



東北大學
Northeastern University

大规模带有时间窗的绿色车辆路径问题的自适应大邻域算法求解

汇报人：张平

导师：于洋

邮箱：1257524054@qq.com

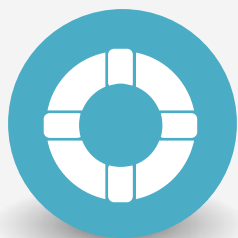
目录



1 GVRPTW介绍



2 自适应大邻域搜索算法介绍与改进

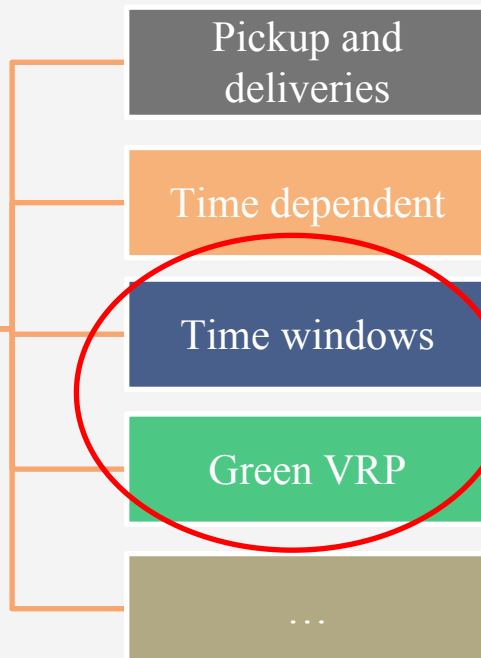


01 GVRPTW的介绍

GVRP与VRP的关系

VRP问题及其变种

车辆路径问题 (Vehicle
routing problem)



研究对象

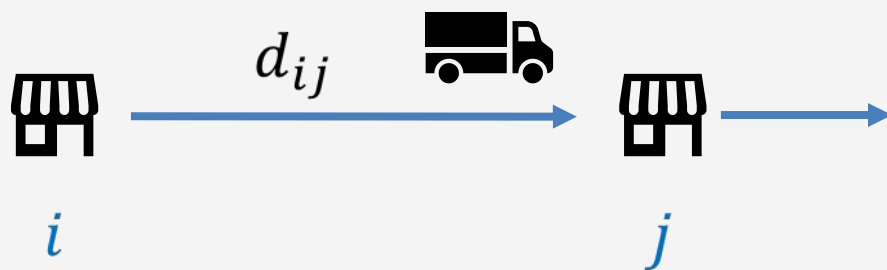
大规模带有时间窗的绿
色车辆路径问题

GVRP与VRP的区别

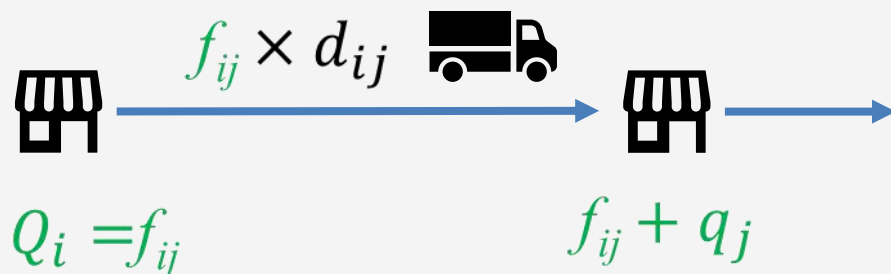
不同于VRP的目标函数只受距离影响，GVRP目标函数受到**载重**和**距离**的双重影响。
目前提出的删除启发式算法都是基于VRP背景下提出的，并不能很好的适GVRPTW问题。

