

基于蒙特卡洛相似度遗传算法的运输问题研究



李远锋 李章维 秦子豪 胡俊 张贵军

浙江工业大学信息工程学院 杭州 310023

(2111703057@zjut.edu.cn)

摘要 针对平衡运输问题,文中提出了一种基于蒙特卡洛相似度遗传算法的求解算法。首先,利用矩阵元素对种群个体进行初始化,增加了种群的多样性;其次,设计了动态变异率算子和随机变异策略,以增强算法的搜索能力,加快收敛速度;最后,采用蒙特卡洛相似度接收的方式,避免算法陷入局部最优解问题。通过收敛速度、最优解偏差率、相对标准差等参数对基本遗传算法 GA 和改进遗传算法 IGA 进行比较,验证了所提算法的有效性。针对杭州地理数据,设计开发了基于 ArcGIS 平台的运输配送系统,实现了平衡运输问题的求解功能,系统测试表明了所提算法的有效性。

关键词: 运输问题;遗传算法;交叉变异;蒙特卡洛相似度

中图分类号 TP301.6

Study on Transportation Problem Using Monte Carlo Similarity Based Genetic Algorithm

LI Yuan-feng, LI Zhang-wei, QIN Zi-hao, HU Jun and ZHANG Gui-jun

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract Aiming at the problem of balanced transportation, this paper proposes a Monte Carlo similarity based genetic algorithm. Firstly, the matrix elements are used to initialize the population, which increases the diversity of the population. Secondly, the dynamic mutation rate operator and the random mutation strategy are designed to enhance the search ability of the algorithm and accelerate the convergence speed. Finally, Monte Carlo similarity is adopted to avoid falling into the local optimal solution problem. The effectiveness of the algorithm is verified by the comparison of the convergence rate, the optimal solution deviation rate and the relative standard deviation by the basic genetic algorithm GA and the improved genetic algorithm IGA. According to the geographic data of Hangzhou, the transportation and distribution system based on ArcGIS platform is designed and developed to realize the function of solving the balanced transportation problem. The test results show the effectiveness of the proposed algorithm.

Keywords Transportation problem, Genetic algorithm, Cross variation, Monte Carlo similarity

1 引言

运输问题是现实生活中应用非常普遍的研究课题。在实际运输问题中,往往需要考虑综合调度、任务安排、天气等因素,其实质是一个复杂的调度问题。1941年, Hitchcock 首次提出了经典运输问题(Transportation Problem, TP),并在论文中证明了 TP 问题必有最优解^[1]。Koopmans 于 1949 年提出了 TP 问题的基本可行解与图论中树结构之间的关系,因此运输问题又被称为 Hitchcock-Koopmans 问题^[2]。Orlin 于 1993 年以最小费用流问题为背景,提出了多项式时间算法来求解 TP 问题^[3]。

运输问题是 NP-hard 问题^[4]。随着维数的增加,传统算法的时间复杂度和空间复杂度都会迅速增长,这给问题的求

解带了巨大的困难。因此,对于大规模运输问题,通常采用智能算法进行求解,如模拟退火、蚁群算法、禁忌搜索、神经网络等^[5]。Sharma 等提出了一种启发式方法,针对经典运输问题给出了一个良好的初始化解^[6]。Papamantou 等提出了单纯形算法与 TP 外部点算法相结合的策略来求解 TP 问题^[7]。Adlakha 等提出了一种简单的启发式算法来解决固定费用的运输问题,然而其求解时间比常规算法更长^[8]。Kannan 等讨论了基于生成树的非线性运输问题^[9]。2012 年, Kiraly 等实现了低成本流^[10]。Lotfi 等^[11]提出一种基于优先级编码的遗传算法来解决运输问题,并将该算法与最小生成树的遗传算法进行比较,证明其更有效率。Molla-Alizadeh-Zavardehi 等采用可变邻域搜索、模拟退火算法和混合 VNS 求解了可变参数的运输问题^[12]。Sabbagh 等提出了一种启发式精确混合算

到稿日期:2019-06-19 返修日期:2019-11-08 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61573317);浙江省教育厅一般科研项目(工程硕士专项)(Y201840644)

This work was supported by the National Natural Science Foundation of China (61573317) and General Project of Education Department of Zhejiang Province (Master of Engineering) (Y201840644).

