

最近邻算法、模拟退火算法与禁忌搜索算法的简单优化与对比

寇凯淇¹⁾ 邱晓鹏¹⁾ 齐泳博¹⁾

¹⁾(东北大学 计算机科学与工程学院, 辽宁 沈阳 110819)

摘 要 TSP 问题是典型的组合优化问题, 有诸多种解决问题的算法。笔者采用了最近邻算法、模拟退火算法和禁忌搜索算法对规模在 2400 以内的对称 TSP 问题进行求解实验, 优化算法并分析对比实验结果。实验表明, 适当加大步长、增加迭代次数有利于启发式算法的求解, 两种启发式算法在小规模数据中优于最近邻算法, 但在面对大规模数据会出现精度低或时间开销大的问题。

关键词 算法; TSP 问题; 最近邻算法; 模拟退火算法; 禁忌搜索算法; 启发式算法

中图法分类号 ***** DOI 号 *投稿时不提供 DOI 号* 分类号

Simple optimization and comparison of Nearest neighbor algorithm, Simulated annealing algorithm and Tabu search algorithm

Kou Kai-Qi¹⁾ Qiu Xiao-Peng¹⁾ Qi Yong-Bo¹⁾

¹⁾(School of Computer Science and Engineering, Northeastern University, Shenyang, 110819)

Abstract Traveling salesman problem(TSP) is a classical combinatorial optimization problem, which has many algorithms to solve. The author do experiments that using Nearest neighbor algorithm, Simulated annealing algorithm and Tabu search algorithm to solve the Symmetric TSP with scale less than 2400, optimize the algorithm and analyze the results of the experiments. The experiments shows that those two Heuristic algorithm do better than Nearest neighbor algorithm when the scale of data is small and do worse in accuracy or time cost when the scale is too big.

Key words algorithm; TSP Problem; Nearest Neighbor; Simulated annealing; Tabu search; Heuristic

1 引言

旅行商问题(TSP)是一种十分典型的组合优化问题,其问题的没描述为:给定 N 个城市 $C = \{C_1, C_2, C_3, \dots, C_n\}$, 其中任意两个城市的距离记为 $d(C_i, C_j)$,

求一条经过集合 C 中所有城市一次且仅一次的路径, 最终返回起点, 使得闭合路径 S 的路径长度最小。

TSP 问题描述简单, 但其本身是十分著名的 NP 难问题。TSP 问题在很多实际问题中都有应用, 如管路铺设、路径规划以及网络、分配等优化问题中

收稿日期: 年-月-日; 最终修改稿收到日期: 年-月-日 *投稿时不填写此项* 本课题得到……基金中文完整名称(No.项目号)、……基金中文完整名称(No.项目号)、……基金中文完整名称(No.项目号)资助.作者名1(通信作者), 性别, xxxx年生, 学位(或目前学历), 职称, 是/否计算机学会(CCF)会员(提供会员号), 主要研究领域为****、****.E-mail: ****.作者名2(通信作者), 性别, xxxx年生, 学位(或目前学历), 职称, 是/否计算机学会(CCF)会员(提供会员号), 主要研究领域为****、****.E-mail: ****.作者名3(通信作者), 性别, xxxx年生, 学位(或目前学历), 职称, 是/否计算机学会(CCF)会员(提供会员号), 主要研究领域为****、****.E-mail: ****.(给出的电子邮件地址应不会因出国、毕业、更换工作单位等原因而变动。请给出所有作者的电子邮件)

第1作者手机号码(投稿时必须提供, 以便紧急联系, 发表时会删除): ……E-mail: ……*此部分6号宋体*

有着广泛的应用。针对 TSP 问题,当前已有多种算法进行解决,本文将选取最近邻算法、模拟退火算法和禁忌搜索算法进行数据的分析与实验,分析算法性能,比较近似算法与启发式算法的优劣性。

2 相关概念与工作

2.1 最近邻算法

最近邻算法是一种近似算法,是线路构造法中最简单的一种。该方法的本质是一种贪心的思想。最近邻算法所求得解通常并不是真正的最优解,而是一个近似解。单次执行最近邻算法的时间复杂度为 $O(n^2)$ ^[1]。

