

改进樽海鞘算法求解带时间窗的应急选址路径问题

徐帆, 马良, 张惠珍*, 陈曦
(上海理工大学 管理学院, 上海 200093)

摘要: **目的** 为使应急物资及时高效地送到灾区, 针对多目标应急选址-路径问题, 在考虑灾区的时间窗及物资运输过程中道路安全的情况下, 以最小化经济成本、最小化时间惩罚成本及最大化道路安全性为目标, 构建多目标优化模型。同时, 设计改进的樽海鞘算法求解问题, 以验证模型的可行性和算法的有效性。**方法** 根据模型的特征对樽海鞘算法进行改进, 运用随机生成和贪心算法相结合的方式生成初始解, 利用交叉算子和邻域搜索算子改进原始算法的位置更新操作, 引入非支配排序遗传算法 (NSGA-II) 的精英保留策略, 以提高算法的性能。**结果** 经过多个算例测试, 该算法能快速获得一簇 Pareto 解, 与基本樽海鞘算法进行对比后可知, 改进后的算法性能更优越。**结论** 对于灾后及时响应的应急选址路径问题, 采用改进的樽海鞘算法具有一定优越性, 并在多个目标权衡的情况下, 可供决策者根据目标的偏好找到较满意的解, 对于研究应急选址路径问题具有一定的参考价值。

关键词: 选址-路径问题; 应急物资; 时间窗; 改进樽海鞘算法

中图分类号: TP301.6; TB485.3 文献标志码: A 文章编号: 1001-3563(2024)05-0220-10

DOI: 10.19554/j.cnki.1001-3563.2024.05.027

Improved Salp Swarm Algorithm for Solving Multi-objective Emergency Location Routing Problem with Time Windows

XU Fan, MA Liang, ZHANG Huizhen*, CHEN Xi

(School of Management, University of Shanghai for Science & Technology, Shanghai 200093, China)

ABSTRACT: In order to ensure timely and efficient delivery of emergency resources to disaster areas, the work aims to construct a multi-objective optimization model for the multi-objective emergency location routing problem by taking into account the time windows of the disaster area and road safety during resources transportation, with the objectives of minimizing economic costs, time penalty costs, and maximizing road safety. At the same time, an improved salp swarm algorithm is designed to solve the problem, in order to verify the feasibility of the model and the effectiveness of the algorithm. Based on the characteristics of the model, the algorithm was improved by combining random generation and greedy algorithm to generate initial solutions. The position update operation of the original algorithm was improved by crossover operators and neighborhood search operators. The elite retention strategy of NSGA-II was introduced to improve the performance of the algorithm. After test of multiple examples, this algorithm could quickly obtain a cluster of Pareto solutions and was compared with the original salp swarm algorithm. The improved algorithm had better performance. For the emergency location routing problem of timely response after a disaster, the improved salp swarm algorithm has certain advantages, and can provide decision-makers with satisfactory solutions based on the preferences of multiple objectives. It has a certain reference value for the field of emergency location routing problems.

KEY WORDS: location routing problem; emergency resources; time windows; improved salp swarm algorithm

收稿日期: 2023-05-11

基金项目: 国家自然科学基金 (72101149); 教育部人文社会科学基金 (21YJC630087); 国家外国专家项目 (G2023013029)

*通信作者

近年来,地震、洪涝等自然灾害突发事件频繁发生。根据中国应急管理部与自然资源部等部门核定公布的数据显示,2022 年我国自然灾害以洪涝、干旱、地震为主,共计造成 1.12 亿人次受灾,直接经济损失高达 2 386.5 亿元(<http://www.mem.gov.cn/>)。在应急管理中,救援仓库的设施选址问题(Facility Location Problem, FLP)与车辆路径问题(Vehicle Routing Problem, VRP)是很重要的 2 个子问题,它们整合了战略和战术决策^[1]。将这 2 个子问题的综合问题称为选址-路径问题(Location-Routing Problem, LRP)。

目前,有大批学者在应急物流选址-路径上进行了相关研究。Vahdani 等^[2]考虑了具有不同容量水平的配送中心和仓库位置,以及仓库的库存和路线的可靠性,构建了一个两阶段多目标的混合整数模型。Zhang 等^[3]建立了一个考虑出行时间、应急救援费用和二氧化碳排放的不确定性多目标应急响应选址规划模型,并将不确定性仿真与遗传算法结合对模型进行求解。Wei 等^[4]考虑时间窗和成本,建立了带时间窗的应急选址-路径多目标模型,并利用混合蚁群算法进行求解。Shen 等^[5]基于粒子群算法结合禁忌搜索的混合两阶段算法,求解了一种模糊低碳应急选址-路径问题。刘长石等^[6]针对震后物资配送模糊选址-路径问题,设计了一种混合免疫遗传算法,以应对震后应急。

现有研究针对应急 LRP 问题重点考虑的是经济因素和时间因素,而实际情境中灾后设施与道路可能会遭到一定破坏,在山区等环境恶劣地区这些因素会对救援人员的生命造成威胁,因而应该考虑物资配送过程中的安全性。文中引入道路安全性系数,构建一个多目标带时间窗的灾后选址-路径模型。

由于 LRP 问题属于 NP-hard 问题^[7],精确算法在求解大规模 LRP 问题时其速度通常较缓慢,甚至无法求到一个可行解,采用智能优化算法可以在短时间内找到满意解^[8]。樽海鞘算法(Salp Swarm Algorithm, SSA)^[9]是根据樽海鞘在海洋中移动和觅食时的群体聚集行为提出的群智能优化算法,相较于遗传算法、蚁群算法等经典算法,SSA 算法的参数较少,具有结构简单、收敛速度快、控制参数少等优点,自开发以来广泛应用于特征选择^[10-11]、图像^[12]、工程设计^[13-14]等多个领域,但很少应用于应急物资调度领域内的选址-路径等离散型问题中。文中基于 SSA 算法求解所构建的多目标 LRP 问题模型,并根据 LRP 模型的特征对 SSA 算法进行改进。通过不同规模的算例测试对构建的模型和算法进行验证,并与基本樽海鞘优化算法(Salp Swarm Algorithm, SSA)的求解结果进行对比,以验证算法的可行性和有效性。该算法在得到帕累托前沿面的同时,可供决策者根据不同目标的重要性选择恰当的物资调度方案,对应急 LRP 领域具有一定的参考意义。

