

求解不确定型车辆路径问题的弱鲁棒优化方法*

孙亮^{1,2}, 王冰¹, 郭栋¹, 徐艺¹

(1. 山东理工大学交通与车辆工程学院, 山东 淄博 255049;

2. 上海大学机电工程与自动化学院, 上海 200072)

摘要:为降低鲁棒优化模型最优解的保守性,以最小化违约车辆数和总惩罚成本为目标,建立针对旅行时间不确定的开放式车辆路径问题的弱鲁棒优化模型。对于不确定数据集的每个取值,该模型的最优解可以使其目标函数值始终不超过某数值,进而改善最优解的保守性。为提高启发式算法发现最优解的概率,提出一种自设计遗传算法对模型进行求解,其主要思想是利用粒子群算法搜索出可使遗传算法预期产生最好解的算法要素,并将其进行组合,从而产生新的遗传算法。采用新产生的遗传算法对模型继续求解,输出最好解。计算结果表明:与以往的鲁棒优化方法相比,弱鲁棒优化方法的最优解的保守性显著降低。

关键词:鲁棒优化;超启发式算法;遗传算法;车辆路径问题

中图分类号:TP301 **文献标志码:**A **文章编号:**1001-2486(2020)03-030-09

Light robust optimization approach for vehicle routing problem under uncertainty

SUN Liang^{1,2}, WANG Bing¹, GUO Dong¹, XU Yi¹

(1. School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255049, China;

2. School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China)

Abstract: Aiming to reduce conservatism of the optimal solution with regard to robust optimization model, a light robust optimization approach was proposed to solve the open vehicle routing problem with travel time uncertainty. This approach yields routes that minimize the weighted sum of the number of defaulted vehicles and the total penalty cost. For each realizations of the uncertain data set, the optimal solution of the approach can ensure that its optimal value never exceeds a certain value, thus improving the conservatism of the optimal solution. To improve the probability of finding the optimal solution, the automatic design of genetic algorithms was proposed to solve the model. Its main idea is to use the particle swarm optimization algorithm to search components of genetic algorithm which can expectedly enable the genetic algorithm to generate the optimal solution and then to combine these components to generate a new genetic algorithm to solve the model. The new genetic algorithm was used to solve the model continuously and give rise to a new optimal solution. Calculation results show that the conservatism of the optimal solution solved by the proposed light robust optimization approach is significantly reduced comparing with the past robust optimization method.

Keywords: robust optimization; hyper-heuristic algorithm; genetic algorithm; vehicle routing problem

开放式车辆路径问题(Open Vehicle Routing Problem, OVRP)指企业租用车辆来完成针对客户的配送任务,在满足一定约束条件下确定相应的车辆行驶路线以有序服务客户,实现决策目标最优化。企业所租用车辆从企业出发,完成配送任务后,不必返回企业。

从物流配送的实际营运过程看,影响第三方物流模式营运效果的因素主要有两个:①旅行时间的不确定性;②客户期望的服务时间段。因此,

针对不确定型 OVRP 优化方法的研究,对于提升不确定环境下第三方物流模式营运效果具有重要的理论和现实意义。

鲁棒优化方法^[1]采用不确定数据的边界特性描述模型参数的不确定性,有效避免了随机优化方法在阐述参数不确定性上过度依赖先验知识及服从概率分布假定的弊端。

目前,求解不确定型车辆路径问题(Vehicle Routing Problem, VRP)的鲁棒优化方法主要包含

* 收稿日期:2018-12-27

基金项目:国家自然科学基金资助项目(51508315);中国博士后面上基金资助项目(2018M642684);山东省自然科学基金资助项目(ZR2018PEE016)

作者简介:孙亮(1976—),男,山东即墨人,博士研究生,E-mail:carter0179@foxmail.com;
王冰(通信作者),女,教授,博士,博士生导师,E-mail:bw02116@sdut.edu.cn

两类:

1) 最坏场景鲁棒优化方法:该方法利用不确定集的边界值,将不确定的优化模型转化为确定的线性规划模型,发现一个对所有观测值可行的最优解,并确保最坏实现下的最优目标函数值达到最优。Sungur 等提出了需求不确定的 VRP 的最坏场景鲁棒优化模型,给出了三种需求限制约束的鲁棒对应式^[2]。Hu 等提出了描述需求和旅行时间不确定 VRP 特征的鲁棒优化模型,并分别使用变邻域搜索方法和分支定界法对各自提出的模型进行求解^[3]。Agra 等针对旅行时间不确定的 VRP,提出此类问题的鲁棒优化模型^[4]。刘洋等以总服务成本最小化为优化目标,建立了路径规划问题的鲁棒优化模型^[5]。Cao 等提出针对需求不确定的 OVRP 的鲁棒优化模型,使用禁忌算法对该模型进行求解^[6]。

2) 弱鲁棒优化方法:该方法是以最小的约束违背来保证目标函数值始终不超过合理范围,从而改善最坏值与预期值之间的偏差^[7]。Han 等提出了基于情景描述旅行时间不确定性的 VRP 的弱鲁棒优化模型^[8-9]。Wu 等针对旅行时间不确定的 VRP,提出了一个抑制目标函数恶化程度的弱鲁棒优化模型^[10]。