2.2 模拟退火算法

模拟退火算法是一种基于概率并用于近似求解最优化问题的著名方法。它的思想最早在 1953 年由 N. Metropolis^[2]提出,在 1983 年被 Kirkpatrick 等人成功引入组合优化领域。模拟退火算法借鉴了固体退火的机制,通过模拟物理退火过程实现搜索最优解。由于算法的随机性,模拟退火算法可以有有效的跳出局部最优解,从而逼近全局最优解,适用于各种组合优化问题^[3]。模拟退火算法的时间复杂度取决于退火参数的设置,算法的效能同样受限于此。

2.3 禁忌搜索算法

禁忌搜索算法是一种基于局部邻域搜索算法优化而来的元启发式算法,由 Glover 在 1986 年首次提出^[4],于 1989 年正式成立^{[5][6]}。M. Malek; M. Huruswamy; H. Owens; M. Pandya 在 1989 年将其引入到旅行商问题的研究上。

3 算法设计

3.1 最近邻算法

3.1.1 设计思想

最近邻算法采用贪心的策略,每次搜索时只考虑局部的最优解,即寻找距离当前城市最近的城市作为下一个访问的城市,直到所有城市均被访问一次后,返回起点。不同的起点对搜索的路径也存在影响,所以分别将各个城市当作起点,多次执行程序,取最优解。

3.1.2 伪代码

- (1) 将第一座城市设为起点。
- (2) 寻找距离当前城市最近的城市。

(3) 如果未发现符合要求的城市 执行(5), 否则执行(4)。

(4) 访问找到的城市,记录访问信息,计算路径长度,执行(2)。

(5) 返回起点,计算路径长度。

(6) 程序结束。

3.1.3 流程图

如图 1 所示

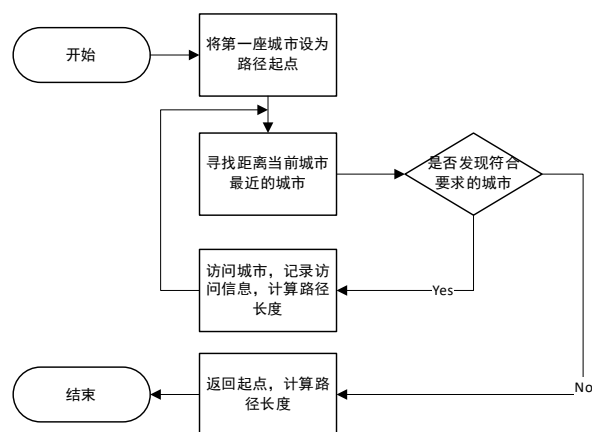


图 1 最近邻算法流程图

3.2 模拟退火算法

3.2.1 设计思想

模拟退火算法模拟物理退火过程。退火是一种物体实现逐渐降温过程的物理现象。对于一个处于非晶体状态的物体,先将该物体加温至一个充分搞得温度,然后对物体进行缓慢降温,在最终阈值温度时,物体达到粒子排列有序的内能最低的结晶状态。

模拟退火算法设定当前温度 T ,初始温度 T_0 ,终点温度 T_{END} ,退火速度 D ,同温度下迭代次数 L 。算法不断寻找新解,若新解更优则更新最优解,否则

以一定概率 $e^{-\frac{\Delta E}{T}}$ 接受新解。在相同温度下达到迭代次数 L ,执行降温操作 $T = T \times D$,直到 $T < T_{END}$ 结束搜索。

3.2.2 参数设置

- (1) 初始温度 $T_0 = 1 \times 10^3$
- (2) 终点温度 $T_{END} = 1 \times 10^{-4}$
- (3) 退火速度 $D = 0.97$
- (4) 迭代次数 $L = 200$
- (5) 退火算法执行次数 $Times = 2$

3.2.3 伪代码

- (1) 随机生成初始解。
- (2) 设置初始温度。

- (3) 随机生成新解。
- (4) 新解更优秀则执行(6), 否则执行(5)。
- (5) 判断概率接受执行(6), 否则执行(7)。
- (6) 更新解。
- (7) 达到迭代次数, 执行(8), 否则执行(3)。
- (8) 达到跳跃降温阈值执行(9), 否则执行(10)。
- (9) 跳跃式降温, 执行(11)。
- (10) 常规降温, 执行(11)。
- (11) 温度低于最终温度执行(12), 否则执行(3)。
- (12) 达到退火次数执行(13), 否则执行(2)。
- (13) 程序结束。

