**调研报告**

1. **实际问题描述**

在B2C农产品电子商务物流配送时，物流车装载当日需要配送的货品从仓库出发，按照事先规划好的最优配送路径为每一个客户进行配送，最后返回仓库。IT系统在配送之前需要根据客户的配送地址间线路间距、经验路况做分析计算出一条最优配送路径。在配送过程中，如果某路段堵车，需要动态调整配送路线。

1. **问题转化**

**旅行商问题（Traveling Salesman Problem，简称TSP）**：是一种典型的组合优化问题，已经被证明是一个ＮＰ complete（non- deterministic poly-nominal complete）难题。经典的ＴＳＰ的描述为：给定ｎ个城市和两两城市之间的距离，有一个旅行商从某一城市出发，要求确定一条经过各城市当且仅当一次的最短路线。设城市i和城市j的距离为d[i,j],如果对于任意i, j满足d[i,j] = d[j,i]，那么该问题为**对称旅行商问题（Symmetric Traveling Salesman Problem，简称STSP）**，否则为**非对称旅行商问题（Asymmetric Traveling Salesman Problem，简称ATSP）**。ATSP问题可以转化成TSP问题，但是问题规模会翻倍，详见维基百科。

实际情况，对于机动车而言道路是分单行道和双行道的， 因此该问题可以描述为在一个有向图的TSP问题，即ATSP问题。

1. **算法介绍**

目前求解TSP问题的方法分为两大类：**精确算法（Exact algorithms）**和**启发式算法（Heuristic and approximation algorithms）**。

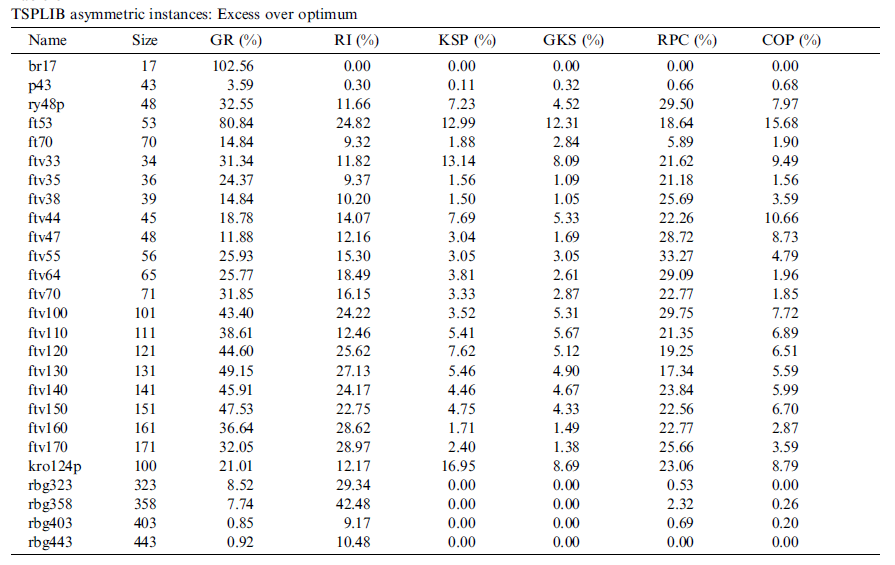
常用的精确算法主要有：**枚举法**、**分支限界（Branch and Bound，简称BB）**、**线性规划（Linear Programming）**、**回溯法**、和**动态规划（**Dynamic Programming，**简称**DP**）**等。其时间复杂度为指数级别，难以解决大规模问题。

启发式算法又分成三类：**构造算法（Constructive heuristics）**、**迭代改进算法（Iterative improvement）**和**随机改进算法（Randomised improvement）**。构造算法主要有：**最近邻点（Nearest Neighbour，简称NN。或者也称贪心）**、**基于最小生成树（Minimum Spanning Tree）的构造算法**等；迭代改进算法主要有：**2-opt（也称Pairwise exchange）**、**k-opt heuristic（也称Lin-Kernighan heuristics）**和**V-opt heuristic**等；随机改进算法主要有：**蚁群优化算法（Ant colony optimization，简称ACO）**等。