通信作者:张贵军(zgj@zjut.edu.cn)

法,该算法基于改进的 Ford-Fulkerson 算法来解决平衡运输问题(Balance Transportation Problem, BTP)^[13]。Xie 等提出了一种高效的启发式算法来求解 TPVDS-A 问题^[14]。Kenan 等提出了 Karagul-Sahin 近似方法,使得求解时间显著缩短^[15]。

本文针对平衡运输问题,基于遗传算法,引入了矩阵元素随机初始化方案,设计了动态变异率算子和随机变异策略;构建了蒙特卡洛相似度算子,有效避免了算法陷入局部最优解;最后针对杭州市矢量数据地图,设计开发了基于 ArcGIS 平台的运输配送平台,实现了平衡运输问题的求解功能。系统测试表明了本文算法的有效性。

2 平衡运输问题的数学模型

平衡运输问题的描述如下:假定某种产品有 m 个产地 ($A_1, A_2, \dots, A_{m-1}, A_m$) 和 n 个销售地 ($B_1, B_2, \dots, B_{n-1}, B_n$)。其中,产地 A_i 的产量为 a_i ,销售地 B_j 的销量为 b_j 。在某一时间段 t ,以某一路径 s 从 A_i 运送一个单位货物到 B_j 的费用为 $c_{ij}(t, s)$,下文用 c_{ij} 表示。本文假设产销平衡,在满足约束的条件下,使得调度方案产生的费用最小。因此平衡运输问题可以表述为如下数学模型:

$$\begin{aligned} \min \quad & \text{tr}(\mathbf{C}^T \mathbf{X}) \\ \text{s. t.} \quad & \mathbf{X}\mathbf{E} = \mathbf{a} \\ & \mathbf{E}^T \mathbf{X} = \mathbf{b} \\ & \mathbf{X} \geq 0 \end{aligned} \tag{1}$$

其中, \mathbf{C} 为 $(c_{ij})_{m \times n}$ 的简写,表示 OD 费用矩阵; \mathbf{X} 为 $(x_{ij})_{m \times n}$ 的简写,表示决策变量矩阵,其中的元素 x_{ij} 表示产地 A_i 给 B_j 的供货量; $\text{tr}(\mathbf{C}^T \mathbf{X})$ 表示总成本; \mathbf{a} 为列向量,表示各个产地的产量,其中 a_i 是第 i 个产地的产量; \mathbf{b} 为横向量,表示各个销售地的销量,其中 b_j 是第 j 个销售地的销量; \mathbf{E} 是列向量,满足 $e_i = 1$ 。

3 基于改进遗传算法的运输问题求解

3.1 编码

用于求解运输问题的遗传算法,其种群可以用 $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ 表示,其中 \mathbf{X}_i 为运输问题的一个可行解,用如下形式表示:

$$\mathbf{X}_i = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix} \tag{2}$$

3.2 适应度函数的设计

遗传算法的适应度函数也称为评估函数,是作为评估群体中个体优劣程度的一种指标。适应度值越大,说明个体越好^[16]。本文采用如下适应度函数:

$$f(\mathbf{X}) = e^{-\lambda[\text{tr}(\mathbf{C}^T \mathbf{X}) - \omega]} \tag{3}$$

其中, ω 为目标函数的最小估计值, λ 为适应度膨胀系数。种群在迭代过程中,最优个体的适应度逐渐逼近于 1。

3.3 种群初始化

种群的初始化需要考虑初始化生成的个体严格满足约束条件,且生成的种群尽可能具有多样性,所以种群个体必须满足如下条件:

$$\sum_{j=1}^n x_{ij} = a_i, i \in \{1, 2, \dots, m\} \tag{4}$$

$$\sum_{i=1}^m x_{ij} = b_j, j \in \{1, 2, \dots, n\} \tag{5}$$

为了避免使用西北角法生成的种群个体单一的情况,本文摒弃西北角法,采用矩阵元素随机初始化方案。具体方案为:(1)从矩阵中随机选择矩阵元素 x_{ij} ,令 $x_{ij} = \min\{a_i, b_j\}$; (2)令 $a_i' = a_i - x_{ij}, b_j' = b_j - x_{ij}$;以此类推,将矩阵中所有的元素填满,求出一个基本可行解,并加入种群中。

3.4 交叉操作

在遗传算法中,交叉操作发挥着非常重要的作用,它是推动种群中染色体多样化的一种极其重要的手段,具有全局搜索的功能,是产生新个体的主要操作^[17]。交叉操作是根据交叉概率将种群中的两个父代个体按照某种交叉规则交换基因,以此产生新的子代个体^[18]。交叉得到的个体既要满足空间可达性,又要满足约束条件。假设 \mathbf{X}^1 和 \mathbf{X}^2 是两个交叉父本,交叉分 3 步完成。