本文以旅行时间不确定的开放式车辆路径问题(Open Vehicle Routing Problems with Uncertain Travel time, OVRP-UT)为研究对象,提出一个描述该问题特征的弱鲁棒优化模型。为了提高启发式算法获取最优解的概率,在超启发式算法的框架下,通过引入预测适应度函数,提出一种自设计遗传算法对弱鲁棒优化模型进行求解。

1 OVRP-UT 的弱鲁棒优化模型

1.1 基本定义

定义 1 惩罚成本:如果配送车辆完成时间超过客户允许的最晚完成时间,根据合同向客户支付的罚金称为惩罚成本。

定义 2 违约车辆:产生惩罚成本的车辆称为违约车辆。

定义 3 关于旅行时间的不确定集

$$U_T = \{\tilde{t} \in \mathbf{R}^{|E|} \mid \tilde{t}_{ij} = \hat{t}_{ij} + \theta_{ij}\rho_{ij},$$

$$\sum_{i \in C} \sum_{j \in C} \rho_{ij} \leq \Lambda, 0 < \rho_{ij} \leq 1, \forall (i, j) \in E\}$$

其中: C 表示客户集(详见 1.2 节); \tilde{t} 表示关于路段的不确定旅行时间的向量, $\tilde{t} = (\dots, \tilde{t}_{ij}, \dots)$; \tilde{t}_{ij} 表示车辆把货物从客户 i 送到客户 j 所耗费旅行时

间的名义值; E 表示路径的集合,其含义详见 1.2 节; θ_{ij} 表示客户 i 到客户 j 的旅行时间相对于 \hat{t}_{ij} 的最大偏移量; ρ_{ij} 表示客户 i 到客户 j 的旅行时间的不确定水平, $\rho_{ij} \in (0, 1]$; Λ 表示旅行时间存在异常波动的路段数量, $\Lambda > 0$;车辆从客户 i 到客户 j 的旅行时间在 $[\hat{t}_{ij}, \hat{t}_{ij} + \theta_{ij}\rho_{ij}]$ 之间任意取值。

定义 4 极点: $\exists x \in U_T$, 对于 $\forall \delta > 0, x + \delta \notin U_T$, 则称 x 是 U_T 的极点。

定义 5 极点集: U_T 中由极点构成的子集,用 $ext(U_T)$ 表示。

定义 6 最坏实现: $ext(U_T)$ 中极点对应的模型参数的观测值称为最坏实现。

1.2 问题描述

一个完全的赋权图 $G = (V, E)$, 这里 $V = \{0, 1, 2, \dots, n\}$ 为点集, 0 表示车场, $C = \{1, 2, \dots, n\}$ 表示客户集, 客户的数量为 n ; $E = \{(i, j) \mid i, j \in V, i \neq j\}$ 为边集。每一条边赋有一个旅行时间, 旅行时间在 U_T 内任意取值。所有车辆具有相同的最大负载 Q 。每一个客户 i 赋有一个需求 $d_i, d_i \leq Q$ 。企业共租用 m 辆车来完成配送任务, 令 $M = \{1, \dots, m\}$ 。 B_i 是配送任务最晚完成时间, 如果配送任务完成时间超过 B_i , 则会产生惩罚成本。 ω_1 表示单位惩罚成本。

对于 U_T 内的任意实现, 该问题的优化目标是为每辆车确定合理的运输路线, 使违约车辆数和总惩罚成本的加权和达到最小。调度方案必须满足以下约束条件:

1) 调度方案对 U_T 内的任意实现均保持可行;

2) 车辆路线始于企业, 终止于某个客户, 服务完成后, 车辆不再返回企业;

3) 客户点需求必须且只能由一辆车来服务完成;

4) 每一条车辆路线上客户点的需求量之和不超过 Q 。

1.3 问题建模

1.3.1 决策变量

$$x_{ij}^k = \begin{cases} 1, & \text{车辆 } k \text{ 先访问客户 } i \text{ 再访问客户 } j \\ 0, & \text{其他} \end{cases}$$

$$s_k = \begin{cases} 1, & \text{车辆 } k \text{ 的配送完成时间超过 } B_i \\ 0, & \text{其他} \end{cases}$$

d_k^+ : 第 k 辆车配送任务完成时间超过 B_i 的偏差值。

s_{ik} : 第 k 辆车到达客户 i 的时刻。

1.3.2 弱鲁棒优化模型

本文提出的弱鲁棒优化模型, 是一个多目标

优化模型^[11]。考虑旅行时间的不确定性,引入关于车辆配送完成时间的正偏差变量 d_k^+ ,给出该问题的弱鲁棒优化模型 NLRO 如下:

$$\min f = \varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k \quad (1)$$

s. t.