1 问题描述和模型建立

1.1 问题描述与模型假设

描述研究的 LRP 问题:给定 m 个候选应急仓库和 n 个需求点,从候选应急仓库中选择若干条设施开放,并规划配送物资的路径,求解目标为最小化总成本、最小化物资延时送达的时间惩罚成本和最大化物资运输过程中的道路安全性。针对模型做如下假设:候选应急仓库与需求点地理位置关系已知;物资到达时间已知;车辆行驶的路网状况已知;需求点的物资需求量已知;应急仓库到需求点及需求点两两之间均存在可通行路径;物资送达的时间应越早越好,因此每个需求点的时间窗为半时间窗,晚到则会产生相应的时间惩罚成本。

1.2 符号与变量

集合符号说明如下: I 为候选救援仓库集合($1, 2, \dots, m$); J 为应急物资需求点集合($1, 2, \dots, n$); V 为应急物流系统所有节点集合, $V = I \cup J$, $i, j \in V$; K 为救援车辆集合, $k \in K$ 。参数符号说明: f_i 为救援仓库开放的固定成本; f_k 为使用车辆 k 的启用成本; c_{ij} 为 i 点到 j 点路径所消耗的单位时间运输费用; Q_i 为救援仓库 i 的储存容量; m_k 为第 k 辆车的容量; d_j 为需求点 j 的需求量; t_{ij} 为 i 点到 j 点的时间; d_{ij} 为 i 点到 j 点的距离; s_i 为需求点 i 的服务时间; o_j 为 j 点的软时间窗最晚到达时间; β 为每单位时间违反时间窗的惩罚成本; v_{ij} 为从车辆 i 点到 j 点的速度; t_{jk} 为车辆 k 在访问 j 点的开始时间; G_{jk} 为车辆 k 在访问 j 点违反时间窗口的长度; p_{ij} 为 i 点与 j 点之间的道路安全系数; M 为足够大的正数。决策变量: z_i , 在救援仓库 i 开放时为 1, 否则为 0; x_{ijk} , 车辆 k 在访问 i 点之后访问 j 点为 1, 否则为 0; y_{ij} , 需求点 j 由救援仓库 i 服务为 1, 否则为 0。

1.3 模型建立

目标函数 1 表示最小化总成本,该成本由 3 个部分组成:救援仓库开放所带来的固定成本、启用车辆的固定成本和应急物资的运输成本,见式(1)。目标函数 2 表示最小化车辆延时送达的时间惩罚成本,见式(2)。目标函数 3 表示最大化车辆行驶的道路安全性,见式(3)。这 3 个目标函数旨在从成本、时间和安全性 3 个维度来优化物流配送,实现在满足服务要求的同时,达到成本效益最大化和安全风险最小化的目标。约束条件 1 表示只有开放的救援仓库才能运输物资,见式(4)。约束条件 2 表示每个需求点有且仅有 1 辆车对其服务,见式(5)。约束条件 3 表示每个需求点有且仅有 1 个救援仓库对其进行服务,见式(6)。约束条件 4 表示救援仓库之间未连通,见式

(7)。约束条件 5 表示救援仓库容量限制, 即任意一个救援仓库发出去的物资不超过仓库的存储容量, 见式 (8)。约束条件 6 表示车辆容量约束, 即每条配送路线的每辆车的负载低于车容量, 见式 (9)。约束条件 7 表示节点的车辆进出流量守恒约束, 见式 (10)。约束条件 8 表示消除子回路, 见式 (11)。约束条件 9 表示车辆在访问需求点时违反的时间窗长度, 见式 (12)。约束条件 10 表示时间窗口约束, 见式 (13)。约束条件 11~14 定义了决策变量及参数, 见式 (14)~(17)。

$$\min F_1 = \sum_{i \in I} f_i z_i + \sum_{j \in J} \sum_{i \in I} \sum_{k \in K} f_k x_{ijk} + \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} t_{ij} x_{ijk} \quad (1)$$

$$\min F_2 = \sum_{j \in J} \sum_{k \in K} \beta G_{jk} \quad (2)$$

$$\max F_3 = \sum_{k \in K} \left(\prod_{i, j \in V, x_{ijk}=1} p_{ij} \right) \quad (3)$$

$$y_{ij} \leq z_i, \forall i \in I, j \in J \quad (4)$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijk} = 1, \forall j \in J \quad (5)$$

$$\sum_{i \in I} y_{ij} = 1, \forall j \in J \quad (6)$$

$$x_{ijk} = 0, \forall i, j \in I, k \in K \quad (7)$$

$$\sum_{j \in J} d_j y_{ij} \leq Q_i, \forall i \in I \quad (8)$$

$$\sum_{j \in J} d_j x_{ijk} \leq m_k, \forall i \in I, k \in K \quad (9)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \forall i \in V, k \in K \quad (10)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} = |S| - 1, \forall k \in K, S \subseteq J \quad (11)$$

$$G_{jk} = \max \{0, t_{jk} - o_j\}, \forall k \in K, j \in V \quad (12)$$

$$t_{ik} + s_i + t_{ij} \leq t_{jk} + M(1 - x_{ijk}), \quad \forall k \in K, i \in I, j \in J \quad (13)$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in V, k \in K \quad (14)$$

$$y_{ij} \in \{0, 1\}, \forall i, j \in V, k \in K \quad (15)$$

$$z_i \in \{0, 1\}, \forall i, j \in V, k \in K \quad (16)$$

$$t_{jk}, G_{jk} \geq 0, \forall j \in J, k \in K \quad (17)$$

2 求解多目标 LRP 问题的改进樽海鞘优化算法

樽海鞘算法 (Salp Swarm Algorithm, SSA) 是一种受樽海鞘群体运动和觅食行为启发的基于群智能的优化算法。在樽海鞘群中, 前一半个体为领导者, 其余的为追随者, 领导者引导种群, 追随者互相跟随, 所有个体形成一条“链”, 通过樽海鞘个体的位置更新移动不断靠近食物源, 最后快速准确地找到最优解^[15]。此算法主要适用于连续优化问题, 算法中樽海鞘的位置更新方法仅适用于连续空间的搜索。考

