

# 21307077 凌国明 嵌入式系统作业三

## 1970 年代 Kernighan-Lin 算法

### 解决的问题

该算法是为解决旅行商问题（TSP）而提出的。旅行商问题是一种经典的组合优化问题，目标是找到一个最短的路径，使得一名旅行商可以访问一系列城市各一次，并最终返回起始城市。形式上，给定一个城市集合和每对城市之间的距离，要求找到一条经过每个城市恰好一次且返回起点的最短可能路线。

### 难点和挑战

复杂度：TSP是一个NP-hard问题，意味着找到确切解的计算复杂度随着城市数量的增加而迅速增长。即使对于中等规模的城市数量，可能的路线组合数量也是天文数字。尽管理论上可以通过穷举所有可能的路径来找到最优解，这在实际中是不可行的，特别是问题规模较大时。

### 解决方法

Kernighan和Lin提出了一种启发式算法。这种方法通过不断交换路径的部分来逐步优化总旅程的长度。

### 初始解

算法首先需要有一个初始解，这通常是一个随机生成的或使用一些简单启发式方法（如最近邻算法）得到的TSP路径。

### 两阶段优化过程

Kernighan-Lin算法通过两个主要阶段来优化路径：交换和反转。

1. 交换：在交换阶段，算法尝试通过交换路径中的两个不连续的城市来找到更短的路径。例如，如果路径是  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ ，算法可能会考虑交换城市 B 和 D，产生新路径  $A \rightarrow D \rightarrow C \rightarrow B \rightarrow E$ 。
2. 反转：在反转阶段，算法通过选择路径上的两个点并反转这两点间的路径来尝试改善总长度。例如，对于同样的路径  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ ，选择 B 和 D 之间的路径并将其反转，得到新路径  $A \rightarrow C \rightarrow B \rightarrow D \rightarrow E$ 。

在每次交换或反转后，算法计算新路径的总长度。只有当这个新路径比当前路径短时，才会接受这个新路径。

### 迭代过程

迭代以上过程，算法反复执行交换和反转操作，直到达到一定的迭代次数或者在连续几次迭代中没有找到更好的解。最终，当算法停止迭代时，它返回当前的路径作为最终解。尽管这个解可能不是全局最优解，但它通常是一个接近最优的高质量解。

# 为什么有效

以下是几个对Kernighan-Lin有效性的说明

1. 启发式搜索：Kernighan-Lin算法采用启发式搜索方法。与完全随机的搜索不同，启发式搜索通过特定的规则（如交换和反转）来引导搜索过程，这样可以更高效地探索解空间。
2. 局部优化策略：通过局部操作（如交换和反转），算法可以逐步改善当前解。这些操作专注于通过小的调整来优化路径，这常常可以有效地减少总路径长度。
3. 迭代可能跳出局部最优：虽然局部优化有可能陷入局部最优解，但Kernighan-Lin算法通过反复的交换和反转操作，增加了跳出局部最优解并接近全局最优解的可能性。
4. 迭代过程灵活：算法的迭代过程允许它在多次尝试后找到更好的解。这个过程不是一次性的，而是一个渐进的优化过程。
5. 经验：理论上Kernighan-Lin算法不能保证找到全局最优解，但在实际中，它常常能够产生非常接近最优的解。

## 相关的思考

在看这篇论文时，我认为这种方法跟模拟退火算法十分接近（之前用过几次模拟退火算法，所以会熟悉一点）。当时认为只要将模拟退火的“接受差解的概率”设置为0，退火扰动方式改成上面提到的交换与反转时，这个“模拟退火”算法就“退化”成了Kernighan-Lin算法。

基于这点思考，认为模拟退火算法是优于Kernighan-Lin算法的，因为模拟退火算法不仅接受改善后的解，也以一定概率接受使解变差的变化。而Kernighan-Lin倾向于陷入局部最优解，因为它通常不接受使解变差的变化。

经过查阅，模拟退火算法的提出时间是1983年