较

3.2 求解速度对比

为了测试 SPKSA 算法的求解速度,本文在同一实验环境下实现了 3 种并行的算法:SPKSA、AGACO 和 KMACO,分别使用 TSPLIB 数据集下的 6 个实例中对 3 种算法独立运行 30 次,对比求解质量和求解时间。

表 4 列出了 AGACO、KMACO 和 SPKSA 对 4 个实

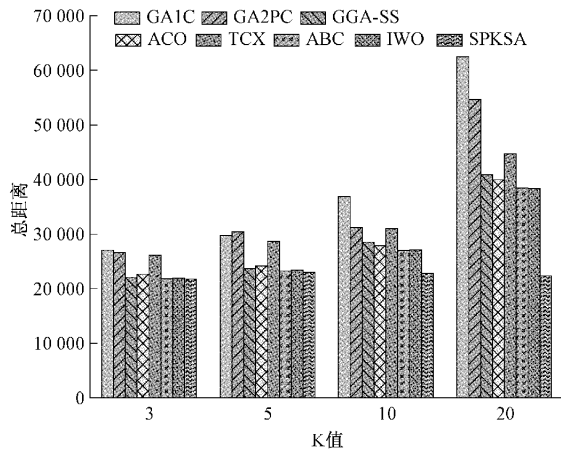


图 5 8 种算法在 Kroa100 中的比较

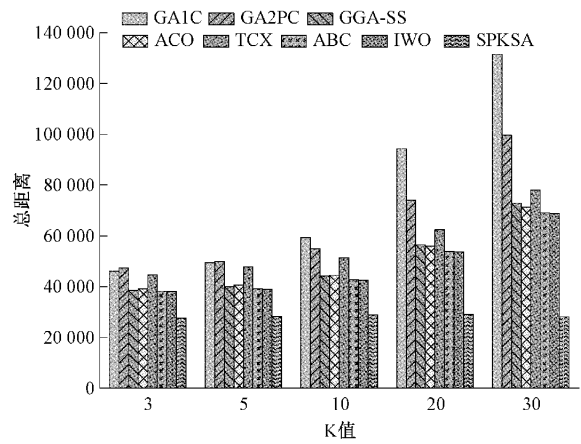


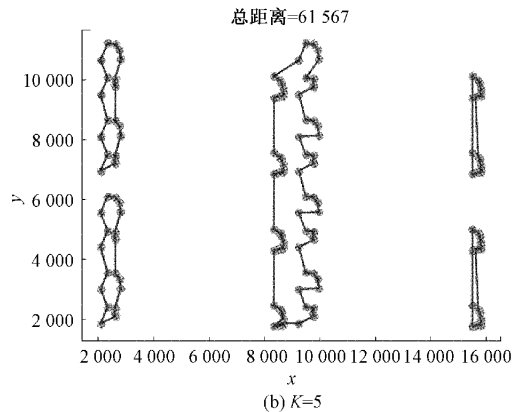
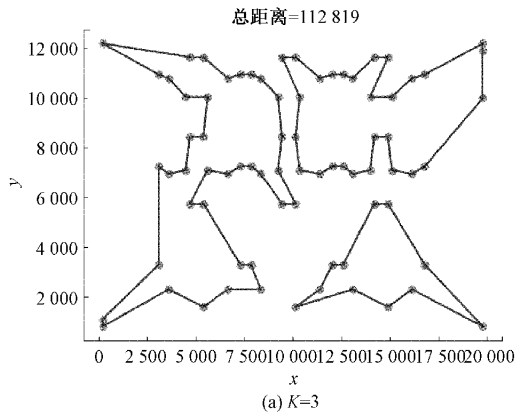
图 6 8 种算法在 Kroa150 中的比较