3.2.4 流程图

如图 2 所示。

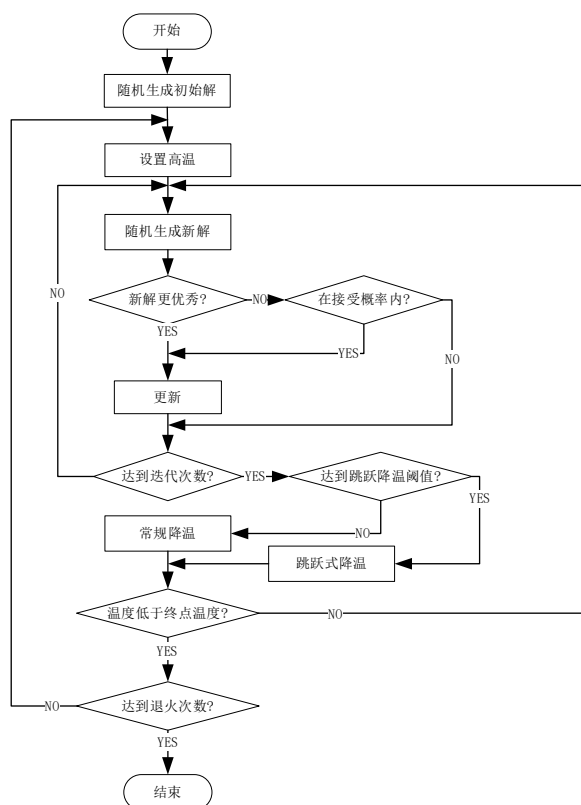


图 2 模拟退火算法流程图

3.3 禁忌搜索算法

3.3.1 设计思想

禁忌搜索算法是在局部邻域搜索算法的基础上, 通过禁忌表记录下已经到达过的局部最优解, 在下次搜索中利用禁忌表不再或有选择的搜索这些点, 避免重复搜索, 以此跳出局部最有解, 找到全局最优解。并且有特赦规则赦免一些被禁忌的优良状态, 以避免最优解被禁忌。虽然时间上没有改进, 但在解的质量上大幅提高。

3.3.2 禁忌表

禁忌表是记录曾经搜索过的状态集合的结构, 有助于算法跳出局部最优解, 到达全局最优解。记录的对象有不同的选择, 例如记录状态, 记录状态转移的方式等等。本文选取了记录状态。禁忌表长度即是被禁忌的对象不允许选取的迭代步数。常有定长、变长等。