(1)建立两个临时矩阵 $\mathbf{D} = (d_{ij})_{m \times n}$ 和 $\mathbf{R} = (r_{ij})_{m \times n}$ (见图 1),其中:

$$d_{ij} = (x_{ij}^1 + x_{ij}^2) / 2 \tag{6}$$

$$r_{ij} = (x_{ij}^1 + x_{ij}^2) \bmod 2 \tag{7}$$

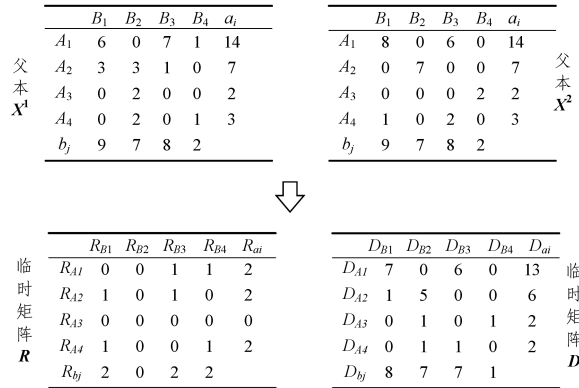


图 1 父本矩阵生成 \mathbf{R} 矩阵和 \mathbf{D} 矩阵

Fig. 1 Parent matrix generates \mathbf{R} matrix and \mathbf{D} matrix

(2)在满足式(9)和式(10)的约束条件下,将生成的 \mathbf{R} 矩阵分解为 \mathbf{R}^1 和 \mathbf{R}^2 (见图 2)。

$$\mathbf{R} = \mathbf{R}^1 + \mathbf{R}^2 \tag{8}$$

$$\sum_{j=1}^n r_{ij}^1 = \sum_{j=1}^n r_{ij}^2 = \frac{1}{2} \sum_{j=1}^n r_{ij} \tag{9}$$

$$\sum_{i=1}^m r_{ij}^1 = \sum_{i=1}^m r_{ij}^2 = \frac{1}{2} \sum_{i=1}^m r_{ij} \tag{10}$$

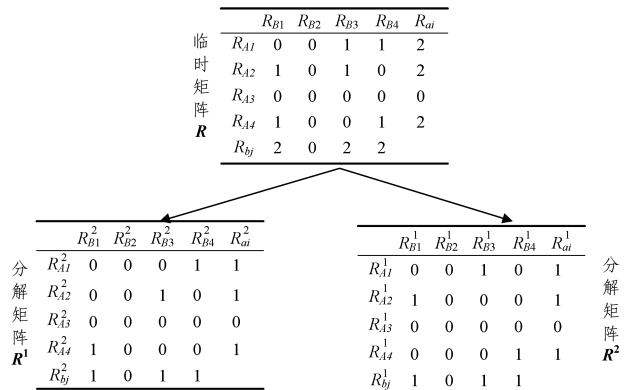


图 2 矩阵 \mathbf{R} 分解成矩阵 \mathbf{R}^1 和 \mathbf{R}^2

Fig. 2 Matrix \mathbf{R} is decomposed into matrices \mathbf{R}^1 and \mathbf{R}^2

(3)生成的两个后代 $\mathbf{X}_{\text{son}}^1$ 和 $\mathbf{X}_{\text{son}}^2$ (见图 3 和图 4)为:

$$\mathbf{X}_{\text{son}}^1 = \mathbf{D} + \mathbf{R}^1 \quad (11)$$

$$\mathbf{X}_{\text{son}}^2 = \mathbf{D} + \mathbf{R}^2 \quad (12)$$

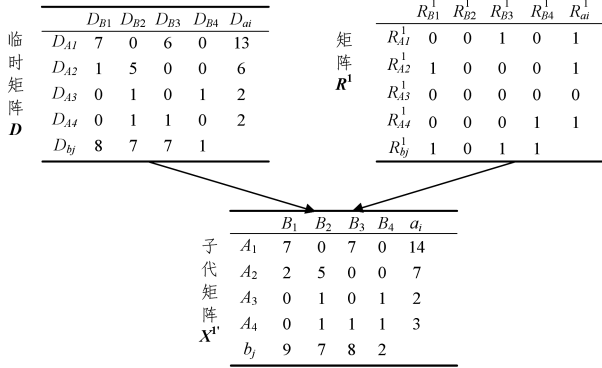


图 3 矩阵 \mathbf{D} 与矩阵 \mathbf{R}^1 生成子代 \mathbf{X}^1

Fig. 3 Matrix \mathbf{D} and matrix \mathbf{R}^1 generate child \mathbf{X}^1

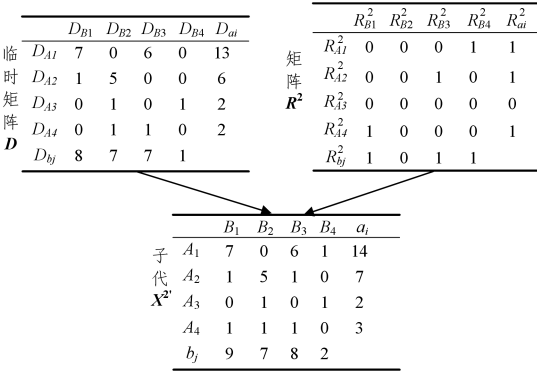


图 4 矩阵 \mathbf{D} 与矩阵 \mathbf{R}^2 生成子代 \mathbf{X}^2

Fig. 4 Matrix \mathbf{D} and matrix \mathbf{R}^2 generate child \mathbf{X}^2

3.5 变异操作

变异操作是推动种群进化的一个必要因素,可以增加种群的多样性,与交叉操作相互配合,兼顾全局和局部的搜索能力^[19]。合适的变异操作,可以减少交叉操作有用信息的流失,而且可以避免种群早熟。变异操作采用了动态变异率和随机变异策略。