虑到文中研究的问题属于离散优化问题, 根据选址一路径问题的特点, 并保有算法的特征, 设计一种改进的樽海鞘算法。

2.1 解的表示

使用两段式的实数编码表示每个个体 (可行解), 所有需求点的数量为 n , 染色体长度为 $2n$, 解的构成由 2 个部分构成, 分别为 x 和 y , 用不同的颜色表示, 如图 1 所示。假设救援仓库、需求点的数量分别为 4、8, 解的编码长度为 16, 其中 x 表示需求点与救援仓库的归属关系; y 表示需求点的初始配送顺序。如图 1 所示, 4 个候选救援仓库中 2 号救援仓库未开放, 其中需求点 4 和 7 由救援仓库 1 提供服务, 车辆配送物资的顺序为 7→4; 需求点 5、1、6 由救援仓库 3 提供服务, 车辆配送物资的顺序为 5→1→6。

同理, 需求点 2、3、8 由救援仓库 4 提供服务, 车辆配送物资的顺序为 3→8→2。这种编码方式简单直观, 可体现出开放的仓库、需求点的归属及救援仓库所服务的需求点的配送顺序。

x部分								y部分							
3	4	4	1	3	3	1	4	5	3	8	7	1	4	2	6

图 1 解的表示

Fig.1 Representation of solution

2.2 初始种群的生成

初始解根据概率选择贪心算法或随机生成初始种群。其中, 贪心算法只在生成需求点和仓库归属关系时发挥作用, 首先计算出每个需求点到所有候选救援仓库的距离, 按照升序排列, 依照概率选择离自己最近或其他救援仓库。随机生成初始解时, 首先随机为各个需求点分配救援仓库, 确立需求点与救援仓库的归属关系, 然后随机生成路径规划, 确立每个救援仓库为需求点配送物资的顺序, 计算见式 (18)~(19)。

$$a_{i,j}^0 = \text{round}[a^L + \text{rand}(a^U - a^L)] \quad (18)$$

$$i = 1, 2, \dots, N_p, j = 1, 2, \dots, n$$

$$a_{(i,j+n)}^0 = \text{randperm}(j) \quad i = 1, 2, \dots, N_p, \quad (19)$$

$$j + n = n + 1, n + 2, \dots, 2n$$

式中: a^U 表示 a 的上界; a^L 表示 a 的下界, 最少有一个救援仓库为需求点提供配送服务, 故 $a^L=1$; 最多所有救援仓库都提供配送服务, 故 a^U 为所有候选救援仓库的数量。初始解的长度为 $2n$, 式 (18) 表示前 n 列需求点与救援仓库的归属关系, 式 (19) 表示 $n+1$ 列到 $2n$ 列需求点被配送的顺序。

2.3 快速非支配排序

在多目标问题中, 由于多个目标之间往往存在冲突, 很难找到一个解使所有目标函数最优, 因此引入 NSGA-II 中的非支配排序和拥挤度的概念^[16], 按照每个个体的非支配和支配关系将他们排序分级, 并且根据拥挤度选出同级别中的优解。拥挤度计算需要根据每个目标函数值按升序总体进行排序。然后, 将等级中的边界解(具有最小和最大函数值的解)的拥挤度设为无穷大。中间解 i 的拥挤度根据式(20)计算。

$$C_i = \sum_{c=1}^o \frac{F_c(i+1) - F_c(i-1)}{\max F_c - \min F_c} \quad (20)$$

式中: o 为目标函数的数量; C_i 为第 i 个个体的拥挤度; $i+1$ 、 $i-1$ 为个体 i 沿着帕累托边界的 2 个相邻个体; $F_c(i)$ 为 i 个体的第 c 个目标函数值。在所有个体中, 排序值更靠前和拥挤度更大的个体更优。

2.4 领导者-跟随者自适应调整

在基本 SSA 算法中, 领导者和追随者的数量始终各占种群总数的一半, 这样导致在算法迭代早期, 全局搜索的领导者比例过低, 跟随者的比例过高, 可能导致算法无法充分进行全局搜索而陷入局部最优解。在算法迭代后期, 跟随者的数量过少, 导致局部搜索不充分, 会影响搜索的精度。

为了解决这个问题, 这里引入了一种领导者-跟随者自适应调整策略。其中, 樽海鞘领导者的数量会随着迭代次数的增加而自适应地减少, 而跟随者的数量会自适应地增加。这样, 算法在迭代早期可以保持强大的全局搜索能力, 同时算法在迭代后期, 其局部搜索能力逐渐增强, 从而提高了整体的优化精度。

计算改进后的领导者-跟随者, 每代中领导者数量等于 rN_p , 跟随者数量等于 $(1-r)N_p$, N_p 为种群数量, 自适应权重因子 r 的计算见式(21)。

$$r = r - (r_{\text{init}} - r_{\text{final}}) \left(\frac{t}{T_{\text{max}}} \right)^2 \quad (21)$$

式中: t 为当前迭代次数; T_{max} 为最大迭代次数; r_{init} 为控制参数 r 的初始值; r_{final} 为控制参数 r 的终值。自适应权重因子 r 在算法迭代中随着迭代次数而变化, 由 r_{init} 、 r_{final} 确定。经过多次对比实验, 最终将 r_{init} 取为 0.7, 将 r_{final} 取为 0.3, 用于动态控制算法初始时和结束时樽海鞘领导者和追随者的数量, 更好地平衡算法的全局搜索和局部开发能力。

2.5 领导者位置更新

这里设计的改进领导者位置部分主要包含交叉操作 Cross1、Cross2。交叉操作产生于领导者个体和食物源个体中, 有利于平衡算法的全局搜索和局部搜索。食物源个体的选择: 在排序值为 1 且拥挤度最

大的个体中随机选择。Cross1 为领导者个体与食物源个体 x 部分的交叉, Cross2 为领导者个体与食物源个体 y 部分的交叉操作。具体操作如图 2~3 所示。