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k = 1, \forall j \in C \quad (2)$$

$$\sum_{k=1}^m \sum_{j=0}^n x_{ij}^k = 1, \forall i \in C \quad (3)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \forall k \in M \quad (4)$$

$$\sum_{i=1}^n x_{i0}^k = 0, \forall k \in M \quad (5)$$

$$\sum_{i=0}^n x_{iu}^k - \sum_{j=1}^n x_{uj}^k = 0, \forall k \in M, \forall u \in C \quad (6)$$

$$\sum_{(i,j) \in S \times S} x_{ij}^k \leq |S| - 1, \forall S \subseteq V, 1 \leq |S| \leq n, \quad \forall k \in M \quad (7)$$

$$\sum_{i \in C} d_i \left(\sum_{j \in V} x_{ij}^k \right) \leq Q \quad (8)$$

$$\sum_{(i,j) \in E} \hat{t}_{ij} x_{ij}^k + \sum_{(i,j) \in \text{ext}(U_T)} \theta_{ij} \rho_{ij} x_{ij}^k - d_k^+ = B_f s_k - N(1 - s_k), \forall k \in M \quad (9)$$

$$\sum_{(i,j) \in E} \hat{t}_{ij} x_{ij}^k + \sum_{(i,j) \in \text{ext}(U_T)} \theta_{ij} \rho_{ij} x_{ij}^k - B_f \geq -N(1 - s_k), \quad \forall k \in M \quad (10)$$

$$\sum_{(i,j) \in E} \hat{t}_{ij} x_{ij}^k + \sum_{(i,j) \in \text{ext}(U_T)} \theta_{ij} \rho_{ij} x_{ij}^k - B_f < N s_k, \forall k \in M \quad (11)$$

$$s_{ik} + \sum_{(i,j) \in E} \hat{t}_{ij} x_{ij}^k + \sum_{(i,j) \in \text{ext}(U_T)} \theta_{ij} \rho_{ij} x_{ij}^k - M_1(1 - x_{ij}^k) \leq s_{jk}, \quad \forall k \in M \quad (12)$$

$$s_{ik} \geq 0, \forall i \in N, \forall k \in M \quad (13)$$

$$x_{ij}^k \in \{0, 1\}, \forall (i, j) \in E, \forall k \in M \quad (14)$$

$$d_k^+ \geq 0, \forall k \in M \quad (15)$$

$$s_k \in \{0, 1\}, \forall k \in M \quad (16)$$

目标函数(1)表示总惩罚成本和违约车辆数

的加权和, $\omega_1 \sum_{k \in M} d_k^+$ 表示总惩罚成本, $\sum_{k=1}^m s_k$ 表示违约车辆数, ε 为加权系数。约束(2) ~ (3) 确保每个客户只被一辆车服务;约束(4) ~ (5) 保证每一条线路只有一辆车从配送中心出发,但不返回配送中心;约束(6) 确保每条路径的连续性;约束(7) 表示避免出现子循环;约束(8) 表示的含义是每一条车辆路线上客户点的需求量之和不超过 Q ; 约束(9) 为弱鲁棒约束,表示对于 U_T 内的任意实现,每辆车的配送完成时间尽量不超过 B_f , N 是一个足够大的正数, N 的作用是确保正偏

差变量恒为正值,当 $s_k = 0$ 时,该可行解不会成为最优解;约束(10) ~ (11) 的作用是根据 s_k 的取值,正偏差变量在合理的范围内取值;约束(12) 表示对于 U_T 内的任意实现,车辆服务客户的顺序保持不变, M_1 是一个充分大的正数;约束(13) ~ (16) 表示决策变量的取值范围。

根据文献[8]的定义给出另一类弱鲁棒优化模型 LRO1 如下:

$$\min f = \sum_{k=1}^m s_k$$

约束条件与 NLRO 相同。

LRO1 的特点在于通过控制违约车辆数,间接实现对总惩罚成本的控制。不难看出, LRO1 是 NLRO 在 $\varepsilon = 0$ 时的特殊情形。

1.4 NLRO 的性质分析

性质1 Z_{NLRO} 表示 NLRO 的最优解对应的总惩罚成本, Z_{LRO1} 表示 LRO1 的最优解对应的总惩罚成本, $Z_{\text{NLRO}} \leq Z_{\text{LRO1}}$ 。

证明: 令 x_{RO} 是 LRO1 的最优解, x_{NLRO}^* 是 NLRO 的最优解。显然, x_{RO} 是 NLRO 的一个可行解。 $\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k$ 是 x_{RO} 的目标函数值, $\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^{*+} + \sum_{k=1}^m s_k^*$ 是 x_{NLRO}^* 的目标函数值。因此,

$$\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k \geq \varepsilon \cdot \omega_1 \sum_{k \in M} d_k^{*+} + \sum_{k=1}^m s_k^*$$

由于

$$\sum_{k=1}^m s_k \leq \sum_{k=1}^m s_k^*$$

所以

$$\omega_1 \sum_{k \in M} d_k^+ \geq \omega_1 \sum_{k \in M} d_k^{*+}$$

则 $Z_{\text{NLRO}} \leq Z_{\text{LRO1}}$ 。 □

2 求解方法

自设计遗传算法 (Automatic design of Genetic Algorithms, AGA) 分为两部分: ① 根据预期适应度函数值最小的原则, 筛选出有利于问题求解的遗传算法的要素; 根据适应度函数值最小的原则, 筛选出包含当前算法优化解的种群。② 将两者组合为一个新的遗传算法, 继续对问题进行求解, 输出最终算法优化解。