3.5.1 自适应变异率

遗传算法参数中的变异概率 P_m 对算法的收敛性和搜索速度有着至关重要的影响。变异率 P_m 过小,不易产生新个体,可能导致局部最优;过大,遗传算法就变成了随机搜索算法^[20]。本文采用动态变异率,有效地平衡了上述问题。自适应变异率的公式如下:

$$P_m = kf(\mathbf{X}) = ke^{-\lambda[\text{tr}(\mathbf{C}^T \mathbf{X}) - \omega]} \quad (13)$$

随着适应度的不断上升,变异率线性增加,当适应度逐渐逼近 1 时,变异率趋于 1,加快了算法的收敛速度。

3.5.2 随机变异策略

随机变异策略分为 3 个步骤。

(1)选择子矩阵 $\mathbf{Y} = (y_{ij})_{p \times q}$ (见图 5)。对染色体对应的决策矩阵 $\mathbf{X} = (x_{ij})_{m \times n}$ 随机选 p 行和 q 行,建立矩阵 $\mathbf{Y} = (y_{ij})_{p \times q}$,其中 $\mathbf{Y} \subseteq \mathbf{X}$ 。 p 和 q 满足如下条件:

$$\begin{cases} p = \text{Random}(2, m), & p \in Z \\ q = \text{Random}(2, n), & q \in Z \end{cases} \quad (14)$$

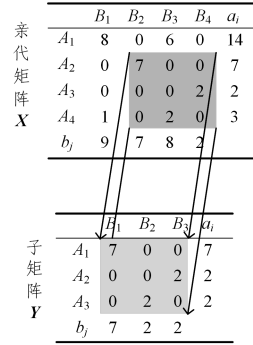


图 5 选择子矩阵 \mathbf{Y}

Fig. 5 Select Submatrix \mathbf{Y}

(2)由 $\mathbf{Y} = (y_{ij})_{p \times q}$ 生成新子矩阵 $\mathbf{Y}' = (y'_{ij})'_{p \times q}$ (见图 6)。

新子矩阵 $\mathbf{Y}' = (y'_{ij})'_{p \times q}$ 满足如下公式:

$$\begin{cases} a_i^{y'} = \sum_{j=1}^q y_{ij} = a_i^y \\ b_j^{y'} = \sum_{i=1}^p y_{ij} = b_j^y \end{cases} \quad (15)$$

$$\quad (16)$$

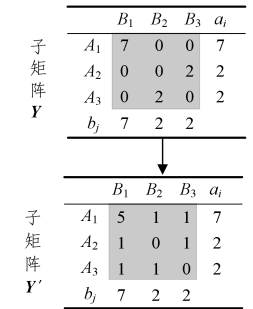


图 6 重新生成新矩阵

Fig. 6 Regenerating a new matrix

(3)用新生成的子矩阵 \mathbf{Y}' ,替换 \mathbf{Y}' 在 $\mathbf{X}_{m \times n}$ 位置对应的元素(见图 7)。

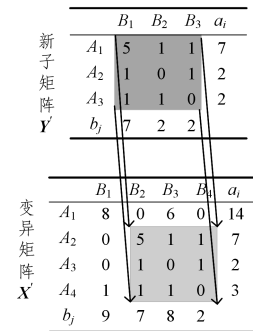


图 7 生成变异后矩阵 \mathbf{X}'

Fig. 7 Generation of the mutated matrix \mathbf{X}'

3.6 更新操作

更新操作是对种群去劣择优的过程。如果该染色体适应度比较高,该染色体被选择并遗传到下一代的概率就比较大。下面给出本文更新策略的具体步骤。

(1)将种群中适应度最高的前 10% 个体保留。

(2)两个父本个体交叉生成两个子代个体,其与生成子代个体进行适应度排序,适应度较好的前两个个体进行有概率的变异。

(3)对于变异后的个体与变异亲代个体,采用蒙特卡洛相似度的方式有概率地对差解进行接收,避免解陷入局部最优解问题^[21]。如果变异后的解的适应度更佳,则接收变异后的解;反之按照概率接收。接收概率由两部分构成:1)蒙特卡洛接收,即变异前与变异后的适应度相差越大,拒绝的概率就越大;2)相似度接收,即变异前与变异后的相似度越高,拒绝的概率就越大。假设 \mathbf{X} 为变异前的染色体, \mathbf{X}' 为变异后的染色体,具体选择方法如下。

