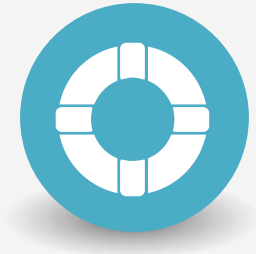
大规模带有时间窗的绿色车辆路径问题的自适应大邻域算法求解

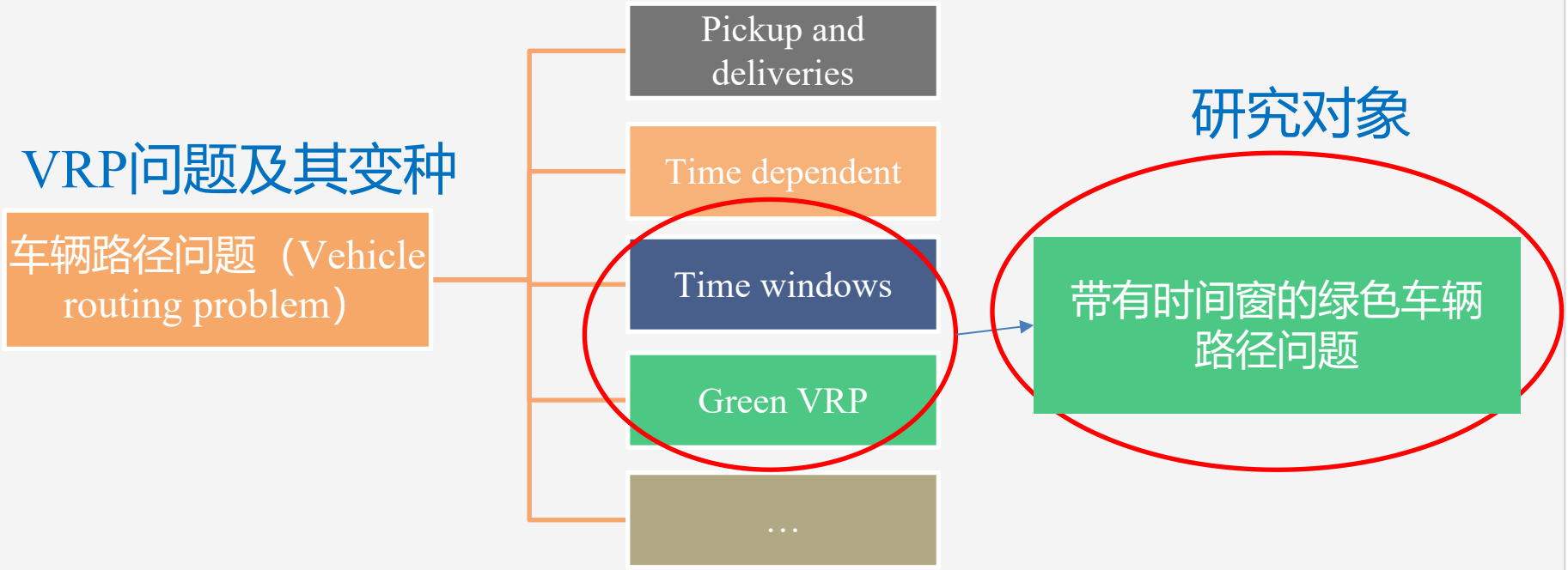
汇报人:	**张平**	**导　师:**	**于洋**
邮　箱:	1257524054@qq.com		