表 4 3 种算法独立运行 30 次的统计结果

数据集	K 值	AGACO			KMACO			SPKSA		
		Best	Avg	Time/s	Best	Avg	Time/s	Best	Avg	Time/s
Pr76	3	118 294	119 164	194.3	114 197	116 695	177.5	112 819	115 595	20.57
	5	116 597	118 948	61.7	116 434	119 353	52.67	115 912	118 787	12.69
	10	109 959	113 630	39.7	113 509	116 968	35.7	107 776	110 733	8.47
Pr152	5	64 565	65 569	371.5	62 759	65 398	382.6	61 567	62 998	360.6
	10	54 974	59 232	103.7	54 974	59 761	105.3	54 718	57 332	47.4
	20	47 975	49 338	58.2	48 319	49 869	66.8	47 870	49 005	12.4
Pr299	20	53 499	54 388	282.4	53 499	54 160	181.1	52 695	53 973	122.2
	30	53 499	54 451	145.6	53 359	54 481	137.6	52 909	54 003	39.6
	40	53 446	54 242	128.3	53 446	53 900	106.9	52 637	53 801	22.8
Pr439	20	108 657	110 476	624.8	109 456	110 764	662.8	108 212	110 304	581.5
	30	103 868	107 042	362.5	106 330	108 385	309.8	103 252	106 328	305.7
	40	104 059	105 220	245.1	104 159	115 422	244.9	103 661	105 045	85.2

例独立运行 30 次的测试结果,3 种算法都是先经过聚类,在对各个类进行单独的 TSP 问题求解,说明 SPKSA 在求解单独 TSP 任务是精度更高。由表 4 结果可知,在 4 个实例上 SPKSA 的结果都是最好的,因为 SPKSA 先使用聚类算法将一个大规模的数据集划分成很多小数据集,

使得单独的模拟退火算法问题搜索空间减小,并且使用局部搜索算子对模拟退火算法产生的解进一步优化。选取 pr76、pr152、pr299、pr439 数据集分别在 K 值等于 3、5、20、30 时的最优路径图,图中路径清晰没有交叉,如图 7 所示。



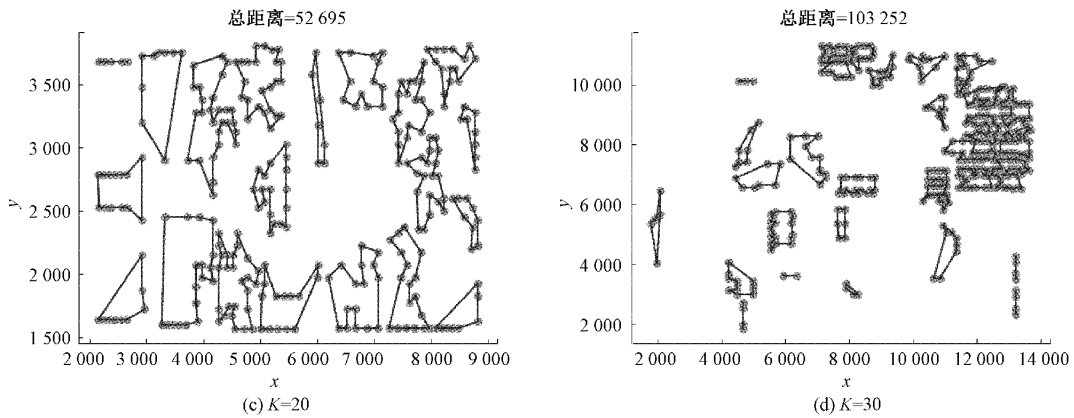
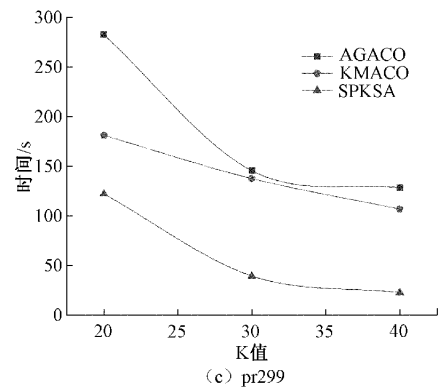
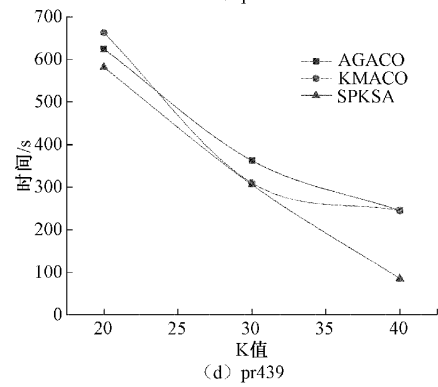