1)如果 $f(\mathbf{X}) \leq f(\mathbf{X}')$,则接收变异后的染色体。

2)如果 $f(\mathbf{X}) > f(\mathbf{X}')$,有概率地接收变异后的染色体,具体接收概率如下:

$$p(\mathbf{X}) = e^{-[\frac{f(\mathbf{X}) - f(\mathbf{X}') + \cos(\mathbf{X}, \mathbf{X}')}{K} + \frac{1}{p}]} \quad (17)$$

其中, $\cos(\mathbf{X}, \mathbf{X}') = \frac{\sum_{i=1}^m \sum_{j=1}^n x_{ij} x'_{ij}}{\sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} \sqrt{\sum_{i=1}^m \sum_{j=1}^n x'_{ij}^2}}$, p 为总迭代次数中第 p 次迭代。

3.7 求解运输问题的流程

求解运输问题的具体流程如图 8 所示。

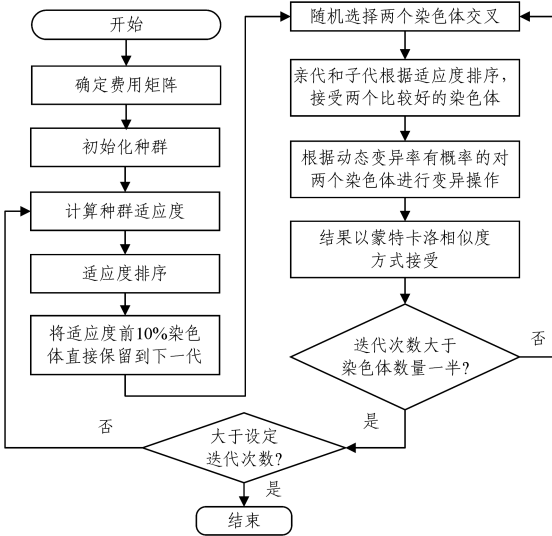


图 8 改进遗传算法的流程

Fig. 8 Flow chart of improved genetic algorithm

(1)对运行参数进行设置,如种群数量、迭代次数、调整系数、适应度膨胀系数等;

(2)确定费用矩阵;

(3)初始化种群;

(4)计算种群中每个染色体的适应度,并根据适应度对染色体进行排序;

(5)保留种群中适应度前 10%高的个体;

(6)两个父本个体交叉生成两个子代个体,其与子代个体进行适应度排序,适应度较好的前两个个体进行有概率的变异;

(7)变异后的个体与变异亲代个体采用蒙特卡洛-相似度的方式有概率地对差解进行接收。

4 算法测试及结果分析

地理数据文件以杭州市滨江区的矢量数据为基础,根据网络连通性、道路等级、道路车辆数量以及每条道路上的车辆行驶速度,来构建基于实际道路的杭州市某区的路网模型,并且利用 GIS 软件(ArcMap)的网络分析模块的接口,精确地得到各个产地到销售地的最短距离和最短时间(由于道路等级和车速不一样,最短时间和最短距离不呈线性关系),同时建立最短路径(s)和最短时间(t)的 OD 矩阵,为所提算法奠定基础。

图 9 显示了 5 个产地、8 个销售地的分布情况(旗帜代表产地,星形代表销售地),以及杭州市滨江区的道路状况。通常情况下,道路越宽,车辆行驶速度越快,相应的配送时间越短,费用越低。图 10 为产地和销售地之间的 OD 费用图。表 1 表示产地和销售地之间的 OD 费用矩阵。图 11 为本文算法生成的产地和销售地之间的最优配送图。

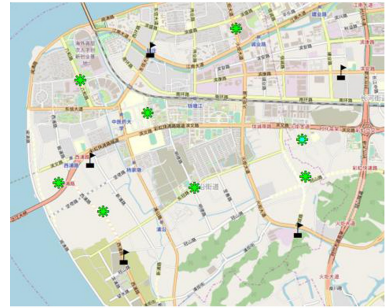


图 9 产地和销售地分布图

Fig. 9 Distribution map of origin and sales

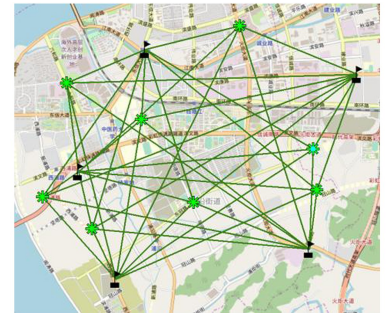


图 10 产地和销售地 OD 费用图

Fig. 10 OD diagram of origin and sales



图 11 本文算法生成的产地和销售地之间的最优配送图

Fig. 11 Optimal distribution map between place of origin and place of sale generated by proposed algorithm in this paper