目录



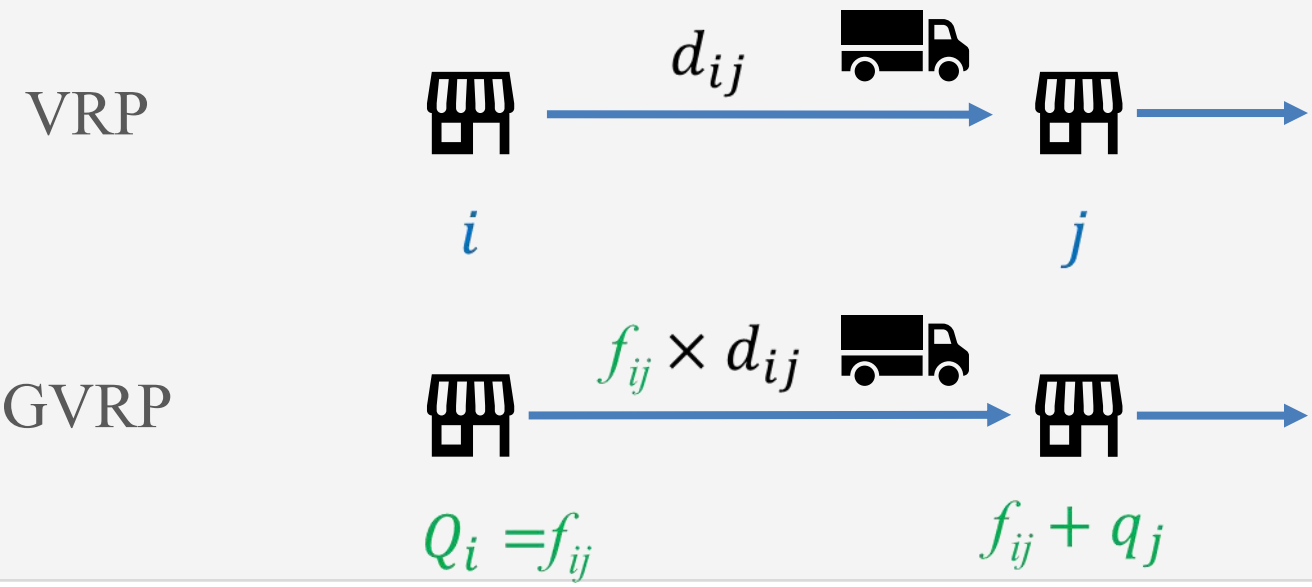
01 GVRPTW的介绍

GVRP与VRP的关系



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。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。



GVRPTW数学模型

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

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

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

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

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

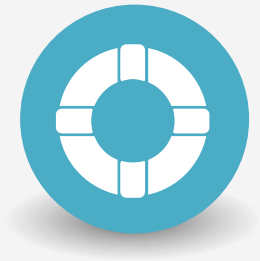
$$\sum_{m=1}^{|M|}\sum_{i\in V_0} x_{i0}^m = \sum_{m=1}^{|M|}\sum_{i\in V_0} x_{0i}^m \ge 0, \tag{6}$$

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

$$x_{ij}^m \in \mathbb{Z}, \quad \forall (i,j)\in E, m=1,\ldots,|M|, \tag{8}$$

模型相关参数

x_{ij}^{m}	[0,1]变量, $m = 1,\ldots, \lvert M\rvert$，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，$m = 1,\ldots, \lvert M\rvert$；
	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，$m = 1,\ldots, \lvert M\rvert$；
V	$V=\{0, 1,\ldots, n+1\}$该问题中所有分布点的集合,其中$V_0=V\backslash\{0, n+1\}$；
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，$m = 1,\ldots, \lvert M\rvert$；
q_j	顾客点j的服务需求；
α_1, α_2	路径计算相关参数。