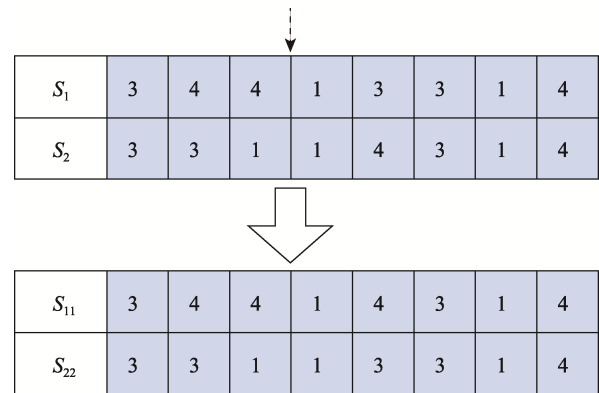


图 2 Cross1 操作演示

Fig.2 Demonstration of Cross1 operation

Cross1, 如图 2 所示, S_1 和 S_2 分别为领导者和食物源的 x 部分, 表示需求点和救援仓库的分配关系, 随机生成的交叉点为 3, 将 S_1 和 S_2 点位 3 后的部分互换, 交换彼此的需求点和救援仓库分配策略, 得到子代 S_{11} 和 S_{22} , 新产生的子代个体不仅可以学习食物源个体的分配策略, 也可能会产生更高质量的解。

Cross2, 将领导者个体与食物源个体的 y 部分进行交叉, y 部分表示配送顺序策略。如图 3 所示, P_1 为领导者的 y 部分, P_2 为食物源的 y 部分, 作为 2 个父代, 随机生成的交叉点为 3、6, 选择两点位之间的基因, 在另一个父代上找到这些基因的位置, 保持未选中的基因不变, 按照选中的基因在另一父代上的出现顺序, 交换 2 个父代个体中基因的位置, 生成 P_{11} 、 P_{22} 。通过这种交叉方式可以学习到食物源个体的配送顺序策略, 在增强种群多样性的同时也可能产生更优解。

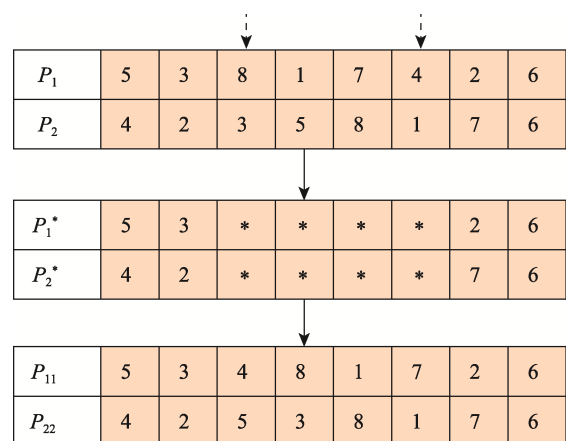


图 3 Cross2 操作演示

Fig.3 Demonstration of Cross2 operation

将领导者个体和食物源个体的 x 部分与 y 部分进行交叉操作, 并随机从子代 $[S_{11} P_{11}]$ 或 $[S_{22} P_{22}]$ 中选择一个作为新的领导者个体的位置。

2.6 追随者位置更新

对于追随者位置的更新, 为了更好地对解进行局部开发, 设计了交叉操作 Cross3 和邻域搜索, 包括单点变异、两点交换、单点插入、逆转及仓库弃用。

1) 交叉算子。在所有父代中随机选择 1 个樽海鞘个体, 与当前追随者个体进行交叉。个体的 x 部分和 y 部分分别是 2 种不同的交叉方式, 其中 x 部分交叉与领导者位置更新时 x 部分的交叉方式 Cross1 相同, y 部分的交叉方式 Cross3 具体操作如图 4 所示。

Cross3, 追随者个体和父代樽海鞘个体的 y 部分进行交叉, 交叉后的个体可以学到父代个体的配送策略。交叉操作步骤: P_1 为随机樽海鞘个体的 y 部分, P_2 为当前追随者的 y 部分, 作为 2 个父代, 在 P_1 上随机生成的交叉点 o_1 , 选择位置 o_1 上的元素 e_1 , 其次找到 P_2 中位置 o_1 的元素 e_2 , 再回到 P_1 中找到元素 e_2 所在的位置 o_2 , 然后找到 P_2 中 o_2 位置上的元素 e_3 。重复之前操作, 直到形成一个环, 环中的所有元素的位置即是最后选中的位置。如图 4 所示, 位置 $2 \rightarrow 7 \rightarrow 4 \rightarrow 2$ 形成一个循环, 将 P_1 中相应位置的元素替换到 P_2 中, 形成新的个体 P_{11} 。

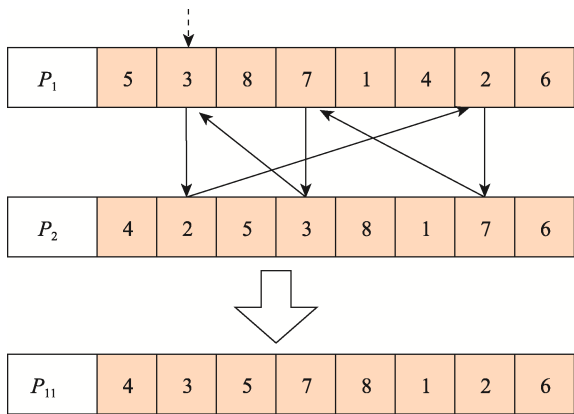


图 4 Cross3 操作演示
Fig.4 Demonstration of Cross3 operation

2) 邻域搜索。为了进一步提高樽海鞘算法在离散优化问题中的寻优能力, 增强种群的多样性, 结合 LRP 问题的特点选择 5 种邻域搜索机制对解进行局部开发, 以相同的概率对解进行搜索。邻域搜索针对 x 部分 (需求点与仓库归属关系) 和 y 部分 (需求点的配送顺序) 进行搜索。针对 x 部分有 5 种邻域搜索操作, 即单点变异、两点交换、插入、逆转、仓库弃用。针对 y 部分有 3 种邻域搜索操作, 即两点交换、插入、逆转。

单点变异: 随机选取一个位置 i , 将其对应的所属仓库进行变异操作。如图 5 所示, 随机选中变异的位置是 3, 原本为 4 号仓库服务, 标记为红色, 随

机生成新的 1 号仓库为需求点配送物资。

两点交换: 随机选取位置 i 和 j , 将其对应的所属仓库的序号互换。如图 6 所示, 随机选中的 2 个位置为 3、6, 标记为红色, 将 3 号需求点和 6 号需求点所归属仓库互换, 形成新的解。原来 3 号需求点由 4 号仓库服务, 6 号需求点由 3 号仓库服务, 交换之后, 3 号需求点由 3 号仓库服务, 6 号需求点由 4 号仓库服务。