表 1 OD 费用矩阵
Table 1 OD cost matrix

Name	OD	Name	OD	Name	OD	Name	OD
产地 1-销地 1	378	产地 3-销地 5	2807	产地 2-销地 8	2495	产地 5-销地 4	1112
产地 1-销地 2	3820	产地 3-销地 6	8332	产地 3-销地 1	4987	产地 5-销地 5	9442
产地 1-销地 3	4275	产地 3-销地 7	10974	产地 3-销地 2	3892	产地 5-销地 6	2453
产地 1-销地 4	7036	产地 3-销地 8	1946	产地 3-销地 3	1193	产地 5-销地 7	1298
产地 1-销地 5	3403	产地 4-销地 1	7661	产地 3-销地 4	11683	产地 5-销地 8	9880
产地 1-销地 6	3685	产地 4-销地 2	11104	产地 2-销地 3	2911	产地 4-销地 7	1993
产地 1-销地 7	6327	产地 4-销地 3	11558	产地 2-销地 4	8535	产地 4-销地 8	11125
产地 1-销地 8	3841	产地 4-销地 4	247	产地 2-销地 5	4269	产地 5-销地 1	6417
产地 2-销地 1	1839	产地 4-销地 5	10686	产地 2-销地 6	5184	产地 5-销地 2	9859
产地 2-销地 2	4686	产地 4-销地 6	3698	产地 2-销地 7	7826	产地 5-销地 3	10314

5 算法性能分析

为了验证所提算法的有效性,在双核 CPU(I5-4200U)、主频 1.6GHz、8GB 内存的笔记本上进行实验测试。操作系统为 Windows8.1 专业版,基于 Java 语言的 JDK1.8 版本进行相关的算法实验。算法分析中使用的绘图工具是 Origin-Pro2017 版本,算法代码采用 IntelliJ IDEA 2018.3.5 编写。设置算法的迭代次数为 5000;种群规模为 10 个、15 个、20 个、25 个、30 个;每次将适应度前 0.1 的染色体保留到下一代;动态变异公式中调整系数 k 取 1;蒙特卡洛中 K 值取 4;目标函数的最小估计值取 250000,适应度膨胀系数 λ 取 1×10^{-4} 。

将改进遗传算法(IGA)和基本遗传算法(GA)的性能进行对比。IGA 算法和 GA 算法的种群规模都取 25,其余参数同上文,各自实验 30 次,取每代的最优值进行性能比较,算法收敛曲线如图 12 所示。可以看到,IGA 算法和 GA 算法都呈现指数收敛,但是 IGA 算法收敛到 253786,与已知最优解 250345 的偏差率为 0.0137;而 GA 算法仅仅收敛到 308884,与已知最优解 250345 的偏差率为 0.2338。因此,IGA 算法更有效。

表 2 GA 算法与 IGA 算法 30 次实验的最优解、最优解平均值及偏差率

Table 2 Optimal solution, optimal solution mean and deviation rate of GA and IGA 30 experiments

编号	种群数量	已知最优解	GA				IGA			
			30 次实验最优解	30 次实验最优解与已知最优解偏差率	30 次实验最优解平均值	30 次实验最优解平均值与已知最优解偏差率	30 次实验最优解	30 次实验最优解与已知最优解偏差率	30 次实验最优解平均值	30 次实验最优解平均值与已知最优解偏差率
1	10	250345	330322	0.3194671	352472.2	0.4079458	266817	0.0657972	272037.8	0.0866516
2	15	250345	318963	0.2740937	327591.4	0.3085597	258047	0.0307655	267480	0.0684455
3	20	250345	292752	0.1693942	306946.2	0.2260927	260171	0.0392498	263840	0.0539056
4	25	250345	308884	0.2338333	314939.4	0.2580215	253786	0.0137450	258868	0.0340450
5	30	250345	302585	0.2086720	307536.2	0.2284495	262082	0.0468833	265886.4	0.0620799

为了进一步验证算法的稳定性,本文对 IGA 算法和 GA 算法的 30 次实验结果取平均值,进行稳定性比较。将 30 次实验中 IGA 算法和 GA 算法每次迭代的最优值和平均值进行对比,如图 13 所示。可以得出,IGA 算法的收敛效果更佳。通过表 3 可以得出,GA 算法的相对标准差在 0.013~0.042 之间波动,而 IGA 算法的相对标准差在 0.016~0.026 波动,因此 IGA 算法的稳定性更好。

将 IGA 算法和 GA 算法的种群规模都设置为 25,分别实验 30 次,求各自每代的适应度平均值,绘制适应度曲线,如图 14 所示。

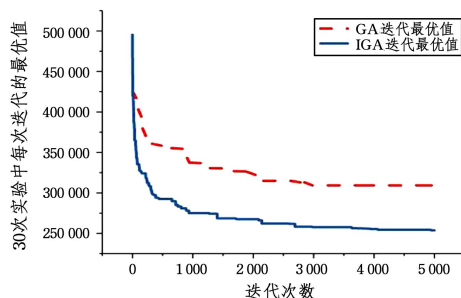


图 12 30 次实验中 GA 算法和 IGA 算法每次迭代的最优值
Fig. 12 Optimal value of each iteration of improved genetic algorithm and common genetic algorithm in 30 experiments