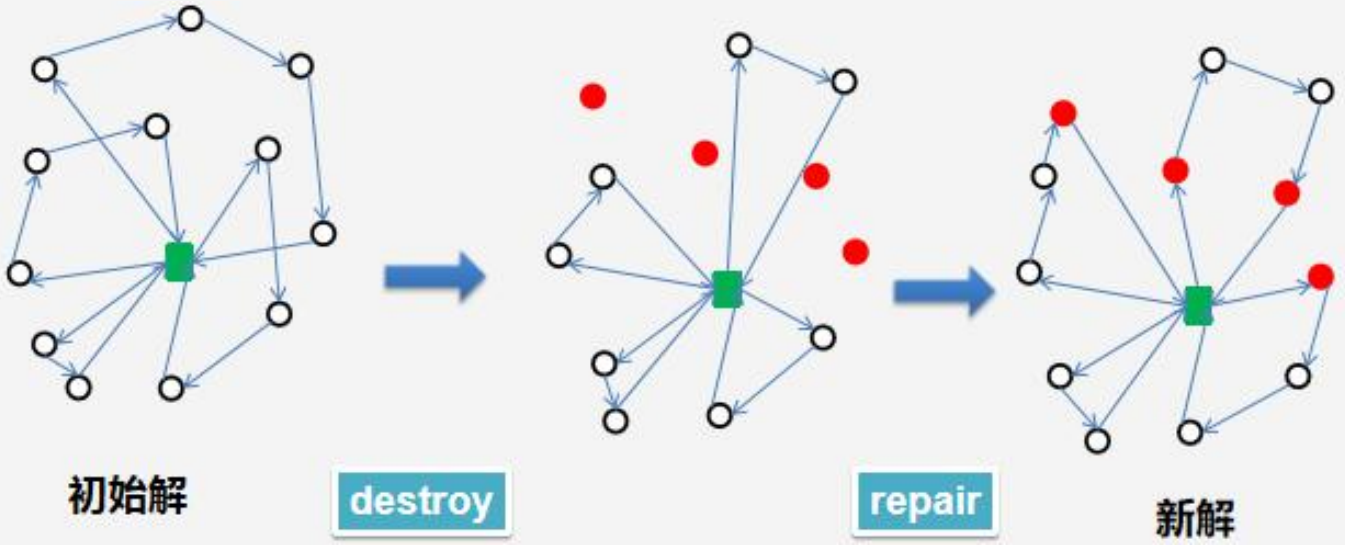


02 自适应大邻域算法介绍与改进

2.1 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。



自适应大邻域算法的介绍与改进

