

Heuristic Search for Capacitated Arc Routing Problem ——A Tabu Search Implementation

Mark Zeng
2017/12/25

Outline

1. CARP 问题的表述
2. 整体算法框架
3. 贪心算法
4. Tabu搜索实现
5. 效果展示及实验

CARP问题的表述

$$S = \{R_1, R_2, R_3, \dots, R_n\}$$

$$R_i = \{t_1, t_2, t_3, \dots, t_n\}$$

$$S = \begin{cases} R_1 = (1, 2), (2, 3), (3, 5), (5, 6), (6, 12) \\ R_2 = (1, 4), (4, 2), (2, 9), (9, 11), (11, 5) \\ R_3 = (1, 12), (12, 5), (3, 4), (9, 10), (10, 1) \\ R_4 = (1, 7), (7, 8), (8, 11), (11, 10), (10, 8) \\ R_5 = (12, 7), (7, 6) \end{cases}$$

贪心算法

贪心算法：

1. 从起点出发，选取距离起点最近的任务
2. 以上一个任务的终点为起点，重复步骤1，直到超过负载

如何打破平衡？

五种打破平衡的方式 < 多次 (>14) 随机选取

是否尽可能装满？
否

	随机解	贪心解	随机贪心(装满)	随机贪心(不装满)	最优解
cost	13000	6400	5900	5500	5018

Tabu算法思想

基于爬山算法:

每次迭代搜索所有邻域, 从中选最好的解作为下一次迭代的起点

邻域表示:

$$S = \begin{cases} R_1 = (1, 2), (2, 3), (3, 5), (5, 6), (6, 12) \\ R_2 = (1, 4), (4, 2), (2, 9), (9, 11), (11, 5) \\ R_3 = (1, 12), (12, 5), (3, 4), (9, 10), (10, 1) \\ R_4 = (1, 7), (7, 8), (8, 11), (11, 10), (10, 8) \\ R_5 = (12, 7), (7, 6) \end{cases}$$

Single Insertion

Double Insertion

Swap

爬山算法的问题:

一旦陷入局部最优就出不来了

Tabu算法思想

在爬山算法的基础上，维护一张禁忌列表，避免陷入局部最优

基于单个解的禁忌列表：

Tabu list
Solution1
Solution2
Solution3
Solution4
.....

基于多个解的禁忌列表：

Tabu list
Solution group1
Solution group2
Solution group3
Solution group4
.....

基于操作的禁忌列表：

Tabu list
Operation1
Operation2
Operation3
Operation4
.....

Tabu算法思想

另一个重要的想法：

给不可行解一个机会，用惩罚因子来控制节奏

不可行解：

不满足约束条件的解

为什么给它机会？

优秀的不可行解有可能变换成优秀的可行解

怎么防止越走越偏？

指数增长的惩罚因子

If 连续10次不可行：

$$P = P * 2$$

If 连续10次可行：

$$P = P / 2$$

整体算法流程

1. 用贪心算法找一个最好的初始解 S
2. 以 S 为起点搜索所有邻域，选取不在禁忌列表的最好的解 S'
3. 用 S' 更新禁忌列表，令 $S = S'$
4. 重复2、3步直到时间结束