表 2 列出在已知最优解 250345 的情况下,种群数量为 10,15,20,25,30 时,通过 30 次实验得出的 GA 算法与 IGA 算法的最优解、最优解平均值及偏差率。GA 算法在不同种群数量下,最优解与已知最优解的偏差率在 0.169~0.319 之间;IGA30 算法在不同种群数量下,最优解与已知最优解的偏差率在 0.0137~0.0469 之间;IGA 算法的最优解平均值与已知最优解的偏差率远远小于 GA 算法。

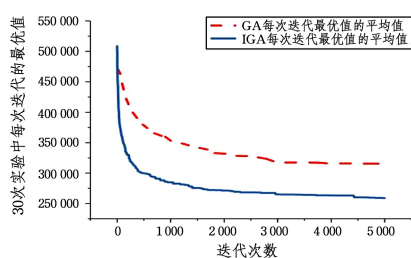


图 13 30 次实验中 GA 算法和 IGA 算法每次迭代最优值的平均值
Fig. 13 Average value of optimal value of each iteration of GA algorithm and IGA algorithm in 30 experiments

表3 GA算法与IGA算法的相对标准差

Table 3 Relative standard deviation between GA algorithm and IGA algorithm

种群数量	GA算法的相对标准差	IGA算法的相对标准差
10	0.039735821	0.016313415
15	0.023380406	0.026190034
20	0.042192573	0.022135806
25	0.026337978	0.020768845
30	0.013495329	0.016421668

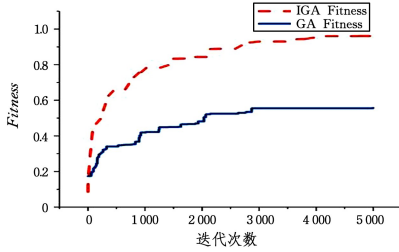


图14 GA算法和IGA算法的适应度曲线

Fig. 14 Fitness curve of GA algorithm and IGA algorithm

可以看出,IGA算法的适应度在5000次迭代中逐渐逼近1;而GA算法的适应度值上升比较慢,最后在5000代以内仅达到0.5。

表4 GA算法与IGA算法在不同规模下实验30次的最优解、最优解平均值及偏差率

Table 4 Optimal solution, optimal solution mean and deviation rate of 30 experiments under different scales of GA algorithm and IGA algorithm

编号	产地数量	销售地数量	种群数量	GA			IGA		
				30次实验最优解与已知最优解偏差率	30次实验最优解平均值与已知最优解偏差率	相对标准差	30次实验最优解与已知最优解偏差率	30次实验最优解平均值与已知最优解偏差率	相对标准差
1	5	8	20	0.1693942	0.2260927	0.0421925	0.0392498	0.0539056	0.0221358
2	5	50	20	0.1736521	0.2321452	0.0369523	0.0365459	0.0632448	0.0236544
3	5	100	20	0.1556849	0.2436412	0.0452612	0.0256547	0.0639741	0.0261289
4	10	100	20	0.1813624	0.2652226	0.0234587	0.0145697	0.0439412	0.0196651



图15 增删产地或销售地页面

Fig. 15 Adding or deleting origin or sales page



图16 实际配送路线图

Fig. 16 Actual delivery route map



图17 OD费用矩阵图

Fig. 17 OD cost matrix

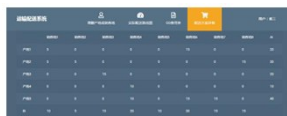


图18 配送方案详情

Fig. 18 Distribution plan details

本文在产地数量为5、销售地数量为8的基础上,进一步验证了产地数量为5、销售地数量为50,产地数量为5、销售地数量为100,以及产地数量为10、销售地数量为100的情况。从表4可以看出,IGA 30次实验的最优解与已知最优解的偏差率在0.014~0.039之间,而GA 30次实验的最优解与已知最优解偏差率在0.1之上,精度提高了将近10倍;IGA 30次实验最优解的平均值与已知最优解偏差率在0.043~0.063之间,而GA 30次实验最优解的平均值与已知最优解的偏差率在0.2左右;IGA的相对标准差在0.0197~0.0261之间,而GA的相对标准差在0.0234~0.045之间,因此IGA更加稳定。

最后,基于J2EE规范,前端采用Bootstrap和Vue框架、后端采用SpringBoot设计并开发了运输配送系统。如图15所示,通过“增删产地或销售地”菜单,可以对产地和销售地的数量进行删除或者增加。

如图16所示,点击“实际配送路线图”菜单,可以得到货物配送的实际路线。如图17所示,通过“OD费用表”菜单,可以获得产地和销售地之间的OD矩阵。如图18所示,点击“配送方案详情”菜单,可以得到每个产地到销售地的配送数量。

结束语 本文针对平衡运输问题进行了研究,在基本遗传算法的基础上,采用随机初始化方案对矩阵进行初始化;在选择策略上加入蒙特卡洛相似度策略,有效地抑制了种群早熟,避免算法陷入局部最优解;最后采用了动态变异率策略,加快了算法的收敛速度。实验结果表明,对于规模较大的平衡运输问题,改进遗传算法比基本遗传算法更有效。

参考文献

[1] HITCHCOCK F L. The distribution of a product from several sources to numerous localities[J]. Journal of Mathematics and Physics, 1941, 20(1/2/3/4): 224-230.