AGA 的算法流程如图 1 所示。

2.1 遗传算法的关键要素

1) 染色体编码方式: 染色体共由两部分组成, 第一部分采用基于客户的编码方式表示具体

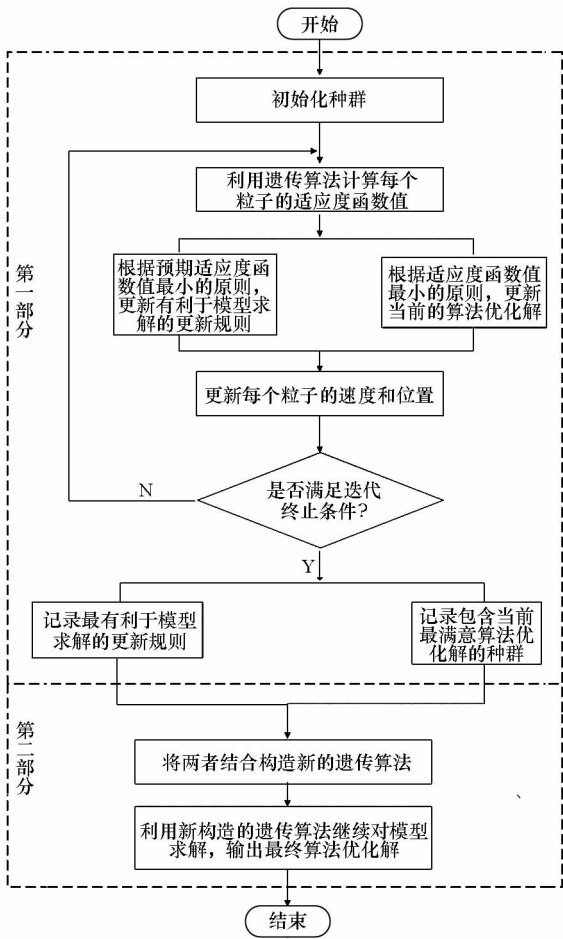


图1 AGA流程图

Fig.1 Flow chart of AGA

的调度方案,第二部分记录该调度方案的适应度函数值。染色体编码结构如图2所示。

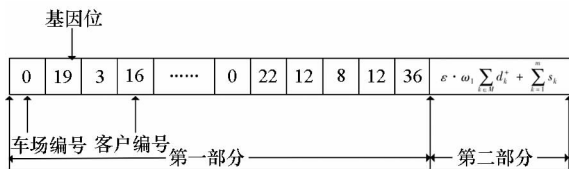


图2 遗传算法染色体编码结构示意图

Fig.2 Chromosome representation of genetic algorithm

2) 染色体解码的具体过程如下:随机生成新序列 $\{i_1, \dots, i_n\}$, 依次将该序列中的客户分配给第1辆车,直到第1辆车的剩余负载无法再为其他客户服务。重复上述过程,直到所有客户都能够被车辆服务。

3) 适应度函数的选取:目标函数(1)作为染色体的适应度函数。

2.2 粒子群算法的关键要素

2.2.1 粒子的编码与解码方法

粒子向量由位置向量、速度向量、适应度函数值三部分组成。

1) 位置向量:构造 1×8 维向量 X ,第1维表示选择因子集合 S 中的操作类型,第2维表示交叉因子集合 Cr 中的操作类型,第3维表示变异因子集合 Mu 中的操作类型,第4维表示遗传算法的种群规模 M_{ga} ,第5维表示遗传算法的迭代次数 T_{ga} ,第6维表示选择操作选中的个体数 i_k ,第7维表示交叉概率 p_c ,第8维表示变异概率 p_m 。遗传算子集合及其构成详见表1。

表1 遗传算子集合及其构成

Tab.1 Operator set of genetic algorithm

算子集合	算子类型
S	轮盘赌算子
	基于排名的选择算子
	锦标赛算子
Cr	部分映射交叉
	Non-abel 交叉算子
	单位位置交叉算子
	线性次序交叉算子
	基于位置的交叉算子
Mu	循环交叉算子
	互换变异算子
	逆序变异算子
	插入变异算子

注:不同算子的操作步骤,详见文献[12]。

2) 速度向量:速度向量也是一个 1×8 维向量,用 V 表示,微粒的运行速度限定在 $[V_{min}, V_{max}]$ 。

3) 适应度函数的选取。适应度函数由两部分组成: $(\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k)^*$ 表示该粒子构造的遗传算法所能获取的最小适应度函数值; $(\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k)^n$ 表示该粒子构造的遗传算法再

迭代 n 次 $(\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k)^*$ 的预测值。由于遗传算法的择优操作,不同代之间最优适应度函数值变化规律与指数变化规律具有一定的相似性,适合选用 $GM(1,1)$ 来作为预测函数。 $GM(1,1)$ 优点在于不需要很多的数据,就能解决序列完整性的问题。

令

$$(\varepsilon \cdot \omega_1 \sum_{k \in M} d_k^+ + \sum_{k=1}^m s_k)^n = (x^0(1) - \frac{b}{a})(1 - e^a)e^{-a(2n-1)} \quad (17)$$

式中, $x^0(i)$ 表示该粒子构造的遗传算法的第 i 次迭

代得到的最优适应度函数值。令 $x^0 = \{x^0(1), \dots, x^0(q)\}$, 对其做一次累加生成运算, 即

$$x^1(k) = \sum_{i=1}^k x^0(i), k = 1, 2, \dots, q$$

令

$$B = \begin{bmatrix} -\frac{1}{2}(x^1(1)+x^1(2)) & -\frac{1}{2}(x^1(2)+x^1(3)) & \dots & -\frac{1}{2}(x^1(q-1)+x^1(q)) \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$Y = (x^0(2) \quad x^0(3) \quad \dots \quad x^0(q))^T$$

