

EVRP数学公式完整版

概述

本文档包含EVRP（电动车辆路径规划问题）的完整数学公式体系，适用于Mathpix Markdown插件，可直接用于PPT和学术论文。

符号定义

集合和索引

- $V = \{0, 1, 2, \dots, n\}$: 所有节点集合，0表示配送中心
- $C = \{1, 2, \dots, n\}$: 客户节点集合
- F : 充电站节点集合
- K : 车辆集合
- (i, j) : 从节点*i*到节点*j*的边

参数

符号	含义	单位
Q	车辆最大载重容量	kg
B	车辆电池最大容量	kWh
d_{ij}	节点 <i>i</i> 到节点 <i>j</i> 的距离	km
q_i	客户 <i>i</i> 的需求量	kg
e_{ij}	从节点 <i>i</i> 到节点 <i>j</i> 的能耗	kWh
t_{ij}	从节点 <i>i</i> 到节点 <i>j</i> 的行驶时间	h
s_i	在节点 <i>i</i> 的服务时间	h
c_{ij}	从节点 <i>i</i> 到节点 <i>j</i> 的行驶成本	元
γ	单位充电成本	元/kWh
β	单位时间成本	元/h

决策变量

符号	类型	含义
x_{ijk}	二进制	车辆 <i>k</i> 是否从节点 <i>i</i> 直接到节点 <i>j</i>
y_{ik}	连续	车辆 <i>k</i> 到达节点 <i>i</i> 时的剩余载重
b_{ik}	连续	车辆 <i>k</i> 到达节点 <i>i</i> 时的剩余电量
u_{ik}	连续	车辆 <i>k</i> 在节点 <i>i</i> 的充电量

符号	类型	含义
t_{ik}	连续	车辆 k 到达节点 i 的时间

目标函数

主目标函数

最小化总成本：
$$\min Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} + \sum_{k \in K} \sum_{i \in F} \gamma u_{ik} + \sum_{k \in K} \sum_{i \in V} \beta t_{ik}$$

分解目标

1. 距离成本：

$$C_{\text{dist}} = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} \cdot c_{\text{unit}} \cdot x_{ijk}$$
2. 充电成本：

$$C_{\text{charge}} = \sum_{k \in K} \sum_{i \in F} \gamma \cdot u_{ik}$$
3. 时间成本：

$$C_{\text{time}} = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} t_{ij} \cdot c_{\text{time}} \cdot x_{ijk}$$

约束条件

1. 客户访问约束

每个客户被访问一次：
$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1, \quad \forall j \in C$$

2. 流量守恒约束

车辆平衡：
$$\sum_{i \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \quad \forall k \in K, \forall i \in V$$

3. 载重容量约束

车辆载重限制：
$$q_i \leq y_{ik} \leq Q, \quad \forall k \in K, \forall i \in V$$
$$y_{jk} \leq y_{ik} - q_i x_{ijk} + Q(1 - x_{ijk}), \quad \forall k \in K, \forall i, j \in V$$

4. 电池容量约束

电量始终 ≥ 0 ：
$$0 \leq b_{ik} \leq B, \quad \forall k \in K, \forall i \in V$$
$$b_{jk} \leq b_{ik} - e_{ij} x_{ijk} + B(1 - x_{ijk}), \quad \forall k \in K, \forall i, j \in V$$

5. 充电站约束

充电站可多次访问：
$$b_{ik} + u_{ik} \leq B, \quad \forall k \in K, \forall i \in F$$

6. 路径连续性约束

从配送中心出发并返回：
$$\sum_{j \in V} x_{0jk} = 1, \quad \forall k \in K$$
$$\sum_{i \in V} x_{i0k} = 1, \quad \forall k \in K$$

7. 时间窗约束（可选）

$$t_{jk} \geq t_{ik} + s_i + t_{ij} - M(1 - x_{ijk}), \quad \forall k \in K, \forall i, j \in V$$
$$a_i \leq t_{ik} \leq b_i, \quad \forall k \in K, \forall i \in C$$
 其中 $[a_i, b_i]$ 是客户 i 的时间窗， M 是足够大的常数