3.3.3 特赦规则

为避免将最优解禁忌而无法找到设立的规则。满足该规则时可以忽略禁忌表的影响。

3.3.4 评价函数

反映一个解优劣的函数。在对称 tsp 问题中, 一条路径确定下来, 其距离是确定的, 且是最后优化的目标。本文直接选取了路径的长度作为评价函数。

3.3.5 邻域

从当前解出发, 通过状态转移函数能够到达的解, 称为当前解的邻域。

3.3.6 伪代码

(1) 生成初始解, 禁忌表置空, 最优解、当前解置为初始解。

(2) 是否满足终止条件? 满足则输出最优解, 否则执行(3)。

(3) 寻找当前解的邻域中符合要求的最优解。

(4) 是否满足特赦规则? 满足执行(5), 否则执行(6)。

(5) 用邻域最优解更新最优解, 并执行一定的特赦操作。

(6) 邻域最优解加入到禁忌表, 当前解置为邻域最优解。

(7) 当前解优于最优解, 则更新最优解。

(8) 执行(2)。

3.3.7 流程图

如图 3 所示。

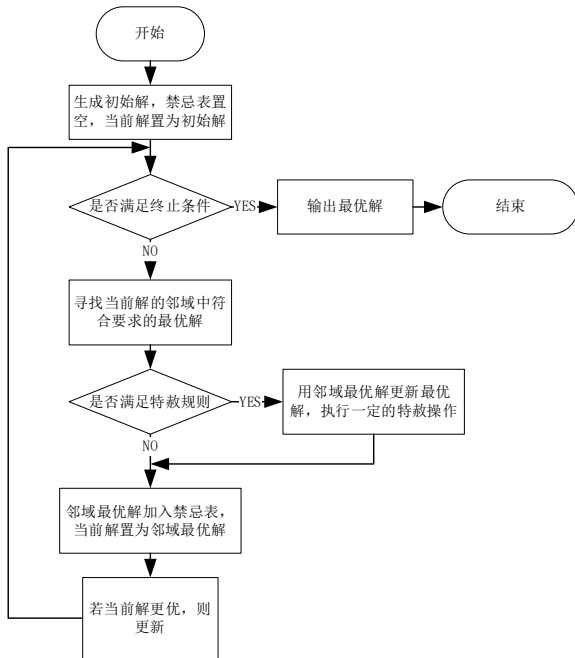


图3 禁忌搜索算法流程图

4 实验环境

实验环境为 Window10 系统, GTX1060 显卡, 6GB 显存, Intel i7-8750H CPU, 8GB 内存。

5 实验过程与分析

5.1 实验过程

实验选取了数据规模在 2400 以内的共 68 条对称 TSP 问题数据集进行运行测试, 并根据数据测试结果对算法进行简单优化, 对比分析算法的性能以及算法之间的优缺点。实验测试精度与时间开销的完整数据如图 4、图 5 所示。

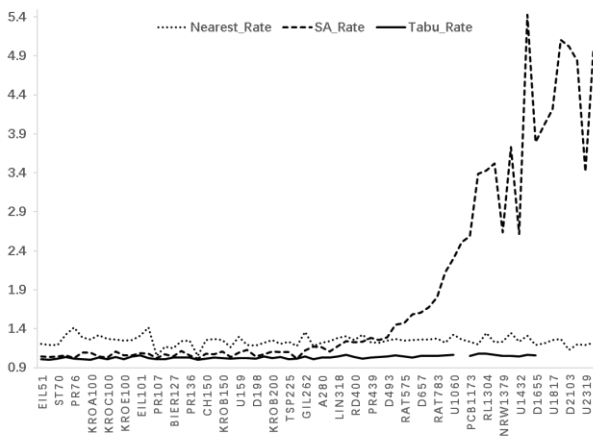


图4 实验精度测试结果

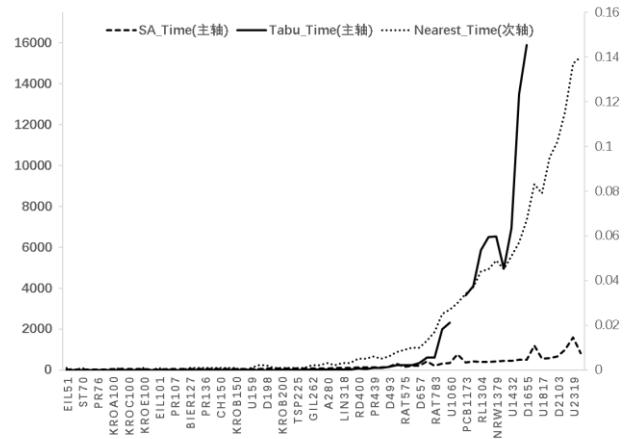


图5 实验时间开销测试结果

5.2 算法优化分析

5.2.1 模拟退火算法的优化

模拟退火算法优化了生成新解的变换函数, 将任意两点间序列逆序插入作为变换函数, 增加临近解的选取范围。加入跳跃式降温, 即在同温度下, 迭代成功次数达到一定阈值时, 进行一次较大范围的降温操作, 用以降低算法的迭代次数。算法也执行二重降温优化操作, 将第一次退火的解重新回到高温状态下进行新一轮迭代。优化前后性能对比如图 6 所示。观察结果得出, 在求解规模为小于 225 的对称 TSP 问题时, 优化后的模拟退火算法结果精度约为 1.07, 优于未优化算法的 1.70, 优化算法方差约为 0.0008, 优于未优化算法的 0.1781。

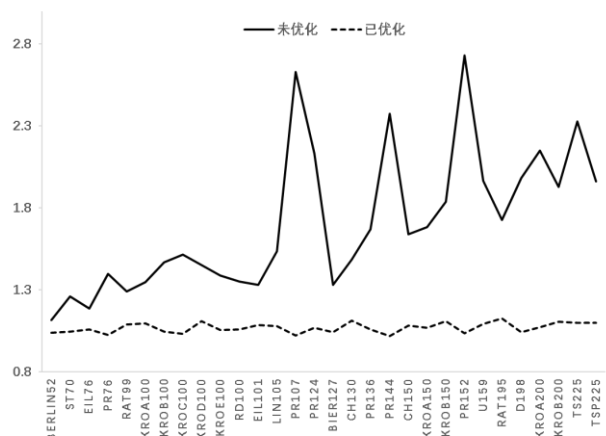


图6 模拟退火算法优化结果对比

5.2.2 禁忌搜索算法的优化

禁忌搜索算法使用两点间序列逆序插入(以下简称为两点反转)的方式作为变换函数进行优化。同时根据输入数据的规模, 动态调整禁忌表的长度, 达到强制收敛的目的。算法执行多起点搜索操