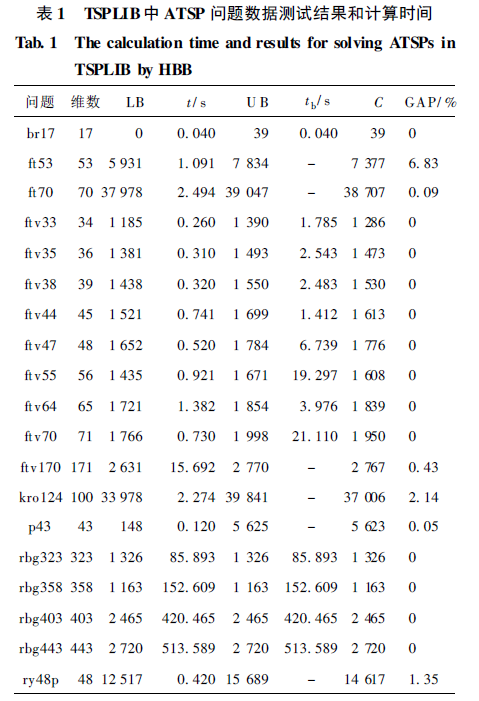
1. **各种算法比较**

“旅行商问题( Traveling Salesman Problem，TSP) 是典型的NP 完全组合优化问题, 具有广泛的 如遗传算法、禁忌搜索算法、蚁群算法, 神经网络算法以及从中发展起来的各种混合算法等。但这些算法大多对于对称或者规模较小的TSP 表现出更好的效果, 而且无法得到确定的最优解。”引自文论《基于阈值深度优先策略求解非对称旅行商问题的混合分枝定界算法》

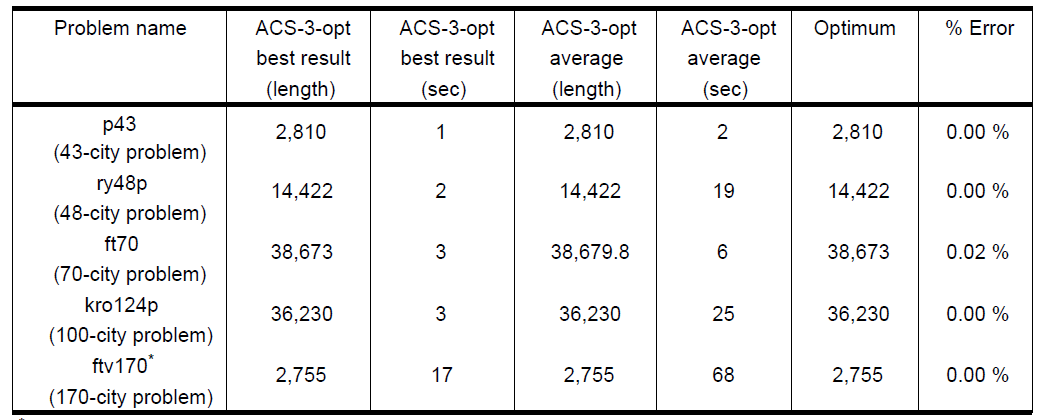
下面给出了一些有关ATSP的论文，这些论文都是以TSPLIB（http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/）上的测试数据来验证算法的效率。

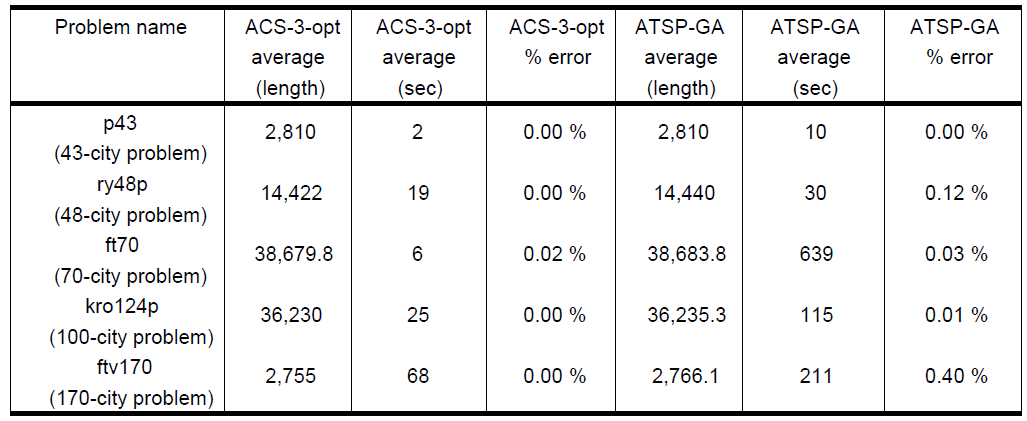


来自论文《Construction heuristics for the asymmetric TSP》，其中Name为测试数据的名字，Size为测试数据的规模（城市数），其余的列都是各种算法求得的结果与最优解的误差比例，GR为贪心算法（Greedy algorithm）；RI为；KSP为Karp-Steele patching heuristic；GKS为Modified Karp-Steele patching heuristic；RPC为Recursive path contraction；COP为Contract or patch heurisitc；