由GVRP目标函数可知，经过顾客点载重的顺序会对目标函数有影响，载重大的客户点越往后访问，路径上的碳排放越小。

VRP



GVRP



GVRPTW数学模型

$$\text{Minimize} \quad r \sum_{m=1}^{|M|} \sum_{(i,j) \in E} d_{ij} [\alpha_1 + \alpha_2 (w_m x_{ij}^m + f_{ij}^m)] \quad (1)$$

$$s.t. \quad \sum_{m=1}^{|M|} \sum_{j \in V \setminus \{i\}} x_{ij}^m = 1, \quad \forall i \in V_0, \quad (2)$$

$$\sum_{m=1}^{|M|} \sum_{j \in V \setminus \{i\}} x_{ji}^m = 1, \quad \forall i \in V_0, \quad (3)$$

$$\sum_{m=1}^{|M|} \sum_{j \in V \setminus \{i\}} f_{ij}^m - \sum_{m=1}^{|M|} \sum_{j \in V \setminus \{i\}} f_{ji}^m = q_i, \quad \forall i \in V_0, \quad (4)$$

$$q_i x_{ij}^m \leq f_{ij}^m \leq (Q_m - q_j) x_{ij}^m, \quad \forall (i,j) \in E, m=1, \dots, |M|, \quad (5)$$

$$\sum_{m=1}^{|M|} \sum_{i \in V_0} x_{i0}^m = \sum_{m=1}^{|M|} \sum_{i \in V_0} x_{0i}^m \geq 0, \quad (6)$$

$$x_{ij}^m, f_{ij}^m \geq 0, \quad \forall (i,j) \in E, m=1, \dots, |M|, \quad (7)$$

$$x_{ij}^m \in \{0,1\}, \quad \forall (i,j) \in E, m=1, \dots, |M|, \quad (8)$$

模型相关参数

x_{ij}^m $[0,1]$ 变量, $m = 1, \dots, |M|$, 为1时表示弧 (i, j) 被某车型为 m 所行驶的路线所使用;

w_m 车型为 m 车辆自重, $m = 1, \dots, |M|$;
连续变量, 表示车型为 m 的车辆在通过弧 (i, j) 时车辆的载重;

M 总车型数, $m = 1, \dots, |M|$;

V $V = \{0, 1, \dots, n+1\}$ 该问题中所有分布点的集合, 其中 $V_0 = V \setminus \{0, n+1\}$;

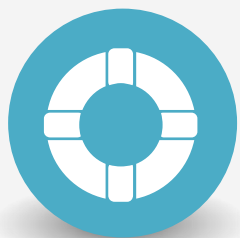
E 所有弧的集合, $(i, j) \in E$;

d_{ij} 顾客 i, j 两地间距离, 也表示弧 (i, j) 的长度;

Q_m 车型为 m 的车辆的载重上限, $m = 1, \dots, |M|$;

q_j 顾客点 j 的服务需求;