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。

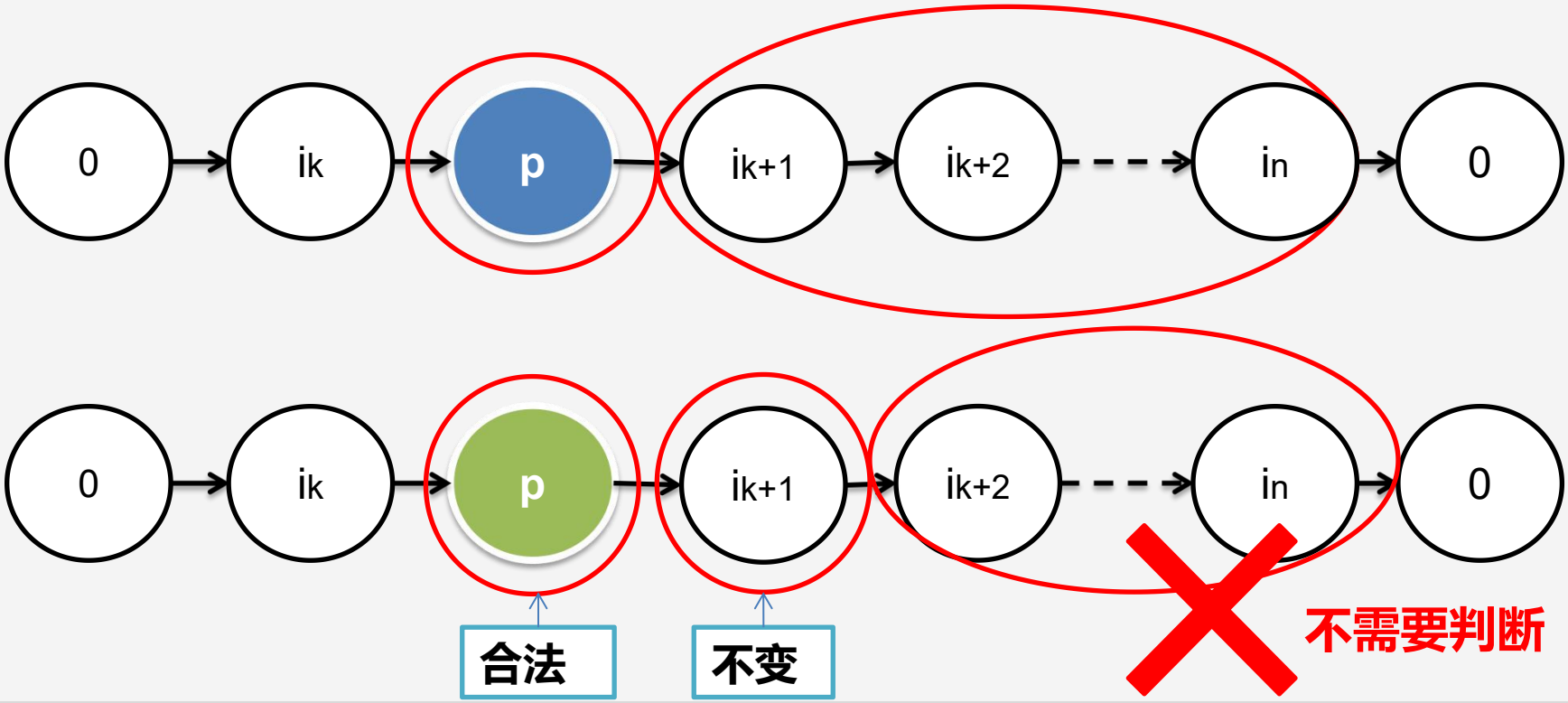
对此我做了以下两点改进：

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。

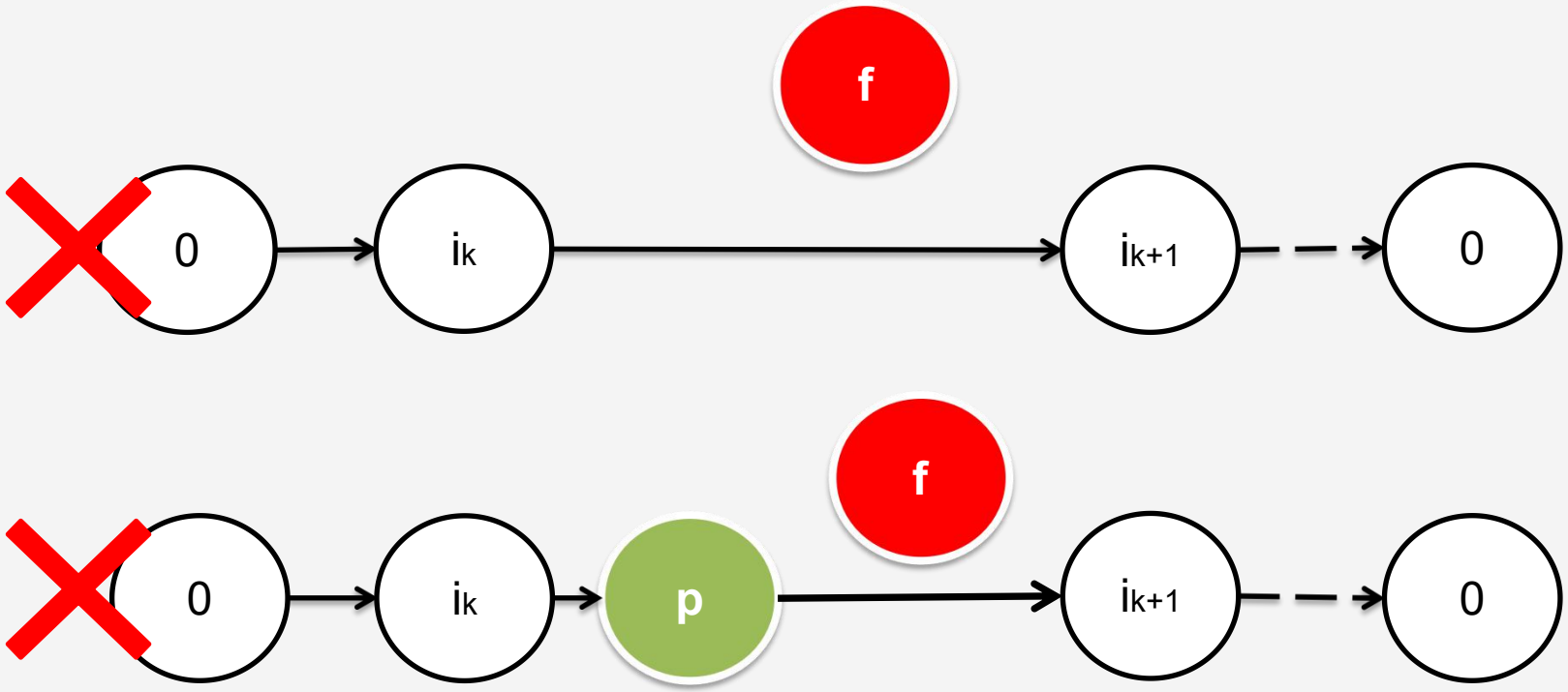