单点插入: 随机选取 x 部分的 2 个位置 i 和 j , 如图 7 所示, 用红色标记。将位置 i 的编码插到 j 的编码之前, 得到新的解。如图 7 所示, 随机选中的位置为 3 和 7, 此时 3 到 6 号需求点所属仓库编码都向前移动, 都发生了相应的改变。

逆转: 随机选取 x 部分的 2 个位置 i 和 j , 然后将 i 和 j 之间 (包括 i 和 j) 的所有序号逆向排序。如图 8 所示, 选取的 2 个随机位置为 3 和 7, 将其中的序号 $[4 \ 1 \ 3 \ 3 \ 1]$ 进行逆向排序, 得到新的解。

仓库弃用: 在开放的仓库索引里随机选择一个弃用仓库索引, 并将其换成任意合理的仓库索引。具体操作如图 9 所示, 选中的弃用仓库索引为 4, 关闭 4 号仓库, 开放 2 号仓库, 得到新的解。

同理, 对于 y 部分有 3 种搜索操作, 操作步骤与 x 相似, 针对 y 部分的操作改变了需求点的顺序, 针对不同部分有不同的领域操作, 在增强种群多样性的同时, 也可能产生更优解。

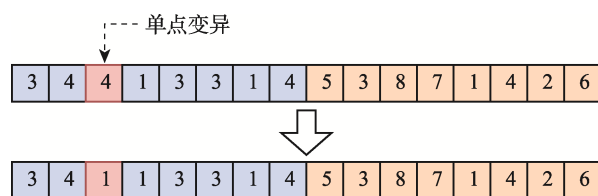


图 5 单点变异
Fig.5 Single point mutation

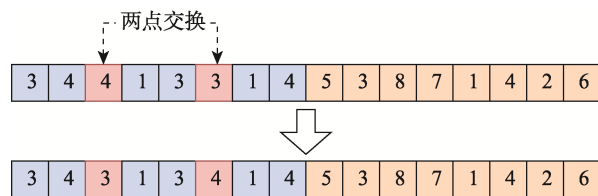


图 6 两点交换
Fig.6 Two point change

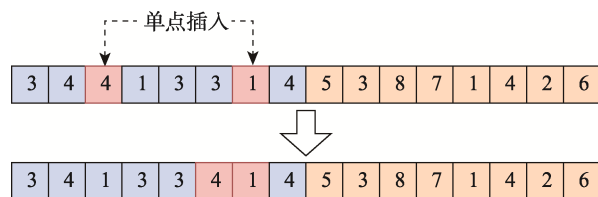


图 7 单点插入
Fig.7 Single insertion

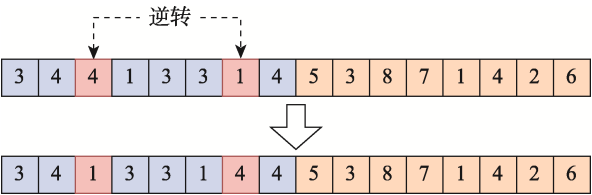


图 8 逆转
Fig.8 Reverse

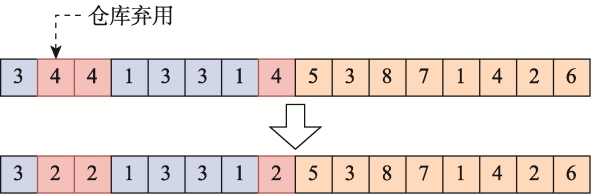


图 9 仓库弃用
Fig.9 Depot abandonment

2.7 精英保留机制

在 ISSA 算法中引入 NSGA-II的精英保留策略。将位置更新前的父代种群与位置更新后的子代种群合并为一个新种群, 然后采用 2.3 节的方法, 重新计算新种群的非支配排序等级和拥挤度, 选取前 N_p (种群数量) 个最优的个体为下一代樽海鞘种群。精英保留机制有利于保留种群中的优良个体, 提高种群的整体进化水平。

2.8 算法步骤

这里提出的改进多目标樽海鞘算法步骤如下。

- 1) 初始化算法参数, 种群规模 N_p 、算法最大迭代次数 $T_{\max}$, 樽海鞘个体长度 L 。
- 2) 种群初始化, 采用贪心算法与随机生成的方式生成 N_p 个樽海鞘个体作为初始可行解。
- 3) 对种群进行非支配排序和拥挤度的计算, 评估每个个体的目标函数值, 并给每个个体排序。根据排序确定领导者和追随者。分配领导者和追随者种群, 根据式 (21) 更新 r , 前 rN_p 个的樽海鞘个体为

- 领导者, 剩余个体为追随者。
- 4) 确定食物源个体, 在排序值最低且拥挤度最大的个体中随机选择一个作为食物源个体。
- 5) 根据 2.5 节的策略更新领导者位置, 根据 2.6 节的策略更新追随者位置。
- 6) 精英保留机制将更新后的种群与父代种群合为一个种群, 并对其进行非支配排序, 根据排序值选择前 N_p 个为新的种群。
- 7) 判断迭代次数是否达到最大迭代次数 $T_{\max}$, 如果达到, 算法终止, 如果未达到则返回步骤 3), 循环操作直到达到终止条件。

3 算例分析

采用 Matlab 2016b, 算法运行环境采用 AMD Ryzen 5 5600H 处理器、3.30 GHz、内存 16.0 GB、64 位 Windows 10 操作系统。为了测试所提算法的性能, 使用改进后的标准算例库中的部分算例及根据实际情景生成的一组算例进行求解, 并对结果进行分析。

3.1 实验设计

借鉴 Prins 等^[17]提出的标准 LRP 算例数据, 所有参数均无量纲。由于算例库中无时间窗及道路安全性的相关数据, 不完全适合于文中的问题。为了模拟真实情景, 对算例中不同规模的部分数据进行适当改进后, 生成文中的数据集。在生成的数据集中, 车容量在 {70, 150} 之间, 仓库容量在 {300, 1 000} 之间。其中, 20-5-1 的软时间窗最晚到达时间在 [50, 300] 之间生成, 20-5-1b 在 [100, 400] 之间生成, 其余算例在 [250, 500] 之间生成。救援仓库需要在突发灾害后短时间内开放, 需要耗费很多的人力、物力, 因此将仓库的开放成本设置较高, 设置为 8 000, 将救援车辆的启用成本设置为 1 000, 单位运输成本设置为 100。同时将违反时间窗口的惩罚值设置为相对较高的 1 000, 每个点之间的道路安全性参数在 (0, 1) 之间随机生成。测试算例的相关信息如表 1 所示。