α_1, α_2 路径计算相关参数。



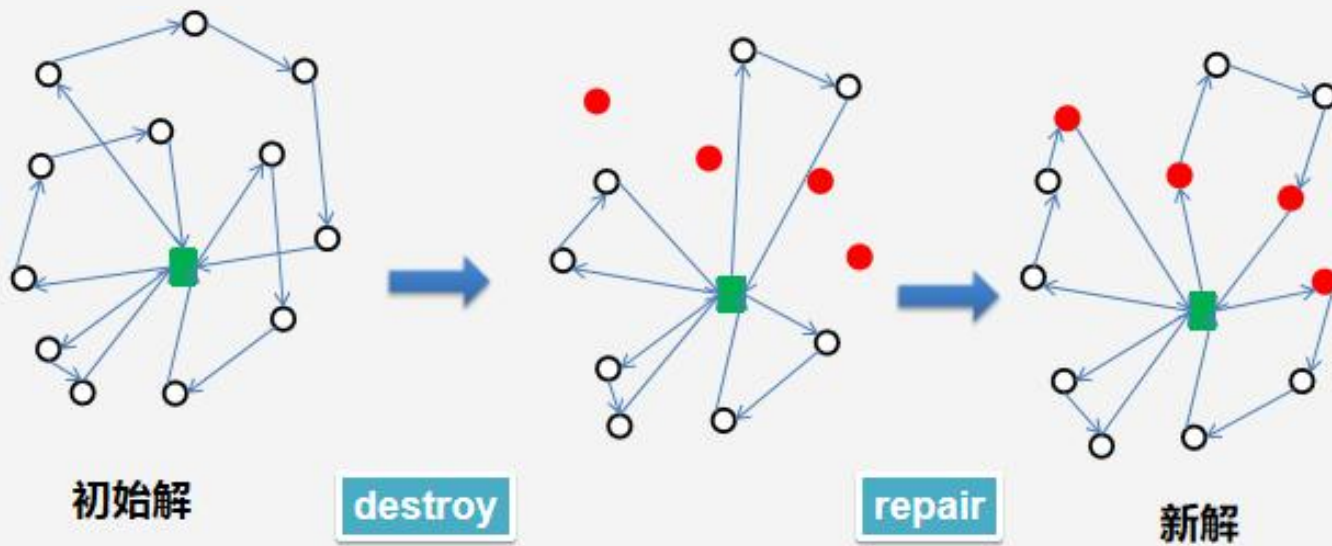
02 自适应大邻域算法介绍与改进

2 自适应大邻域算法的介绍与改进

2.1 自适应大邻域算法介绍

在ALNS(Adaptive Large Neighborhood Search)算法中，邻域是由破坏和修复方法隐式定义的。

破坏方法会破坏当前解的一部分，而后修复方法会对被破坏的解进行重建，从而得到的一系列解。





2.2 自适应大邻域算法改进

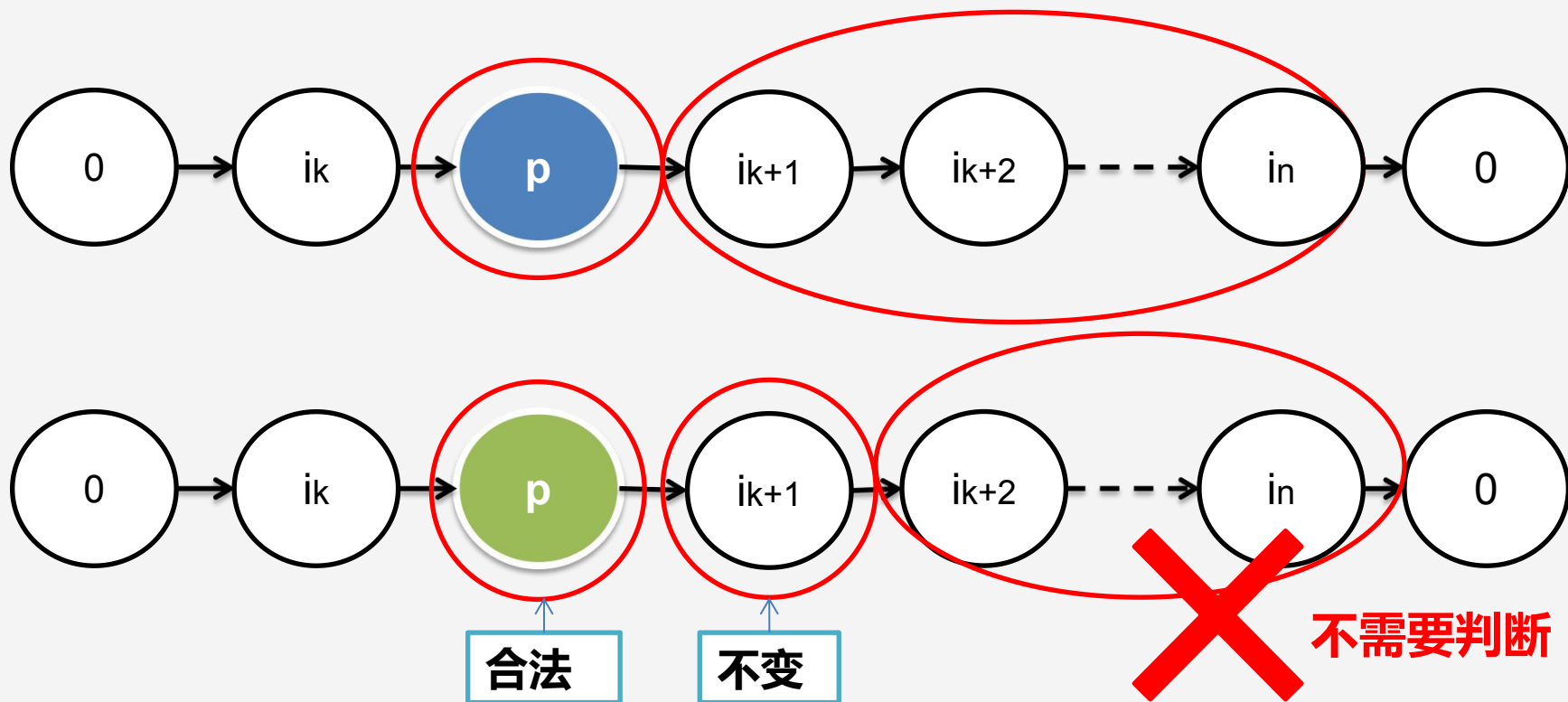
传统自适应大邻域搜索算法在修复环节对于非法解的杜绝通常使用**惩罚手段**，代价是**先计算后判断，计算开销较大**。