作,进一步优化结果。实验对采用两点交换的变换函数、采用动态禁忌表且两点反转的变换函数以及固定长度为 50 的禁忌表且两点反转的变换函数三种处理方式进行比较,比较结果如图 7 所示。分析结果可知,采用两点交换作为变换函数的结果精度平均为 1.57,相比其他两种处理方式的结果更差。其他两种处理方式均采用两点反转的变换函数,结果精度总体相近,约为 1.025,但采用动态紧急表长度的处理方式在规模大于 200 的数据时,结果优于采用固定禁忌表长度的处理方式。

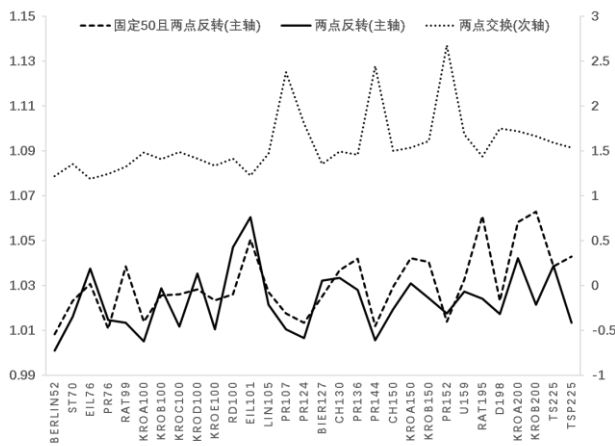


图 7 禁忌搜索算法优化结果对比

5.3 算法性能分析

5.3.1 最近邻算法

分析图 4 数据可得,最近邻算法算法的结果精度没有出现随数据规模增大而下降的情况,测试数据结果分布在 1.05 到 1.42 之间。在时间开销方面,最近邻算法的时间复杂度为 $O(n^2)$,如图 5 所示,在规模为 2400 以内的测试数据中,时间开销小于 0.14 秒。

5.3.2 模拟退火算法

由于实验采用的模拟退火算法采用的是固定参数设置,并未根据输入数据规模而调整参数大小。如图 4 所示,该模拟退火算法在小规模数据(规模小于 500)时,精度表现尚可,约为 1.103。但在更大规模的测试数据中,由于设置的固定参数过小,迭代次数不足以支持算法跳出局部最优解,导致算法精度随着规模的增大而迅速降低,在规模为 2392 的测试数据下精度已超过 5,即无法求解。模拟退火算法的时间复杂度与算法参数设置以及变换函数的时间复杂度相关。设数据规模为 n 、算法中使用的变换函数时间开销为 $f(n)$,则模拟退火算法的时间复杂度为 $T(n) = L \times \log_D(T_{END}/T_0) \times f(n)$,

即 $O(n)$ 。

5.3.3 禁忌搜索算法

分析图 4 数据可知,禁忌搜索算法的结果精度优秀,保持在 1 到 1.08 之间,最优时可以达到 1.0001 的精度。由于实验使用的禁忌搜索算法采用了动态调整禁忌表长度的优化方式,其精度虽然随规模增大而降低,但由于迭代次数也随之增加,使精度下降十分缓慢。分析图 5 数据可知,禁忌搜索算法随着规模的增大呈指数级增长。规模为 1577 的数据花费了近 3.75 个小时进行计算。保持精度需要更多的迭代次数,导致了时间开销的迅速增长,这是动态调整禁忌表长度的禁忌搜索算法的巨大弊端。

5.4 算法对比分析

分析图 4、图 8 实验结果可知,数据规模约在 0-225 时,模拟退火、禁忌搜索算法的精度优于最近邻算法。模拟退火算法约优于最近邻算法 10%-25%,禁忌搜索算法约优于最近邻算法 10%-27%,部分数据两者均可到达 40%以上的优化。随着数据规模增加,模拟退火算法的精度下降明显。约在 450 规模,模拟退火的精度已经低于最近邻精度了。禁忌搜索的精度上升幅度较小。最近邻算法则没有呈现出与数据规模有关的特征,保持稳定。