表 1 测试算例相关信息
Tab.1 Relative data of test examples

名称	需求点数量	需求	仓库数量	仓库容量	车辆启用成本	车辆容量
20-5-1	20	$U(10, 20)$	5	300	1 000	70
20-5-1b	20	$U(10, 20)$	5	300	1 000	150
50-5-1	50	$U(10, 20)$	5	300	1 000	70
50-5-1b	50	$U(10, 20)$	5	300	1 000	150
50-5-2	50	$U(10, 20)$	5	300	1 000	70
50-5-2b	50	$U(10, 20)$	5	300	1 000	150
100-5-1	100	$U(10, 20)$	5	1 000	1 000	150
100-5-1b	100	$U(10, 20)$	5	1 000	1 000	150
100-5-2b	100	$U(10, 20)$	5	1 000	1 000	150
100-10-1	100	$U(10, 20)$	10	300	1 000	150

3.2 参数设置

设置改进多目标樽海鞘算法的相关参数,每个个体的长度 $L=2n$, n 表示需求点的数量。为了准确评估各个参数对 ISSA 算法性能的影响,这里采用参数调优策略。通过大量实验,根据算法迭代效果设置 $T_{\max}=300$ 。以下主要对种群规模 N_p 进行分析。为了测试种群规模 N_p 对算法性能的影响,利用算例 20-5-1、50-5-1b 进行实验对比。在保持其他参数恒定的情况下,设置种群为 $0.5n$ 、 n 、 $1.5n$ 、 $2n$ 、 $2.5n$ 、 $3n$ (n 为算例中的需求点数量),将实验运行 20 次,取帕累托前沿所有非支配解的平均值。

由表 2 可知,当种群规模较小时,算法很容易陷入局部最优,从而得到较差的结果。随着种群数量的增大,算法运行需要的时间也相应增加。当规模为 $2n$ 时,综合考虑解的质量和运行时间,设置种群数

量 $N_p=2n$ (n 为需求点的数量),后续研究也一直沿用。

3.3 算例结果分析

对于每个算例,采用改进樽海鞘算法 (ISSA) 和原始樽海鞘优化 (SSA) 算法都运行 20 次,计算所有非支配解的经济成本 (F_1)、时间成本 (F_2)、道路安全性 (F_3) 的平均值,并找出所有非支配解中的最优解,结果见表 3。此外,以 50-5-1b 为例,给出 ISSA 和 SSA 求解的三维帕累托前沿图,结果如图 10 所示。

由表 3 可知,ISSA 算法在各算例中求解的 F_1 、 F_2 、 F_3 的最优值和平均值均优于 SSA 算法。从图 10 可以很清晰地看出,ISSA 求出的帕累托最优解数量更多,并且帕累托面位于 SSA 求解出的帕累托面上方,说明改进后樽海鞘算法的性能更加优越,能得到更满意的解。

表 2 种群数量 N_p 对解的影响
Tab.2 Effect of parameter N_p on solution

规模	F_1		F_2		F_3		运行时间/s	
	20-5-1	50-5-1b	20-5-1	50-5-1b	20-5-1	50-5-1b	20-5-1	50-5-1b
$0.5n$	76 301.89	202 241.53	233 876.80	598 586.24	3.48	0.96	9.43	13.78
n	75 224.09	198 141.40	272 543.58	493 003.30	3.55	1.02	9.61	23.25
$1.5n$	74 750.72	191 854.20	196 962.61	483 603.90	3.47	1.08	14.32	38.52
$2n$	72 883.79	186 284.20	172 775.50	445 503.11	3.34	1.13	20.90	90.11
$2.5n$	77 489.19	207 946.15	213 220.42	455 386.43	3.14	1.13	17.92	61.16
$3n$	75 000.09	206 310.50	196 848.30	463 848.60	3.52	1.05	27.97	121.72

表 3 算法结果对比
Tab.3 Comparison of algorithm results

算例	SSA						ISSA					
	最优 $F_1/10^4$	最优 $F_2/10^4$	最优 F_3	平均 $F_1/10^4$	平均 $F_2/10^5$	平均 F_3	最优 $F_1/10^4$	最优 $ F_2/10^3$	最优 F_3	平均 $F_1/10^4$	平均 $F_2/10^5$	平均 F_3
20-5-1	6.54	2.87	3.81	7.74	2.07	3.44	5.67	2.39	3.74	7.29	1.82	3.34
20-5-1b	12.6	12.6	2.86	14.2	6.73	2.02	8.87	11	3.14	11.8	4.78	2.52
50-5-1	30.5	4.42	2.00	32.9	4.11	1.82	20.3	0	2.07	25.3	1.54	1.99
50-5-1b	25.0	98.5	0.82	25.6	13.2	0.57	15.8	0	1.27	19.2	6.27	1.12
50-5-2	27.4	4.20	2.00	29.1	2.61	1.02	20.7	0	2.03	25.5	1.92	1.74
50-5-2b	26.8	38.8	2.00	58.4	4.08	1.22	13.2	0	2.02	15.1	1.76	1.29
100-5-1	54.8	139	1.77	59	26.6	0.97	32.9	1.02	2.00	42.8	6.74	1.22
100-5-1b	60.1	41.9	0.32	65.2	11	0.20	40.8	0	1.00	49.5	2.67	0.89
100-5-2b	62.1	43.9	0.29	67.1	11.9	0.15	41.1	0	1.00	48.5	2.31	0.96
100-10-1	53.5	948	3.65	57	137	2.58	25	17.3	4.06	31.1	24.8	3.16