对此我做了以下两点改进：

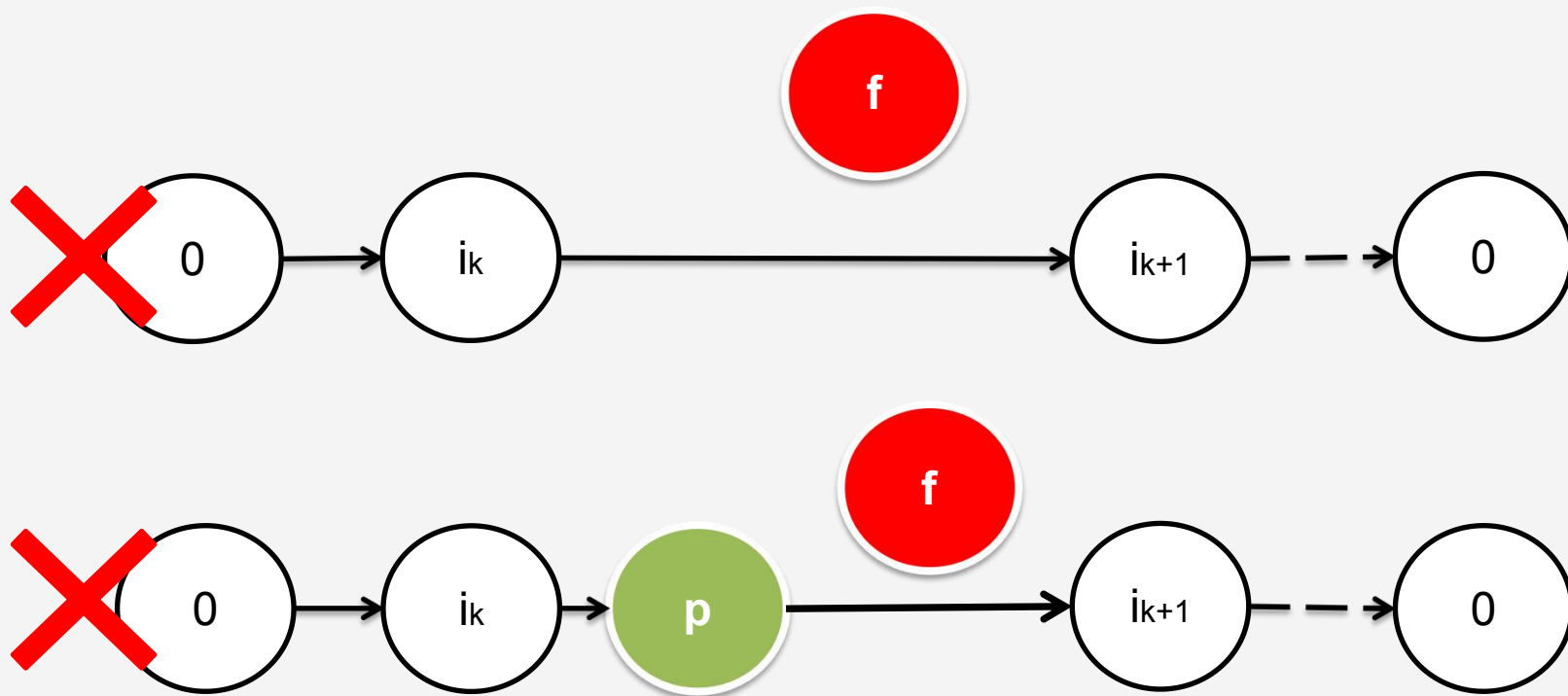
- 1.我通过**先判断**解的合法性之后**再计算**，从而避免无效计算。
- 2.在此基础上，我提出一些**插入规则**来**简化判断的过程**。