则 a, b 可采用下式计算:

$$\begin{pmatrix} a \\ b \end{pmatrix}^T = (B^T B)^{-1} B^T Y \quad (18)$$

粒子的编码结构如图 3 所示。

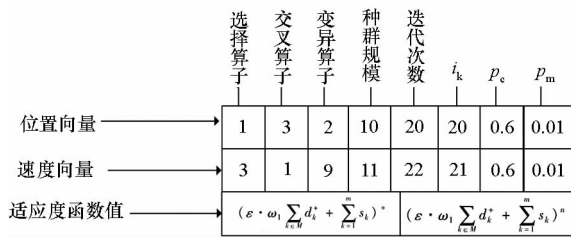


图 3 粒子编码结构示意图

Fig. 3 Particle representation

2.2.2 更新规则

速度和位置更新公式为:

$$V_i(t) = \omega_t \times V_i(t-1) + a_1 \times r_1 \times (X_i^P - X_i(t-1)) + a_2 \times r_2 \times (X^G - X_i(t-1)),$$

$$i = 1, 2, 3, 4, 5, 6, 7, 8 \quad (19)$$

$$X_i(t) = \text{mod}(\text{ceil}(X_i(t-1) + V_i(t)), D_{\max}^i) + 1,$$

$$i = 1, 2, 3 \quad (20)$$

$$X_i(t) = \text{ceil}(X_i(t-1) + V_i(t)), i = 4, 5, 6 \quad (21)$$

$$X_i(t) = \frac{(a_{\max}^i - a_{\min}^i)(\{X_i(t-1)\} + V_i(t))}{1 + V_{\max}} + a_{\min}^i,$$

$$i = 7, 8 \quad (22)$$

其中: $\text{ceil}(\cdot)$ 表示向上取整函数; $\text{mod}(\cdot)$ 表示取余函数; $\{\cdot\}$ 表示取实数的小数部分; D_{\max}^i 表示位置向量第 i 维的最大值 ($i = 1, 2, 3$); a_{\max}^i 表示位置向量第 i 维的最大值 ($i = 7, 8$); a_{\min}^i 表示位置向量第 i 维的最小值 ($i = 7, 8$); $X_i(t) = \{x_{i1}(t), \dots, x_{i8}(t)\}$ 为粒子 i 在第 t 代的位置向量; $V_i(t) = \{v_{i1}(t), \dots, v_{i8}(t)\}$ 为粒子 i 在第 t 代的速度向量; $X_i^P = \{x_{i1}^P, \dots, x_{i8}^P\}$ 为当前代最优粒子的位置向量; $X^G = \{x_1^G, \dots, x_8^G\}$ 为截至当前代最优粒子的位置向量; ω_t 表示第 t 代的惯性权重, 计算方式为

$$\omega_t = \omega_e - \frac{\omega_e - \omega_0}{T} t \quad (23)$$

其中, ω_0 为惯性因子的初始值, ω_e 为惯性因子的最终值, T 表示粒子群算法迭代的最大次数; r_1 和 r_2 为 $(0, 1)$ 之间的随机数; a_1 和 a_2 表示加速因子。

2.3 AGA 的算法步骤

AGA 的伪代码如算法 1 所示。

算法 1 AGA 的伪代码

Alg. 1 Pseudo-code of AGA

已知: $S, C, M, T, V_{\min}, V_{\max}$ 等

1. 随机生成遗传算法的初始化种群 $P(0) = \{H_{0,1}, \dots, H_{0,k}, \dots, H_{0,n}\}$;
2. 选取 $P(0)$ 中适应度函数值最小的染色体 H_0^{Ga} ; 令 $H_0^{Ga} = H^{Ga}$;
3. 随机生成高层粒子群算法的初始化种群 $Pop(0) = \{X_{0,1}, \dots, X_{0,k}, \dots, X_{0,n}\}$;
4. 确定 X_0^P, X^G ;
5. $t = 0$;
6. While($t \leq T$)

根据式(23)计算惯性因子 ω_t ;

应用粒子的解码算法^①求出 $Pop(t)$ 中每个粒子的预期适应度函数值和适应度函数值;

根据预期适应度函数值最小的原则, 更新 (X_i^P, X^G) ;

根据适应度函数值最小的原则, 记录当前粒子对应遗传算法的最优个体 H_{Ga}^* ;

根据式(19)~(22)更新粒子的速度和位置;

保留当前代 X^G ;

$t = t + 1$;

end

7. 以包含 H_{Ga}^* 的种群和 X^G 中位置向量构造新遗传算法;

8. 用新遗传算法继续求解获取算法优化解 H^{G*} ;

输出: H^{G*}

注: ①以粒子的位置向量表示的遗传算法参数作为输入, 采用标准遗传算法^[12]作为该粒子的解码算法。

2.4 AGA 的性质分析

性质 2 令 $P(A_m)$ 表示第 m 个粒子所构造的遗传算法能够在有限时间内发现模型最优解的概率, $P(A^{new})$ 表示新生成遗传算法在有限时间内能够发现模型最优解的概率,

$$B = \bigcup_{i=1}^{M \cdot T} A_i \cup A^{new}$$

表示 AGA 发现最优解的概率事件, 则 AGA 发现 NLRO 的最优解的概率满足如下性质:

$$P(B) > \max \{A_1, \dots, A_{M \cdot T}, A^{new}\}$$

证明:粒子群算法的搜索过程具有隐含并行性,不同粒子的解码过程彼此之间不互相影响,因此, A_m 和 A_j 是相互独立的。令 $P(A_i) = r_i, i = 1, \dots, M \cdot T + 1$, AGA 发现最优解的概率为:

$$P(B) = 1 - P(\overline{A_1 \cdots A_{M \cdot T} \cdot A^{new}}) \\ = 1 - \prod_{i=1}^{M \cdot T + 1} P(\overline{A_i})$$