2.在此基础上，我提出一些插入规则来简化判断的过程。

插入规则介绍

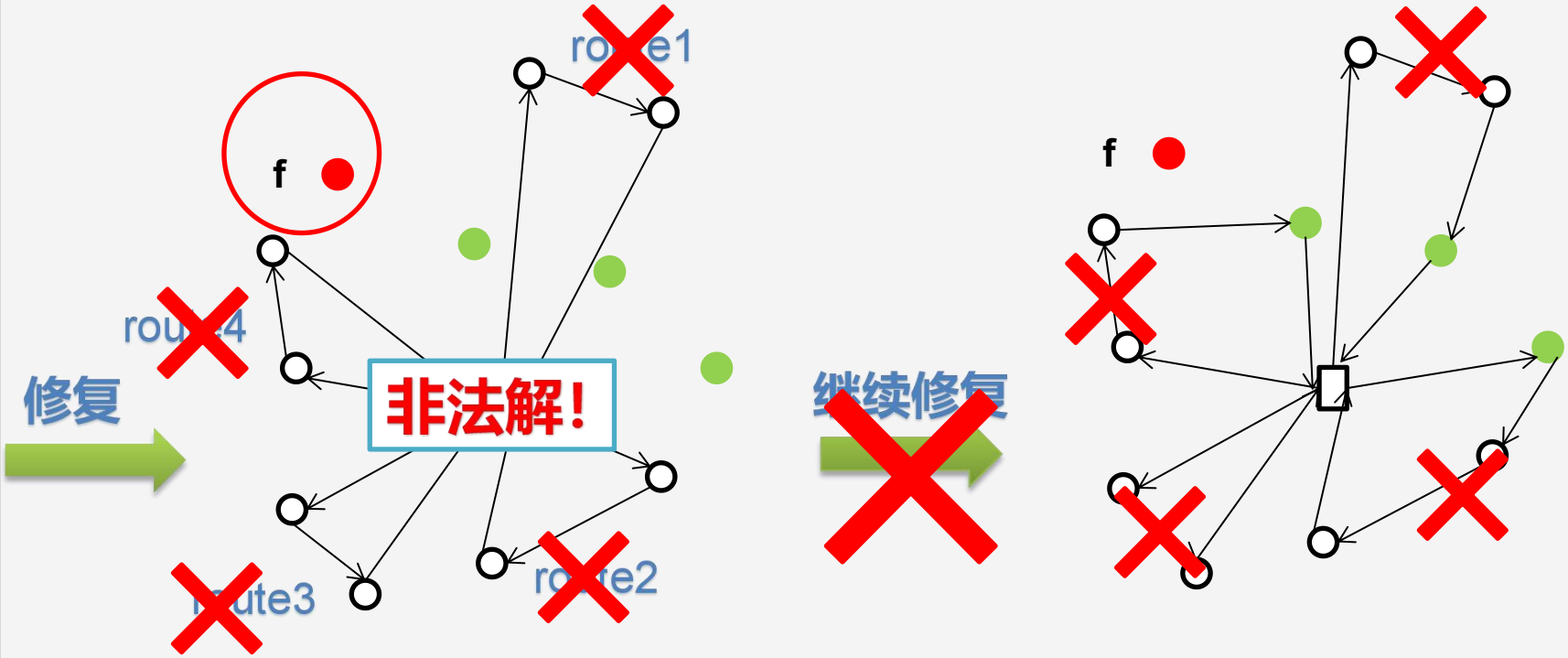
规则1：在满足容量约束的条件下,如果插入一个节点p，该节点合法且后一个节点的发车时间不变，那么不需要继续判断，即可得出该路径为合法路径。