[2] KOOPMANS T C. Optimum utilization of the transportation system[J]. Econometrica: Journal of the Econometric Society, 1949, 17: 136-146.

[3] ORLIN J B. A faster strongly polynomial minimum cost flow algorithm[J]. Operations research, 1993, 41(2): 338-350.

[4] KLOSE A. Algorithms for solving the single-sink fixed-charge transportation problem[J]. Computers & Operations Research, 2008, 35(6): 2079-2092.

[5] ZHU L L. Intelligent Optimization Algorithm for Upper Minimal

- Total Cost Bound of Transportation Problem with Varying Demands and Supplies[D]. Nanchang: Nanchang University, 2015.
- [6] SHARMA R R K, PRASAD S. Obtaining a good primal solution to the uncapacitated transportation problem[J]. *European Journal of Operational Research*, 2003, 144(3): 560-564.
- [7] PAPAMANTHOU C, PAPARRIZOS K, SAMARAS N. Computational experience with exterior point algorithms for the transportation problem[J]. *Applied Mathematics and Computation*, 2004, 158(2): 459-475.
- [8] ADLAKHA V, KOWALSKI K, VEMUGANTI R R. Heuristic algorithms for the fixed-charge transportation problem [J]. *OPSEARCH*, 2006, 43(2): 132-151.
- [9] KANNAN G, KUMAR P S, VINAY V P. Comments on nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm by Jung-Bok Jo, Yinzhen Li, Mitsuo Gen, Computers & Industrial Engineering (2007)[J]. *Computers & Industrial Engineering*, 2008, 55(2): 533-534.
- [10] KIRALY Z, KOVACS P. Efficient implementations of minimum-cost flow algorithms[J]. *Acta Universitatis Sapientiae, Informatica*, 2012, 4(1): 67-118.
- [11] LOTFI M M, TAVAKKOLI-MOGHADDAM R. A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems[J]. *Applied Soft Computing*, 2013, 13(5): 2711-2726.
- [12] MOLLA-ALIZADEH-ZAVARDEHIS, NEZHADSS, TAVAKKOLI-MOGHADDAM R, et al. Solving a fuzzy fixed charge solid transportation problem by metaheuristics[J]. *Mathematical & Computer Modelling*, 2013, 57(5/6): 1543-1558.
- [13] SABBAGH M S, GHAFARI H, MOUSAVI S R. A new hybrid algorithm for the balanced transportation problem[M]// Pergamon Press, 2015.
- [14] XIE F, BUTT M M, LI Z, et al. An upper bound on the minimal total cost of the transportation problem with varying demands and supplies[J]. *Omega*, 2017, 68: 105-118.
- [15] KARAGUL K, SAHIN Y. A Novel Approximation Method to Obtain Initial Basic Feasible Solution of Transportation Problem [J]. *Journal of King Saud University-Engineering Sciences*, 2020 (32): 211-218.
- [16] LIU Y M, WEI O, HUANG M Y, et al. Feature Selection Optimization Based on Atomic Set and Genetic Algorithm in Software Product Line [J]. *Journal of Chinese Computer Systems*, 2017, 38(1): 35-39.
- [17] YANG X W, YANG L J. An Improved Genetic Algorithm Based on Crossover Model [J]. *Control and Decision*, 2016, 31(10): 1837-1844.
- [18] HAN K K, XIE Z P, LV X. Fog Computing Task Scheduling Strategy Based on Improved Genetic Algorithm [J]. *Computer Science*, 2018, 45(4): 137-142.
- [19] WU X L, XIAO X, ZHAO N. Flexible Job Shop Dual Resource Scheduling Problem Considering Loading and Unloading [J]. *Control and Decision*, 2020, 35(10): 2475-2485.
- [20] SHI T F. Research on Path Planning for Mobile Robot Based on Improved Genetic Algorithm [J]. *Computer Simulation*, 2011, 28(4): 193-195.
- [21] WEN Y, PAN D Z. Improved Genetic Algorithm for Traveling Salesman Problem[J]. *Computer Science*, 2016, 43(S1): 90-92.



LI Yuan-feng, born in 1993, postgraduate. His main research interests include intelligent information processing and so on.



ZHANG Gui-jun, born in 1974, is a member of China Computer Federation. His main research interests include intelligent information processing, optimization theory and algorithm design and bioinformatics.