又

$$r_i = 1 - (1 - r_i)$$

显然

$$\prod_{i=1}^{M \cdot T + 1} (1 - r_i) < (1 - r_i)$$

所以

$$1 - \prod_{i=1}^{M \cdot T + 1} (1 - r_i) > 1 - (1 - r_i) = r_i$$

从而,命题成立。□

性质3 AGA 最坏情况下的时间复杂度与遗传算法的时间复杂度相同。

证明:第 m 个粒子构造的遗传算法中选择、交叉、变异算子的最坏情况下的时间复杂度分别为 $O(G_1)$ 、 $O(G_2)$ 、 $O(G_3)$,种群规模为 M_{ga}^m ,迭代次数为 T_{ga}^m 。第 m 个粒子构造的遗传算法在最坏情况下的时间复杂度为 $O[M_{ga}^m \cdot T_{ga}^m \cdot (G_1 + G_2 + G_3)]$,因此,AGA 算法在最坏情况下的时间复杂度为 $O\{M \cdot T \cdot O[M_{ga}^m \cdot T_{ga}^m \cdot (G_1 + G_2 + G_3)]\}$,因为 M 和 T 是常数,所以 AGA 与遗传算法在最坏情况下时间复杂度相同。□

性质4 AGA 最坏情况下的空间复杂度为 $O(18M + L)$ 。

证明:就空间复杂度而言,每个粒子需要 18 个变量空间进行存储,另计算适应度函数值需要 L 个辅助变量空间,从算法 1 看,AGA 存储粒子和生成适应度函数值是重复利用的,因此 AGA 的空间复杂度为 $O(18M + L)$ 。□

3 仿真分析

3.1 仿真环境与参数设置

3.1.1 算例选取与基本参数设置

在对 NLRO 和 LRO1 的优化解进行分析时,通常可以借助文献[13]的标准测试数据(C1 ~ C14, O1 ~ O8, vrpncl ~ vrpncl4, F11, F12)。此外,还可利用文献[14]的标准测试数据(A - n34 - k5, B - n34 - k5)分析不确定参数对 NLRO 最优解的影响。NLRO 和 LRO1 中确定型参数的取值与标准测试数据中的对应取值相同。令

$\omega_1 = 1, \varepsilon = 1, \theta_{ij} = 3, \rho_{ij} = 1, \Lambda = 225$, 算法参数信息详见表 2。

表 2 算法参数设置

Tab. 2 Parameter settings for the algorithm

AGA 参数设置	取值
ω_0	0.8
ω_e	1.2
a_1	2.05
a_2	2.05
初始变异概率 p_{m0}	0.005
初始交叉概率 p_{c0}	0.4
初始选择个体数 i_{k0}	10
初始迭代次数 g_{iter}^0	10
初始种群规模 M_{ga}^0	30
粒子群规模 M	9
最大迭代次数 T	10
V_{min}	-10
V_{max}	10

遗传算法初始种群的生成方式:首先,随机生成一组满足约束(2) ~ (8)的调度方案,并记录该组可行解的每一辆车对应的最大配送完成时间 $Maxroutetime_k$ 。其中, B_f 采用如下方式生成:

$$B_f = 0.5 \times \max_{k \in M} \{maxroutetime_k\} \quad (24)$$

然后,根据 B_f 的值,随机生成满足约束(2) ~ (16)的调度方案,并作为遗传算法的初始种群。

3.1.2 测试环境

用 MATLAB R2016a 64 位实现了求解 NLRO 和 LRO1 的 AGA,在 Intel (R) Core (TM) i5 - 6200U CPU @ 2.3 GHz,8 GB 内存的计算机上进行了仿真实验。

3.2 算法优化解的性能分析

针对表 2 中的模型与算法参数信息,根据表 3 和表 4 中的 B_f ,使用 AGA 对模型 NLRO 和 LRO1 分别独立求解 30 次,分别取其中的最好解作为各自的算法优化解,该解对应的违约车辆数记录在表 3 和表 4 中的 veh_n 和 veh_nl ,总惩罚成本分别记录在 fc_n 和 fc_nl 这两列中。每一次的运算时间上限设定为 3600 s。表 3 和表 4 中, cum 列表示测试数据的客户数, $problem$ 列表示测试数据的名称。