图 7 SPKSA 最优路径图

由表 4 的实验结果可知,在各个规模上 SPKSA 的平均解都是更好的,说明 SPKSA 相对于另外两种算法能更充分的对搜索空间进行探索。由于 SPKSA 使用 Spark 来求解 MMTSP,在 K 值较少时,能够划分成并行处理的子任务数较少,求解速度比较慢,但当 K 逐渐增大时,并发处理的优势会大幅提高求解速度。图 8 为 3 种算法在 4 个实例中平均运行时间变化曲线。当 K 值较小时单个节点处理的点量相对较多,模拟退火算法问题搜索空间太大,由于为提高模拟退火算法在单个 TSP 的求解精度,引入局部搜索算子,使得模拟退火算法在处理较大数据量时效率较慢。当 K 值越来越大时,由于依托并行处理框架并发完成 TSP 问题的求解,使得每个模拟退火算法问题搜索空间大大减小,因此当 K 值越来越大时的求解时间逐渐降低。在

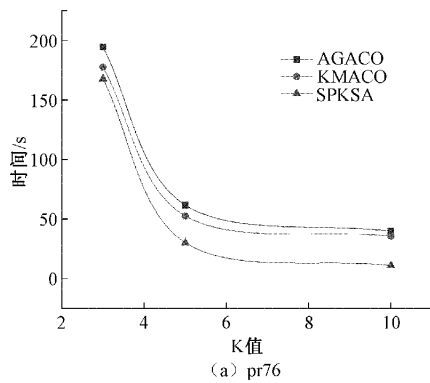


(c) pr299

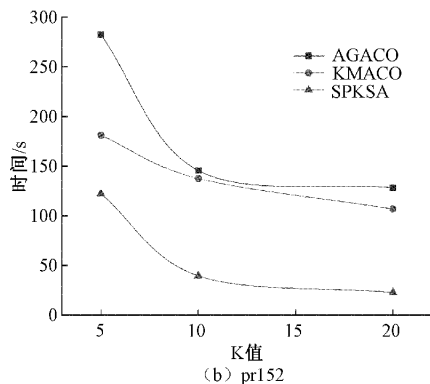


(d) pr439

图 8 3 种算法在 4 个实例上的平均运行时间变化曲线



(a) pr76



(b) pr152

实际问题中,需要的 K 值往往很大,本文设计的 SPKSA 能够很快且较为精准的解决问题。

4 结 论

本文针对求解 MMTSP 提出了一种基于 Spark 的并行 k 均值聚类模拟退火算法 SPKSA,通过实验发现该算法利用 k 均值聚类对城市进行分类,减少了搜索空间。在模拟退火算法的基础上,修改了退温函数,提升了算法的搜索能力,结合了 3-opt 和 LNS 局部搜索策略,加强算法局部搜索性能,使得每个小的搜索空间都能得到充分探索。该算法利用 spark 计算框架并行计算使得将 MMTSP 简化