分析图 5、图 9 可知,模拟退火算法时间呈现线性增长趋势,在数据规模较小时需要 10 秒到 70 秒,耗时偏长。在 1000 及以上的数据规模时,逐步增加,于 2400 规模时达到 1000 多秒耗时,在可接受范围内。禁忌搜索的时间消耗增长较快。在 200 数据规模左右耗时不到 10 秒,在 600 数据规模时超过模拟退火,在测试 d1655. tsp 数据集时消耗了近 16000 秒。更大的数据规模的时间消耗已经难以进行测试,故本文禁忌搜索算法只测试到了 d1655. tsp。最近邻算法在时间消耗上有明显优势,测试的全部数据集均未超过 1 秒。

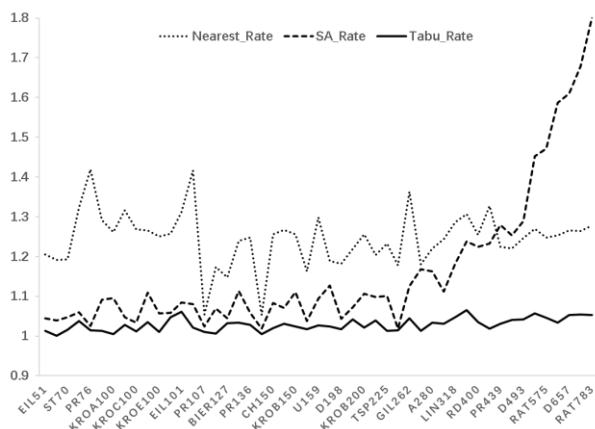


图8 实验精度测试结果(规模800以内)

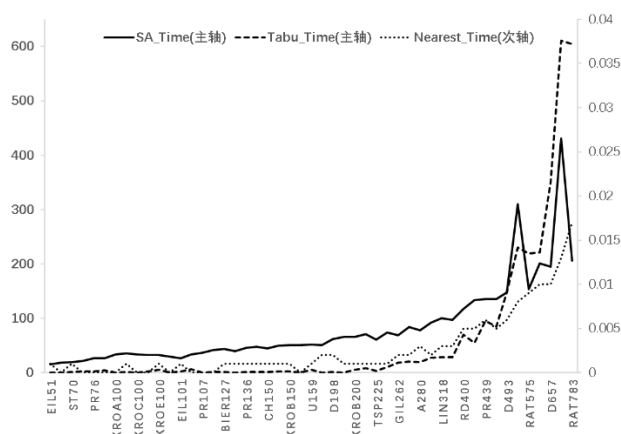


图9 实验时间测试结果(规模800以内)

6 实验结论

(1) 适当加长步长、增加迭代次数有利于启发式算法的求解

(2) 最近邻算法结果分布稳定,在本次实验测试数据中精度上限为 1.05,下限为 1.42,精度与数据规模之间没有发现直接关系。在小规模数据上,精度不及启发式算法,大规模数据上精度不及动态参数的启发式算法。但时间开销,相较而言始终保持很少。

(3) 两种启发式算法精度与时间开销随着数据规模的增大而增大。在小规模数据上,启发式算法明显优于最近邻算法。但在大规模数据上,无动态参数的启发式算法(如本次使用的模拟退火算法)精度与时间开销均劣于最近邻算法(阈值约为 450)。动态调整参数的启发式算法(如本次使用的禁忌搜索算法)精度依旧保持优秀,但时间开销极大。

致 谢 感谢郭楠老师在实验过程中提供的支持与帮助。

参 考 文 献

- [1] Christian Nilsson. Heuristics for the Traveling Salesman Problem[M]. Wiley Encyclopedia of Operations Research and Management Science. John Wiley & Sons, Inc. 2011.
- [2] Steinbrunn M, Moerkotte G, Kemper A. Heuristic and Ran2 domized Optimization for the Join Ordering Problem[J]. The VLDB Journal, 1997, 6 (3): 8 - 17.
- [3] 姚新, 陈国良. 模拟退火算法及其应用[J]. 计算机研究与发展, 1990(7): 1-6.
- [4] Fred Glover (1986). "Future Paths for Integer Programming and Links to Artificial Intelligence". Computers and Operations Research. 13 (5): 533-549. doi:10.1016/0305-0548(86)90048-1.
- [5] Fred Glover (1989). "Tabu Search - Part 1". ORSA Journal on Computing. 1 (2): 190-206. doi:10.1287/ijoc.1.3.190.
- [6] Fred Glover (1990). "Tabu Search - Part 2". ORSA Journal on Computing. 2 (1): 4-32. doi:10.1287/ijoc.2.1.4