遗传算法公式

适应度函数

对于个体 s ： $f(s) = \frac{1}{Z(s) + \alpha \cdot P(s)}$

选择操作

锦标赛选择： $P_{\text{select}}(i) = \frac{f(i)}{\sum_{j=1}^t f(j)}$

排序选择： $P_{\text{select}}(i) = \frac{2-s+2(s-1)\frac{i-1}{N-1}}{N}$

变异操作

交换变异： $\text{Swap}(\pi, i, j): \pi[i] \leftrightarrow \pi[j]$

插入变异： $\text{Insert}(\pi, i, j): \text{move } \pi[i] \text{ to position } j$

反转变异： $\text{Reverse}(\pi, i, j): \text{reverse subsequence } [i, j]$

性能评估指标

收敛性指标

1. 收敛代数： $G_{\text{conv}} = \min\{g \mid |f_{\text{best}}(g) - f_{\text{best}}(g-\Delta)| < \epsilon \cdot f_{\text{best}}(g)\}$
2. 改进率： $\text{Improvement} = \frac{f_{\text{initial}} - f_{\text{final}}}{f_{\text{initial}}} \times 100\%$
3. 收敛速率： $\lambda = -\frac{1}{G} \ln\left(\frac{f_G - f_{\infty}}{f_0 - f_{\infty}}\right)$

算法质量指标

1. 最优性差距： $\text{Gap} = \frac{f_{\text{alg}} - f_{\text{opt}}}{f_{\text{opt}}} \times 100\%$
2. 鲁棒性： $\text{Robustness} = \frac{\sigma}{\mu} \times 100\%$
3. 计算效率： $\text{Efficiency} = \frac{\text{CPU时间}}{\text{问题规模}} \times 100\%$

统计检验

t检验： $t = \frac{\bar{f}_{GA} - \bar{f}_{\text{baseline}}}{\sqrt{\frac{s_{GA}^2}{n_{GA}} + \frac{s_{\text{baseline}}^2}{n_{\text{baseline}}}}}$

充电策略公式

充电判断条件

1. 电量安全约束： $b_{\text{current}} < \alpha \cdot B$
2. 可达性约束： $b_{\text{current}} < e_{i,j} + \beta \cdot B$
3. 充电量决策： 完全充电： $\nu_{\text{charge}} = B - b_{\text{current}}$

部分充电： $\nu_{\text{min}} = \max\{e_{\text{total}} - b_{\text{current}}, 0\}$

智能充电： $\nu^* = \arg\min_{\nu \in [0, B - b_{\text{current}}]} C_{\text{total}}(\nu)$

充电时间

$t_{\text{charge}} = \frac{\nu}{r_{\text{charge}}}$

充电站选择

$f^* = \arg\min_{f \in F} \left\{ d_{\text{current},f} + \lambda \cdot t_{\text{wait},f} + \mu \cdot C_{\text{charge},f} \right\}$

复杂度分析

复杂度分析

时间复杂度

- 路径评估： $O(n^2)$
- 选择操作： $O(N \log N)$
- 交叉操作： $O(n)$
- 变异操作： $O(n)$
- 整体算法： $O(G \cdot P \cdot n^2)$

空间复杂度

- 种群存储： $O(P \cdot n)$
- 辅助数组： $O(n^2)$
- 总空间： $O(P \cdot n + n^2)$

实际应用公式

成本计算实例

距离成本： $C_{\text{dist}} = \sum_{i=1}^{n-1} d_{i,i+1} \cdot c_{\text{fuel}}$

充电成本： $C_{\text{charge}} = \sum_{i \in \{\text{charging stations}\}} \Delta E_i \cdot c_{\text{electricity}}$

时间成本： $C_{\text{time}} = \sum_{i=1}^n t_i \cdot c_{\text{driver}}$

环保效益

碳排放减少： $\Delta \text{CO}_2 = \sum_{i=1}^n d_i \cdot (e_{\text{gasoline}} - e_{\text{electric}}) \cdot \text{conversion factor}$

参数推荐值

参数	推荐值	说明
种群大小	$P = 100$	平衡计算效率和解质量

参数	推荐值	说明
最大代数	$G = 500$	确保充分收敛
交叉率	$p_c = 0.8$	保持种群多样性
变异率	$p_m = 0.1$	避免早熟收敛
精英比例	$\epsilon = 0.05$	保留最优解
锦标赛大小	$t = 3$	中等选择压力
惩罚系数	$\alpha = 10^6$	强约束处理

公式使用说明

Mathpix兼容性

所有公式使用标准LaTeX语法，兼容Mathpix Markdown：

- 行内公式： $...$
- 行间公式：
$$...$$
- 对齐环境：
$$\begin{align} ... \end{align}$$

复制粘贴

可直接复制到支持LaTeX的编辑器中使用，包括：

- Typora
- VSCode + Markdown插件
- Jupyter Notebook
- Overleaf

文档版本：v2.0 更新时间：2024年 兼容：Mathpix Markdown