为 TSP 的每个类并行计算,大大缩短了整体的求解时间。

如何将其他聚类算法与模拟退火算法求解 MMTSP 以及如何利用 spark 框架使求解 MMTSP 的算法进一步并行化来减少求解时间还需进一步研究。

参考文献

- [1] 孙鉴,刘淞佐,武晓晓,等.基于 Spark 的并行模拟退火算法求解 TSP[J].电子测量技术,2022,45(4):53-58.
- [2] BEKTAS T. The multiple traveling salesman problem: an overview of formulations and solution procedures[J]. Omega, 2006, 34(3): 209-219.
- [3] 王沛栋,唐功友,李扬.带容量约束车辆路由问题的改进蚁群算法[J].控制与决策,2012, 27(11): 1633-1638.
- [4] 赵星,吉康,申珂.基于多旅行商问题的应急设施服务区划分模型[J].交通运输系统工程与信息,2020, 20(5):205-211.
- [5] 王庆波,孙宇涵,杨阳,等.最佳灾情巡视路线[J].哈尔滨工业大学学报,1999(2):14-17.
- [6] BENAVENT E, A MARTINEZ. A polyhedral study of the Multi-Depot Multiple TSP [J]. Annals of Operations Research, 2013, DOI: 10.1007/s10479-011-1024-y.
- [7] 孙魏,伍度志.基于防御雷达监控的多无人机协同侦察优化模型[J].兵工自动化,2018,37(4):5-9.
- [8] STEVEN A, HERTONO G, HANDARI B. Implementation of clustered ant colony optimization in solving fixed destination multiple depot multiple traveling salesman problem [C]. 2017 1st International Conference on Informatics and Computational Sciences(ICICoS), 2018.
- [9] LIU W, LI S, ZHAO F, et al. An ant colony optimization algorithm for the multiple traveling salesman problem [C]. 4th IEEE Industrial Conference on Industrial Electronics and Applications (ICIEA 2009), IEEE, 2009: 1533-1537.
- [10] ZHOU H L, SONG M L, WITOLD P. A comparative study of improved GA and PSO in solving multiple traveling salesman problem [J]. Applied Soft Computing, 2018, 64: 564-580, DOI: 10.1016/j.asoc.2017.12.031.
- [11] 胡士娟,鲁海燕,向蕾,等.求解 MMTSP 的模糊聚类单亲遗传算法[J].计算机科学,2020,47(6):219-224.
- [12] SINGH A, BAGHEL A S. A new grouping genetic algorithm approach to the multiple traveling salesman problem[J]. Soft Computing, 2009, 13(1): 95-101.
- [13] 欧阳杰平.使用遗传算法解决 MTSP 问题的一种新的染色体设计[J].舰船电子工程,2006(3):107-109.
- [14] BROWN E C, RAGSDALE C T, CARTER A E. A grouping genetic algorithm for the multiple traveling salesman problem [J]. International Journal of Information Technology & Decision Making, 2007, 6(2):333-347.
- [15] YUAN S, SKINNER B, HUANG S, et al. A new crossover approach for solving the multiple travelling salesman problem using genetic algorithms [J]. European Journal of Operational Research, 2013, 228: 72-82, DOI: 10.1016/j.ejor.2013.01.043.
- [16] VENKATESH P, SINGH A. Two metaheuristic approaches for the multiple traveling salesman problem [J]. Applied Soft Computing, 2015, 26: 74-89, DOI: 10.1016/j.asoc.2014.09.029.
- [17] 王勇臻,陈燕,于莹莹.求解多旅行商问题的改进分组遗传算法[J].电子与信息学报,2017,39(1):198-205.
- [18] 吴信东,嵇圣础. MapReduce 与 Spark 用于大数据分析之比较[J].软件学报,2018,29(6):1770-1791.
- [19] 袁方,周志勇,宋鑫.初始聚类中心优化的 K-means 算法[J].计算机工程,2007(3):65-66.
- [20] 王雪梅,王义和.模拟退火算法与遗传算法的结合[J].计算机学报,1997(4):381-384.
- [21] 李妍峰,李军,高自友.大规模邻域搜索算法求解时变车辆调度问题[J].管理科学学报,2012,15(1):22-32.
- [22] MISEVICIUS A. Combining 2-OPT, 3-OPT and 4-OPT with K-SWAP-KICK perturbations for the traveling salesman problem [C]. 17th International Conference on Information and Software Technologies, 2011.

作者简介

孙鉴,工学博士,讲师,主要研究方向为大数据存储与管理等。

E-mail: 2014132@nun.edu.cn

李昊,硕士研究生,主要研究方向为大数据分析 with 知识工程等。

E-mail: 419253120@qq.com

刘淞佐,硕士研究生,主要研究方向为大数据分析 with 知识工程等。

E-mail: a_bacd@qq.com

刘品,硕士研究生,主要研究方向为大数据分析 with 知识工程等。

E-mail: 1257224069@qq.com

陈攀,硕士研究生,主要研究方向为大数据分析 with 知识工程等。

E-mail: 934926124@qq.com

雷冰冰,工学博士,讲师,主要研究方向为物联网 with 嵌入式技术等。

E-mail: x_generation@163.com