表 3 C 类和 O 类问题优化性能分析
Tab. 3 Detailed results for C - class and O - class instances

problem	cum	B_t	NLRO		LRO1	
			veh_n	fc_n	veh_{nl}	fc_{nl}
C1	50	304.1	0	0	0	0
C2	75	177.3	7	171.0	4	270.9
C3	100	292.1	2	131.3	2	171.5
C4	150	322.6	4	130.7	3	151.4
C5	199	326.0	3	165.8	3	177.5
C6	50	117.2	6	509.0	5	528.3
C7	75	164.0	7	291.1	5	594.6
C8	100	556.7	0	0	0	0
C9	150	322.6	3	131.8	4	296.3
C10	199	289.4	5	144.3	5	475.2
C11	120	544.2	0	0	0	0
C12	100	239.5	0	0	0	0
C13	120	548.6	0	0	0	0
C14	100	273.4	0	0	0	0
O1	200	3834.3	0	0	0	0
O2	240	6378.8	0	0	0	0
O3	280	9435.0	0	0	0	0
O4	320	9435.0	0	0	0	0
O5	360	8131.0	0	0	0	0
O6	400	9687.1	1	281.0	1	2107.2
O7	440	9961.7	0	0	1	311.7
O8	480	9168.6	0	0	0	0

表 4 vrpnc 问题的优化性能分析
Tab. 4 Detailed results for vrpnc - class instances

problem	cum	B_t	NLRO		LRO1	
			veh_n	fc_n	veh_{nl}	fc_{nl}
vrpnc1	50	120.8	6	464.3	5	526.5
vrpnc2	75	156.3	0	0	5	841.3
vrpnc3	100	285.2	3	267.3	3	287.4
vrpnc4	150	330.9	2	179.4	3	153.6
vrpnc 5	199	312	7	373.5	7	373.5
vrpnc 6	50	164.2	3	153.1	2	383.7
vrpnc 7	75	149.5	7	433.4	6	569.4
vrpnc 8	100	345.7	0	0	0	0
vrpnc 9	150	338.1	1	26.0	1	178.4
vrpnc 10	199	290.2	8	471.3	7	993.9
vrpnc 11	120	538.0	0	0	0	0
vrpnc 12	100	244.1	0	0	0	0
vrpnc 13	120	579.5	0	0	0	0
vrpnc 14	100	262.7	0	0	0	0

针对表3和表4中 veh_n 和 veh_nl 两列数据,利用 Wilcoxon 秩和检验进行分析,在违约车辆数的优化能力方面,两者无显著差异 ($p > 0.05$)。对于表3和表4中 fc_n 和 fc_nl 这两列数据,利用 Wilcoxon 秩和检验进行分析,分析结果表明,NLRO 产生的惩罚成本显著小于 LRO1 ($p < 0.05$)。LRO1 单纯以违约车辆数最小为优化目标,可能存在多个违约车辆数相同的算法优化解,当迭代条件终止时,输出的不一定是总惩罚成本最小的算法优化解。与 LRO1 不同,NLRO 兼顾违约车辆数和总惩罚成本两个因素。因此,NLRO 算法优化解对应的总惩罚成本低于 LRO1。

表3和表4数据表明:对中小规模和大规模测试数据而言,AGA 均具有在有限时间内发现最优解的能力。根据性质2,使用 AGA 求解表2和表3中的算例对应的未发现最优解的概率的范围在 $[0.000\ 1, 0.000\ 7]$ 之间,因此,此时的算法优化解表现出了性质1所描述的最优解相类似的性质。

3.3 AGA 的求解性能分析

以文献[15]提到的超启发式遗传算法作为 AGA 的对比算法,记为 S-HPSO。使用两种算法对 NLRO 独立求解 30 次。S-HPSO 与 AGA 采用相同的初始染色体种群。S-HPSO 的相关参数设

置详见文献[15]。S-HPSO 与 AGA 遍历的可行解总数相同。其他参数设置详见第3.1小节。分别将两种算法对应的 30 次运算结果的平均值、标准差、最差适应度函数值、最好适应度函数值、运算时间的均值记录于表5中。表5中标有“*”的测试数据表示算法搜索到最优目标函数值为0的最优解。

在表5中, Avg_a 和 Avg_s 分别表示两种算法在 30 次运算结果中所有最优适应度函数值的平均值,这两列中的括号内数字表示该测试数据搜索到模型最优解的次数。 Std_a 和 Std_s 分别表示两种算法在 30 次运算结果当中所有最优适应度函数值的标准差。 T_{avg_a} 和 T_{avg_s} 分别表示两种算法对应的平均运行时间。

对表5中 Avg_a 和 Avg_s 这两列数据进行 Wilcoxon 秩和检验,结果表明 AGA 的平均求解性能显著优于 S-HPSO 的假设成立 ($p < 0.05$)。对表5中 T_{avg_a} 和 T_{avg_s} 这两列数据进行 Wilcoxon 秩和检验,结果表明两种算法的平均运行时间不存在统计学差异 ($p > 0.05$)。虽然 AGA 增加了利用预期适应度函数筛选更新规则环节,但这部分运算时间的增加不足以引起统计量秩次的变化,因此,两者在平均运算时间上不存在统计学差异。