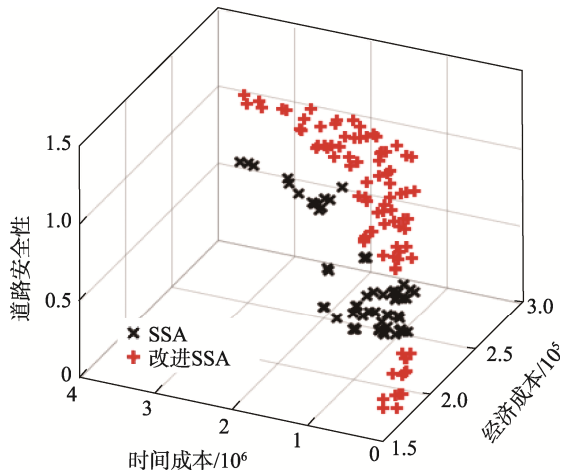


图 10 ISSA 和 SSA 求解的 Pareto
Optimal Surface (50-5-1b)
Fig.10 Pareto Optimal Surface of
ISSA and SSA (50-5-1b)

3.4 仿真实验分析

为了进一步验证文中模型和 ISSA 算法的有效性, 选取上海市进行模拟实验, 从上海市的地图中选取 25 个医院作为需求点, 以火车站和机场作为候选仓库点。在选取这些点时, 综合考虑了需求点的地理位置和紧急需求等因素。候选仓库点和需求点的具体信息见表 4、5。需求点与候选仓库点之间的距离根据经纬度坐标计算得到。两点之间的距离根据式 (22) 计算。

$$L = R \arccos[(\sin y_1 \sin y_2 + \cos y_1 \cos y_2 \cos(x_1 - x_2))] \quad (22)$$

式中: R 为地球半径, $R=6\,371\text{ km}$; (x_1, y_1) 、 (x_2, y_2) 为两点的经纬度坐标。

表 4 候选仓库坐标
Tab.4 Candidate depot coordinates

编号	经度/(°)	纬度/(°)	容量/kg
o_1	121.462 056	31.255 923	2 500
o_2	121.435 865	31.159 439	2 500
o_3	121.327 012	31.200 458	2 500
o_4	121.815 09	31.157 478	2 500

为了验证算法及算法各个步骤的有效性, 将未引入领导者-追随者自适应权重因子、未加入邻域搜索及未采用精英保留机制的算法记为 ISSA-I、ISSA-II、ISSA-III, 将其运行 10 次, 将它产生的最佳目标值 (best) 和最佳目标值的平均值 (mean) 与文中提出的 ISSA 算法结果进行对比, 见表 6。可以看出, 算法求得的经济成本、时间成本及道路安全性都存在明显差距, 这是由路径规划不同所致。从各个算法求得的平均值来看, 采用 ISSA 算法求得的经济成本、时间

成本及道路安全性都优于其他算法, 说明在进行一系列改进后, 在一定程度上提高了算法的寻优能力。

在多目标优化领域, 不同目标函数之间的潜在冲突和权衡常常使得一种解无法同时满足所有目标最优。这种现象源于各个目标函数的不同性质和优化方向。可见, 寻求多目标问题的解通常不是寻找所有目标都最优的解, 而是找到一种权衡各目标后取得的理想解。这里采用 ISSA 对此问题进行求解, 求得的具有代表性的解如表 7~8 所示。其中, 表 7 中 Best1、Best2、Best3 分别为 F_1 (经济成本)、 F_2 (时间惩罚成本)、 F_3 (道路安全性) 个体最优情况下求得的解, 理想解指帕累托最前沿且拥挤度最大的个体, 最劣解指等级最低的非支配集合中拥挤度最大的个体。在表 8 中, 编号 1~25 表示需求点, o_1 、 o_2 、 o_3 、 o_4 表示候选仓库。由表 7 可知, 理想点求得的目标函数值对于最劣点函数值及函数平均值, 更具有优越性, 表明所提出的算法能够有效优化问题。这也证明了文中提出的模型和算法的有效性及其现实意义。

表 5 需求点信息
Tab.5 Information of demand point

编号	经度/(°)	纬度/(°)	需求量/kg	最晚到达时间/h
1	121.495 9	31.259 08	380	1.5
2	121.253 4	31.355 57	200	1
3	121.524 7	31.278 97	400	2
4	121.381 9	31.113 9	300	1.5
5	121.709 9	31.195 55	380	2.5
6	121.629	31.122 02	220	1
7	121.743 4	31.064 28	200	1.5
8	121.414	31.011 79	400	1.5
9	121.227 3	31.013 52	300	2
10	121.513	30.928 16	220	1.5
11	121.437 5	31.272 97	250	1.5
12	121.600 4	31.192 32	240	1
13	121.490 6	31.411 8	250	1.5
14	121.46	31.197 37	200	1.5
15	121.144	31.161 29	250	1
16	121.337 9	30.762 86	240	2.5
17	121.534 9	31.314 61	230	1.5
18	121.383 9	31.214 77	200	2
19	121.428 9	31.185 29	180	1.5
20	121.518 7	31.243 46	192	1.5
21	121.431 7	31.203 93	229	1
22	121.473 6	31.238 91	223	2.5
23	121.648 2	30.916 52	228	1
24	121.460 4	31.203 96	284	2
25	121.414	31.011 79	246	1.5

表 6 算法改进对比
Tab.6 Algorithm improvement comparison

变量	ISSA-I		ISSA-II		ISSA-III		ISSA	
	best	mean	best	mean	best	mean	best	mean
F_1	42 559.60	42 956.183	43 157.80	49 701.53	45 169.95	48 506.89	42 553.31	42 766.39
F_2	1 721.61	2 456.70	2 154.86	2 810.55	8 263.29	14 372.50	1 316.67	1 929.68
F_3	1.26	1.19	1.11	0.82	1.19	0.88	1.28	1.19
运行时间/s	23.37	23.8	20.05	21.92	23.04	23.85	23.05	23.76

表 7 具有代表性的有效解 ($m=4, n=25$)
Tab.7 Representative valid solutions ($m=4, n=25$)

方案	X_j	目标函数		
		F_1	F_2	F_3
Best1	$X_1=1, X_2=1, X_3=1$	42 604.69	9 938.14	1.01
Best2	$X_1=1, X_2=1, X_3=1, X_4=1$	55 432.91	2 510.91	0.10
Best3	$X_1=1, X_2=1, X_3=1$	46 825.53	17 167.25	1.13
理想解	$X_1=1, X_2=1, X_3=1$	42 604.69	9 938.14	1.01
最劣解	$X_1=1, X_2=1, X_3=1, X_4=1$	54 930.58	22 429.02	0.57
平均值		48 479.68	12 396.69	0.76