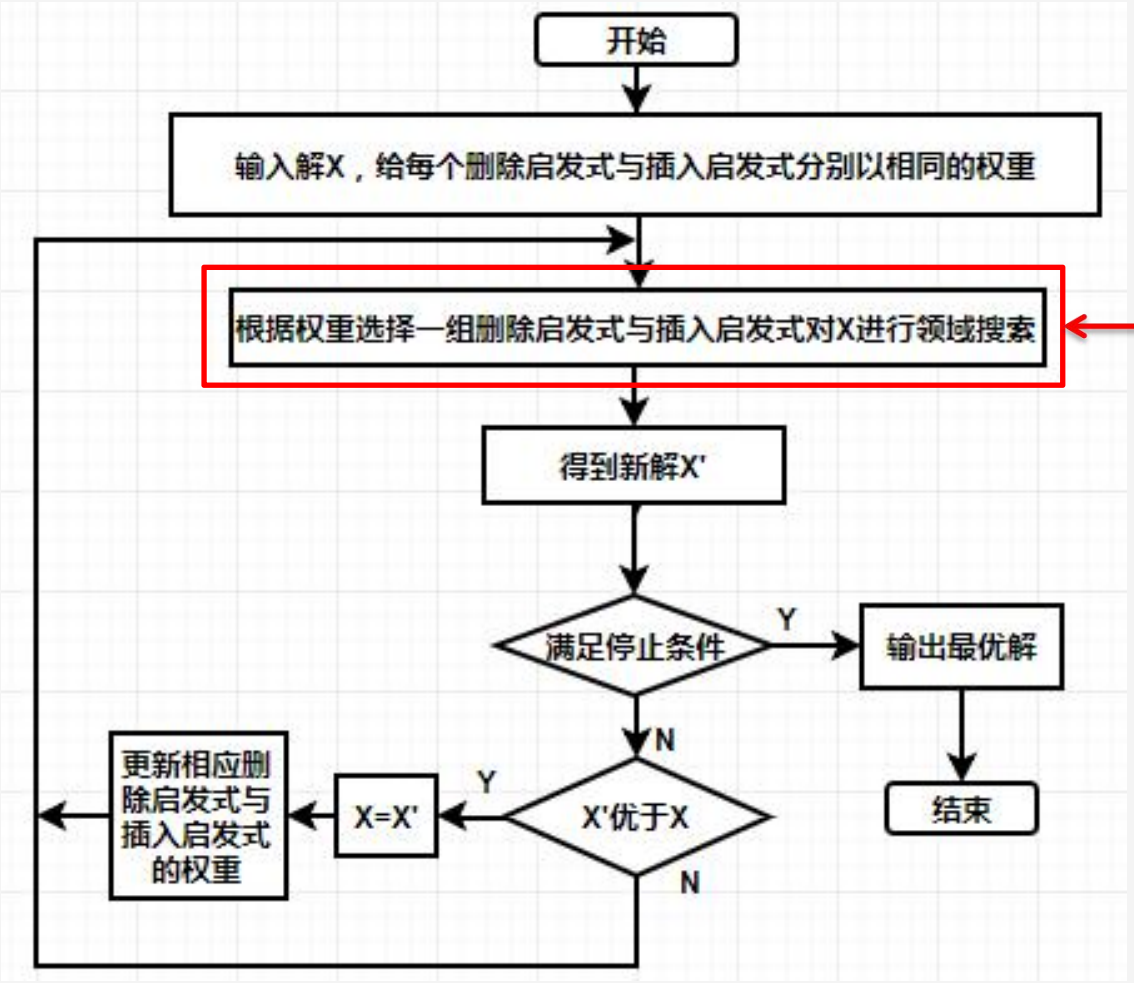
规则2：在满足容量约束的条件下,如果待插入点f在route中现在没有合法插入位置，那么之后route发生变化也不会存在合法插入位置。



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。



2.3 改进型ALNS算法流程图



2.4 实验数据

Table 4 Computational results of the instances with 100 customers

Instance	Exact		Classic ALNS		the proposed ALNS	
	ECE	*CT* (s)	*MCE*	*Gap* %	*MCE*	*Gap* %
r101	1127.68	171.72	1166.28	3.42%	1164.65	3.28%
r201	1027.12	22363.17	1095.67	6.67%	1057.08	2.92%
c101	818.56	44.20	850.54	3.91%	835.15	2.03%
c105	597.03	32.72	610.56	2.27%	610.56	2.27%
Average				4.07%		2.62%

谢谢！