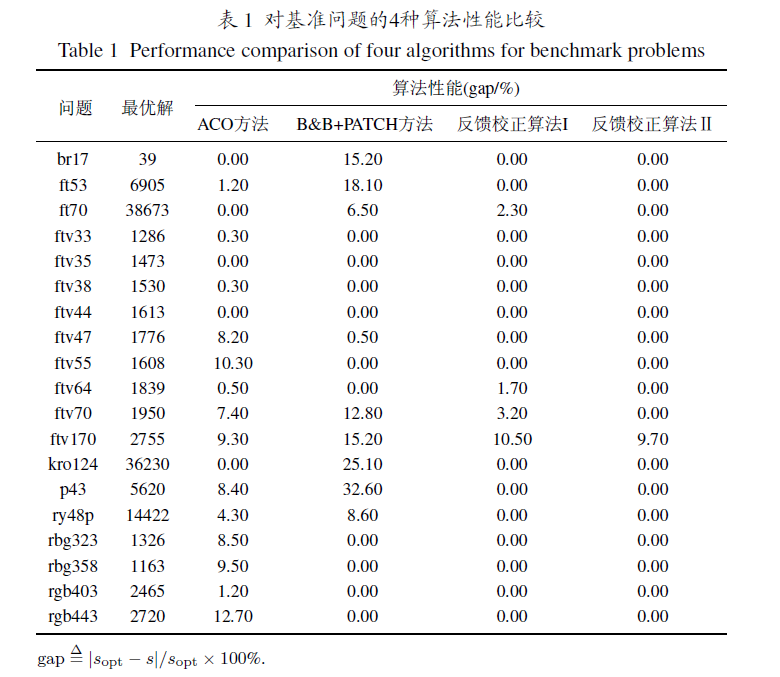


来自论文《基于阈值深度优先策略求解非对称旅行商问题的混合分枝定界算法》，其中问题为测试数据的名字，维度为测试数据的规模（城市数），t/s为时间（该论文测试的环境：采用C+ + 实现本文算法, 运行PC 机的配置为: 256MB 内存, Intel Pentium III 处理器, 主频845MHz. ）GAP/%是算法求得的结果与最优解的误差比例。LB和UB 为论文里算法所设定的下界和上界，C为论文算法所求结果。

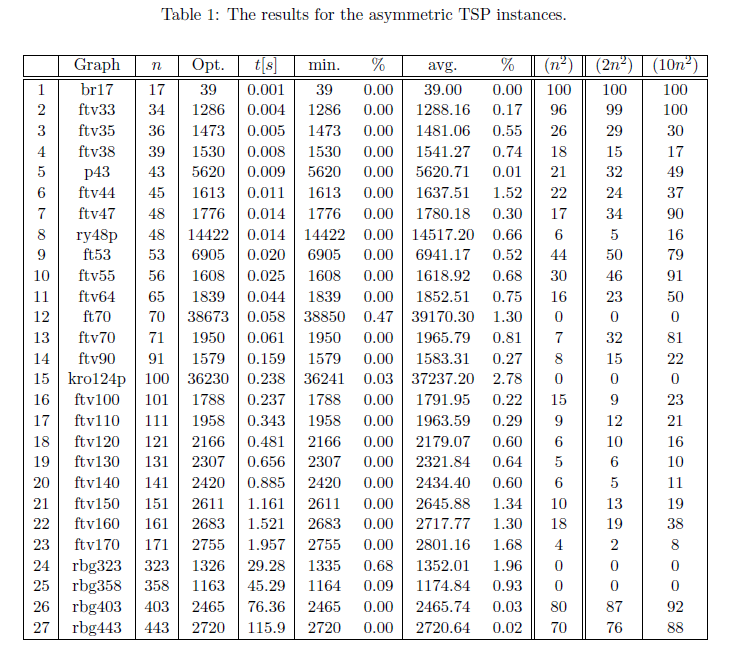




来自论文《Ant Colony System:A Cooperative Learning Approach to theTraveling Salesman Problem》，其中Problem name为测试数据的名字和问题规模，ACS-3-opt average（length）、ACS-3-opt average（sec）和ACS-3-opt % error分别是蚁群算法+ 3-opt优化算法所求得的结果、所需时间（单位秒） 和最优解的误差比例。ATSP-GA average（length）、ATSP-GA average(sec)和ATSP-GA % error分别是遗传算法所求得的结果、所需时间（单位秒） 和最优解的误差比例。



来自论文《基于反馈校正原理的非对称旅行商问题的自收敛优化算法》，其中问题为测试数据的名字，其他列很容易从图中得知。 仿真环境为PC机：2.8G Pentium IV处理器，512M内存，VC++6.0开发环境。



来自论文《A Heuristic for the Asymmetric Traveling Salesman Problem》，其中Graph为测试数据的名字，n为问题规模（城市数），Opt.为最优解，t（s）为求解时间（仿真环境： SunFire v40z服务器 FC3 linux操作系统），min.为单独执行100次取最小值，%为最小值和最优解的误差率，avg.为单独执行100次后取平均值，%为平均值与最优解的误差率，其余三个是设置算法里面的迭代次数后算法在单独执行100中可以得到最优解的次数。

1. **总结**

目前找到最好的算法，是使用TSPLIB上的数据测试的《 A Heuristic for the Asymmetric Traveling Salesman Problem 》

具体算法：RAI(Randomized Arbitrary Insertion), 最坏时间复杂度O（n^4）,空间复杂度O（n），实现简单，效果好。

亲自测试，结果如论文所示。