插入规则介绍

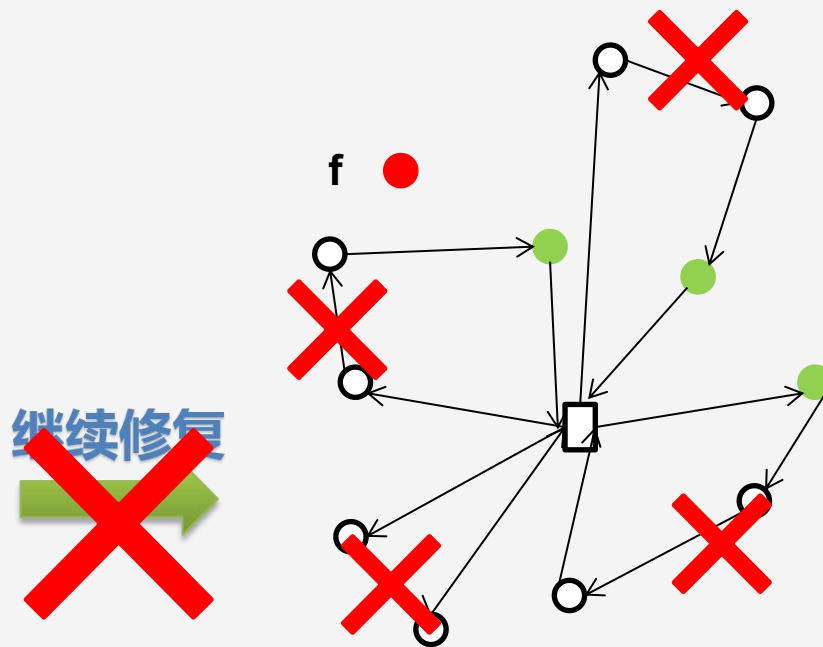
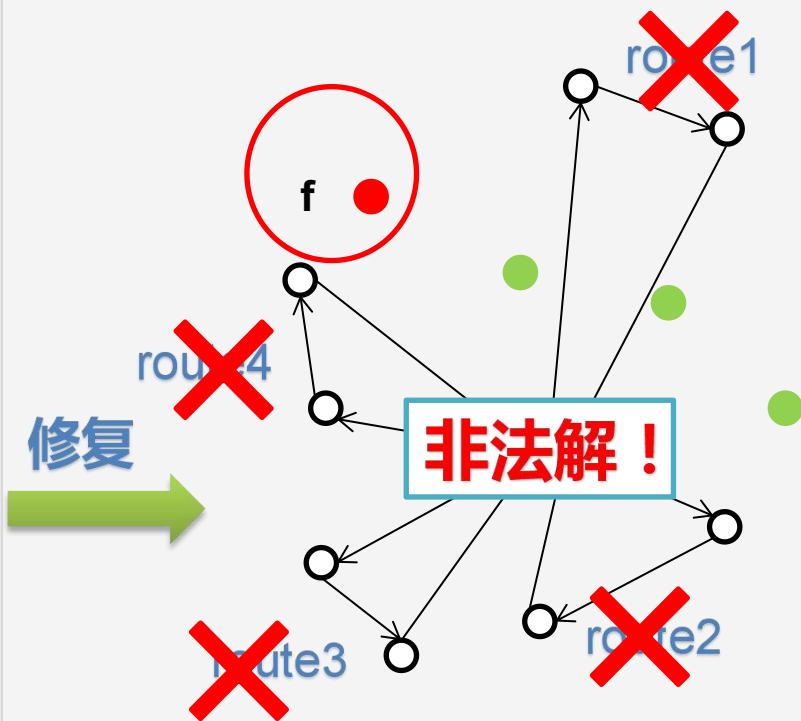
规则1：在满足容量约束的条件下,如果插入一个节点 p ，该节点合法且后一个节点的发车时间不变，那么不需要继续判断，即可得出该路径为合法路径。