表 8 理想解的路径规划
Tab.8 Route planning for the ideal solution

车辆编号	路线	车辆编号	路线
车辆 1	$o_1 \rightarrow 22 \rightarrow 12 \rightarrow 11 \rightarrow o_1$	车辆 5	$o_2 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow o_2$
车辆 2	$o_1 \rightarrow 1 \rightarrow 14 \rightarrow 20 \rightarrow o_1$	车辆 6	$o_2 \rightarrow 4 \rightarrow 23 \rightarrow 10 \rightarrow 25 \rightarrow o_2$
车辆 3	$o_1 \rightarrow 3 \rightarrow 17 \rightarrow 13 \rightarrow o_2$	车辆 7	$o_3 \rightarrow 2 \rightarrow 15 \rightarrow 9 \rightarrow o_3$
车辆 4	$o_2 \rightarrow 19 \rightarrow 21 \rightarrow 24 \rightarrow o_2$	车辆 8	$o_3 \rightarrow 8 \rightarrow 16 \rightarrow 18 \rightarrow o_3$

4 结论

针对突发灾害后的应急物资选址-路径问题进行了研究,为了达到配送物资的经济性、时效性及运输过程中的安全性,建立了一个在灾后及时响应的环境下具有时间窗口的多目标选址-路径模型,以经济成本最小化、时间惩罚成本最小化及道路安全性最大化为目标,为决策救援仓库选址及救援车辆的配送路线提供方案。为了有效解决多目标优化问题,根据 LRP 模型的特点及基本樽海鞘算法(SSA)的搜索机制提出了一种改进的樽海鞘算法(ISSA),并与未改进的基本樽海鞘算法进行对比。基于多个算例的分析比较,证实了文中所提模型和算法的科学性和有效性,在求解质量上具有一定的优越性。最后将 ISSA 用于实际背景下的算例中,验证了算法的有效性。同时,在权衡多个目标的状况下,提供了一种方案可供决策者找到较满意的解。

此研究也存在一些不足,如只考虑了确定性方案,而现实中信息的滞后性会导致很多信息不确定。后续会进一步考虑不确定情况对结果的影响,更加全面地考虑灾后物资调度的规划与决策。

参考文献:

- [1] ZHONG S P, CHENG R, JIANG Y, et al. Risk-Averse Optimization of Disaster Relief Facility Location and Vehicle Routing under Stochastic Demand[J]. Transportation Research Part E: Logistics and Transportation Review, 2020, 141: 102015.
- [2] VAHDANI B, VEYSMORADI D, NOORI F, et al. Two-Stage Multi-Objective Location-Routing-Inventory Model for Humanitarian Logistics Network Design under Uncertainty[J]. International Journal of Disaster Risk Reduction, 2018, 27: 290-306.

- [3] ZHANG B, LI H, LI S G, et al. Sustainable Multi-Depot Emergency Facilities Location-Routing Problem with Uncertain Information[J]. *Applied Mathematics and Computation*, 2018, 333: 506-520.
- [4] WEI X W, QIU H X, WANG D J, et al. An Integrated Location-Routing Problem with Post-Disaster Relief Distribution[J]. *Computers & Industrial Engineering*, 2020, 147: 106632.
- [5] SHEN L, TAO F M, SHI Y H, et al. Optimization of Location-Routing Problem in Emergency Logistics Considering Carbon Emissions[J]. *International Journal of Environmental Research and Public Health*, 2019, 16(16): 2982.
- [6] 刘长石, 彭怡, 寇纲. 震后应急物资配送的模糊定位-路径问题研究[J]. *中国管理科学*, 2016, 24(5): 111-118.
- LIU C S, PENG Y, KOU G. Research on Fuzzy Location-Routing Problem in Post-Earthquake Delivery of Relief Materials[J]. *Chinese Journal of Management Science*, 2016, 24(5): 111-118.
- [7] PRODHON C, PRINS C. A Survey of Recent Research on Location-Routing Problems[J]. *European Journal of Operational Research*, 2014, 238(1): 1-17.
- [8] LIU B J, ZHU L, REN J L. Intelligent Optimization Algorithm Grid Computing-Based Applications[J]. *Journal of Intelligent & Fuzzy Systems*, 2020, 39(4): 5201-5211.
- [9] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems[J]. *Advances in Engineering Software*, 2017, 114: 163-191.
- [10] 张达敏, 陈忠云, 辛梓芸, 等. 基于疯狂自适应的樽海鞘群算法[J]. *控制与决策*, 2020, 35(9): 2112-2120.
- ZHANG D M, CHEN Z Y, XIN Z Y, et al. Salp Swarm Algorithm Based on Craziness and Adaptive[J]. *Control and Decision*, 2020, 35(9): 2112-2120.
- [11] ALJARAH I, HABIB M, FARIS H, et al. A Dynamic Locality Multi-Objective Salp Swarm Algorithm for Feature Selection[J]. *Computers & Industrial Engineering*, 2020, 147: 106628.
- [12] 李志杰, 王力, 张习恒. 改进樽海鞘群优化 K-means 算法的图像分割[J]. *包装工程*, 2022, 43(9): 207-216.
- LI Z J, WANG L, ZHANG X H. Improved Salp Swarm Optimization K-Means Algorithm for Image Segmentation[J]. *Packaging Engineering*, 2022, 43(9): 207-216.
- [13] SALGOTRA R, SINGH U, SINGH S, et al. Self-Adaptive Salp Swarm Algorithm for Engineering Optimization Problems[J]. *Applied Mathematical Modelling*, 2021, 89: 188-207.
- [14] NAUTIYAL B, PRAKASH R, VIMAL V, et al. Improved Salp Swarm Algorithm with Mutation Schemes for Solving Global Optimization and Engineering Problems[J]. *Engineering with Computers*, 2022, 38(5): 3927-3949.
- [15] ZHANG H L, CAI Z N, YE X J, et al. A Multi-Strategy Enhanced Salp Swarm Algorithm for Global Optimization[J]. *Engineering with Computers*, 2022, 38(2): 1177-1203.
- [16] DEB K, PRATAP A, AGARWAL S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [17] PRINS C, PRODHON C, CALVO R W. Solving the Capacitated Location-Routing Problem by a GRASP Complemented by a Learning Process and a Path Re-linking[J]. *4OR*, 2006, 4(3): 221-238.