表5 算法统计特性的比较与分析
Tab.5 Statistical performance of different algorithms

problem	cum	B_f	AGA			S-HPSO		
			Avg_a	Std_a	T_{avg_a}	Avg_s	Std_s	T_{avg_s}
C2*	76	237.7	15.2(18)	22.9	80.1	97.3(0)	51.7	99.1
C3*	101	373.8	0(27)	0	146.0	327.6(0)	170.0	165.9
C4*	151	377.8	94.9(3)	75.6	414.3	534.7(0)	298.1	390.1
C5	200	346.9	564.4	66.3	648.0	802.1	342.6	634.3
C6*	51	243.2	51.5(3)	26.3	52.5	98.2(0)	27.3	54.3
C7	76	203.7	218.2	28.5	98.4	423.4	41.0	103.4
C8	101	366.8	125.2	65.3	212.3	118.1	190.5	106.5
C9	151	342.6	610.6	261.4	356.3	941.6	153.3	371.8
C10	200	384.3	328.0	156.9	899.6	392.3	277.5	440.8
C11	121	316.9	284.7	80.5	236.0	496.4	1655.5	233.9
C12*	101	368.3	10.3(18)	26.3	205.6	192.6(12)	56.7	162.9
C13	121	356.6	733.5	1007.7	279.8	1863.2	1468.5	269.9
C14*	101	379.4	20.1(21)	775.1	208.7	126.4(12)	110.8	169.2
O5*	360	3848.6	14 133.8(21)	2428.0	1876.3	28 226.2(3)	25 962.9	2771
O6	400	3404.7	14 157.8	1825.7	7340.5	19 235.3	5467.6	6933.5

4 结论

对 OVRP-UT 建立弱鲁棒优化模型并求解, 核心内容如下: ①模型方面, 基于现有的弱鲁棒优化的框架构建了针对 OVRP-UT 的弱鲁棒优化模型; ②算法方面, 设计 AGA 求解 NLRO, 提高了启发式算法发现最优解的概率, 且所得算法优化解更接近于最优解。

目前求解不确定型车辆路径问题的鲁棒优化模型的启发式算法主要是元启发式算法, 下一步研究应关注如何使用超启发式算法求解不确定型车辆路径问题的鲁棒优化模型。

参考文献 (References)

- [1] Gabrel V, Murat C, Thiele A. Recent advances in robust optimization: an overview [J]. *European Journal of Operational Research*, 2014, 235(3): 471–483.
- [2] Sungur I, Ordóñez F, Dessouky M. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty [J]. *IIE Transactions*, 2008, 40(5): 509–523.
- [3] Hu C, Lu J, Liu X, et al. Robust vehicle routing problem with hard time windows under demand and travel time uncertainty [J]. *Computers and Operations Research*, 2018, 94: 139–153.
- [4] Agra A, Christiansen M, Figueiredo R, et al. The robust vehicle routing problem with time windows [J]. *Computers and Operations Research*, 2013, 40(3): 856–866.
- [5] 刘洋, 陈璐. 养护车辆路径规划的鲁棒性优化方法 [J]. *上海交通大学学报(自然版)*, 2018, 52(4): 388–394. LIU Yang, CHEN Lu. A robust optimization approach for the routing problem of road network daily maintenance [J]. *Journal of Shanghai Jiaotong University (Science)*, 2018, 52(4): 388–394. (in Chinese)
- [6] Cao E, Lai M Y, Yang H M. Open vehicle routing problem with demand uncertainty and its robust strategies [J]. *Expert Systems with Applications*, 2014, 41(7): 3569–3575.
- [7] Schöbel A. Generalized light robustness and the trade-off between robustness and nominal quality [J]. *Mathematical Methods of Operations Research*, 2014, 80(2): 161–191.
- [8] Han J, Lee C, Park S. A robust scenario approach for the vehicle routing problem with uncertain travel times [J]. *Transportation Science*, 2013, 48(3): 373–390.
- [9] 段征宇, 雷曾翔, 孙硕, 等. 随机时变车辆路径问题的多目标鲁棒优化方法 [J]. *西南交通大学学报*, 2019, 54(3): 565–572. DUAN Zhengyu, LEI Zengxiang, SUN Shuo, et al. Multi-objective robust optimization method for stochastic time-dependent vehicle routing problem [J]. *Journal of Southwest Jiaotong University*, 2019, 54(3): 565–572. (in Chinese)
- [10] Wu L, Hifi M, Bederina H. A new robust criterion for the vehicle routing problem with uncertain travel time [J]. *Computers and Industrial Engineering*, 2017, 112: 607–615.
- [11] 张骁雄, 葛冰峰, 谭跃进. 军事攻防中的多属性资源分配对策模型 [J]. *国防科技大学学报*, 2018, 40(5): 153–160. ZHANG Xiaoxiong, GE Bingfeng, TAN Yuejin. Multi-attribute game theoretic model for resource allocation in military attack-defense application [J]. *Journal of National University of Defense Technology*, 2018, 40(5): 153–160. (in Chinese)
- [12] 王凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003. WANG Ling. *Shop scheduling with genetic algorithms* [M]. Beijing: Tsinghua University Press, 2003. (in Chinese)
- [13] Li F Y, Golden B, Wasil E. The open vehicle routing problem: algorithms, large-scale test problems, and computational results [J]. *Computers and Operations Research*, 2007, 34(10): 2918–2930.
- [14] Solomon M M. Algorithms for the vehicle routing and scheduling problems with time window constraints [J]. *Operations Research*, 1987, 35(2): 254–265.
- [15] 韩亚娟, 彭运芳, 魏航, 等. 超启发式遗传算法求解带软时间窗的车辆路径问题 [J]. *计算机集成制造系统*, 2019, 25(10): 2571–2579. HAN Yajuan, PENG Yunfang, WEI Hang, et al. A hyper-heuristic genetic algorithm for vehicle routing problem with soft time windows [J]. *Computer Integrated Manufacturing Systems*, 2019, 25(10): 2571–2579. (in Chinese)