规则2：在满足容量约束的条件下,如果待插入点 f 在route中现在没有合法插入位置,那么之后route发生变化也不会存在合法插入位置。

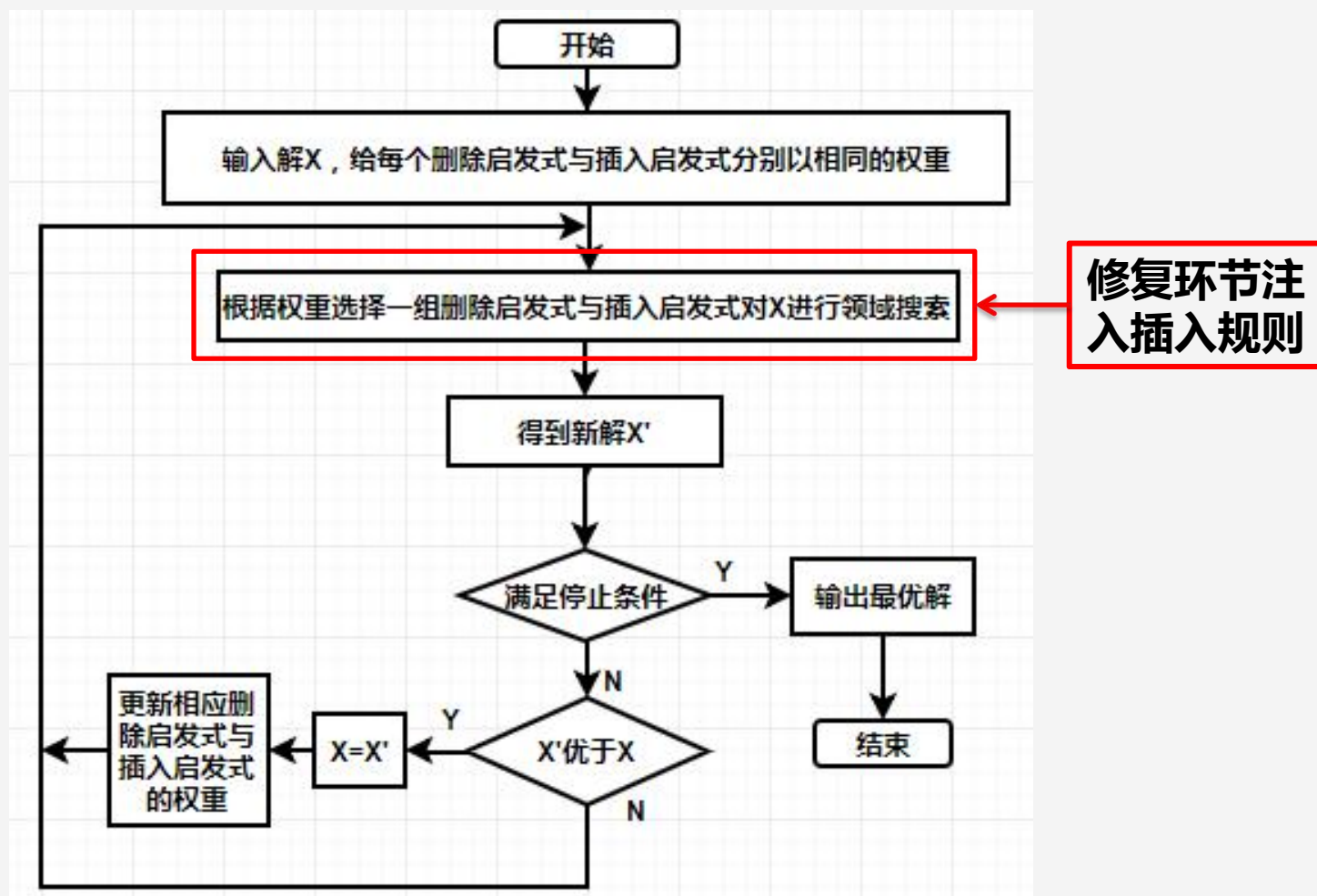


规则3：在修复操作中，如果发现这样一个待插入f节点,在所有route中都无法合法插入位置，计算即可停止。由**规则2**可以知，之后也不会有合法插入，故为非法解。



2.2 自适应大邻域算法的介绍与改进

2.3 改进型ALNS算法流程图



谢谢！

