

突发灾害下带软时间窗多车路径搜索建模

刘天虎, 许维胜, 吴启迪

(同济大学 电子与信息工程学院, 上海 201804)

摘要: 以带软时间窗的多救援车辆路径搜索问题为建模核心, 首先利用近邻启发算法对混合遗传算法进行初始化, 通过迭代交换产生新的子代染色体; 其次由轮盘赌法选择健康的子代染色体, 引入交叉算子、变异/突变算子对子代染色体进行改良, 从而得到最小总成本消耗下的医疗救援车辆行驶路径, 并有效缩短车辆行驶时间; 最后以非常规突发灾害下的医疗救助为案例对该模型及算法的有效性和可行性进行了说明。

关键词: 非常规突发事件; 医疗援救; 多车路径搜索

中图分类号: C934

文献标识码: A

Modeling of Multi-vehicle Route Searching with Soft Time Windows Under Sudden-onset Disaster

LIU Tianhu, XU Weisheng, WU Qidi

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: This study focuses on the modeling of the multi-vehicle route searching with soft time windows for medical rescue. First of all, the hybrid genetic algorithm (HGA) is initialized by nearest neighbor heuristic algorithm and the new chromosomes are generated through iterated swap procedure. Secondly, the roulette wheel selection is adopted to choose fitter chromosomes to undergo genetic operations and the offspring can be improved through order crossover, heuristic mutation and inversion mutation. And thereby the optimal route for medical rescue is obtained under the minimum total cost as well as the travel time of vehicles is shorthed effectively. Eventually, a medical rescue example under unconventional sudden-onset disaster is provided to demonstrate the validity and feasibility of this model and algorithm.

Key words: unconventional emergency; medical rescue; multi-vehicle route searching

车辆路径问题(vehicle routing problem, VRP)是在满足一定约束条件下使适宜的行车路径达到既定的目标(如费用最小、耗时最短、行程最短等)。而带时间窗的车辆路径问题(vehicle routing problem with time windows, VRPTW)则是在 VRP 问题上加上了访问需求的时间窗口。Desaulniers 等^[1]研究了具有等待成本的多资源站的 VRPTW 问题, 得到了最优解。Qureshi 等^[2]提出了在软时间窗约束下的 VRP 搜索的精确最优解决方案, 从而识别最短的行驶路径, 控制最小的成本消耗。Ghoseiri 等^[3]对多目标的 VRPTW 搜索问题进行了系统研究, 利用目标规划及遗传算法得到了最优路径解。为了求解 VRPTW 问题, 过去的研究中最常用的算法有禁忌搜索算法^[4]、插入启发式算法^[5]、粒子群算法^[6]、人工智能算法^[7]、拉格朗日松弛算法、近似算法等, 本文在以上研究的基础上, 利用混合遗传算法(hybrid genetic algorithm, HGA)^[8]研究当医疗救援站仅有 1 个时, 如何调度多辆医疗救援车辆的最优 VRPTW 问题。

1 问题描述

突发灾害事件的发生往往对医疗救援活动提出急迫的需求, 如何在有限的资源条件下实现最有效的救援问题越来越引起人们的重视, 特别是如发生地震这样的灾害事件, 要求医疗急救资源迅速做出响应, 这就需要各个医疗救援站对救援车辆进行合理调度, 然而一个医疗救援站不可能为所有的灾害点提供救援服务, 而且每个灾害点在不同阶段的医疗救助需求也可能不同。本文需要解决以下 2 个问题: ①当发生突发灾害事件时如何最小化救援的总成本。总成本的控制对救援站资源的使用效率将带

来重要影响. ②如何评价 HGA 算法在解决带软时间窗 VRPTW 搜索问题方面的有效性. 需要深入分析寻优模型, 并通过与精确算法数例仿真比较分析来说明其有效性.

2 模型的建立

实际救援中需要依据灾害点的特征及时调整救援路径, 利用软时间窗作为约束条件比硬时间窗更加具有弹性, 可以得到更加可行的解决方案. 当灾害点不能在软时间窗的时限内得到及时救助时, 引入惩罚成本, 惩罚成本参数可以依据实际情况加以调整, 通过权衡惩罚成本和使用额外车辆的固定成本, 可以对医疗救援车辆进行合理调度. 假定条件: ①假定仅有 1 个医疗救援站为所属灾害点提供救援服务, 救援车辆完成服务后需返回医疗救援站. ②假定医疗救援站的车辆数及所有车辆的运载能力已知. ③假定救援车辆与医疗救援站之间可以实现无障碍沟通, 医疗救援站可监控所有救援车辆的所在位置, 可以评估出救援车辆的行驶时间信息.

这里可以通过图 1 来说明软时间窗下的多车路径搜索问题. 图中的第 1 个括号内的数值表示灾害点的 x, y 坐标方位, 而第 2 个括号内的数值表示灾害点对医疗救援的需求量, 而单车的最大承载量不能超过 13 个单位量; A_0 表示救援站; D_i 表示灾害点, $i=1, \dots, 13$.

从图 1a 中可以看出, 最初的灾害点有 10 个, 有 2 部医疗救援车辆提供救援服务, 车辆 1 的行驶路径为: $A_0 \rightarrow D_5 \rightarrow D_1 \rightarrow D_9 \rightarrow D_8 \rightarrow D_6 \rightarrow A_0$. 而车辆 2 的行驶路径为: $A_0 \rightarrow D_4 \rightarrow D_{10} \rightarrow D_3 \rightarrow D_7 \rightarrow D_2 \rightarrow A_0$.

当 T_1 时刻车辆 1 行驶到灾害点 D_5 , 而车辆 2 行驶到 D_{10} 时, 此时收到医疗救援站的最新信息, 又有 4 个灾害点 $D_{11}, D_{12}, D_{13}, D_{14}$ 需要救助, 从图 1b 可以看出, 在规划的路径范围内, 灾害点 D_{11} 和 D_{13} 可由车辆 1 提供救助, 而灾害点 D_{12} 和 D_{14} 可由车辆 2 提供救助, 在不超过车载容量的前提下, 对 2 部医疗救援车辆的后续行驶路径进行了优化, 车辆 1 后续行驶路径优化为: $D_5 \rightarrow D_{13} \rightarrow D_1 \rightarrow A_0, A_0 \rightarrow D_{11} \rightarrow D_9 \rightarrow D_8 \rightarrow D_6 \rightarrow A_0$, 车辆 2 后续行驶路径优化为: $D_{10} \rightarrow D_3 \rightarrow D_{14} \rightarrow A_0, A_0 \rightarrow D_7 \rightarrow D_{12} \rightarrow D_2 \rightarrow A_0$.

从以上的分析可以看出, 软时间窗下的路径搜索问题实际是一个混合整数线性规划, 于是可以建立相应的数学模型, 模型的目标是为了最小化救援的总成本(包括车辆成本、路径成本和惩罚成本). 而

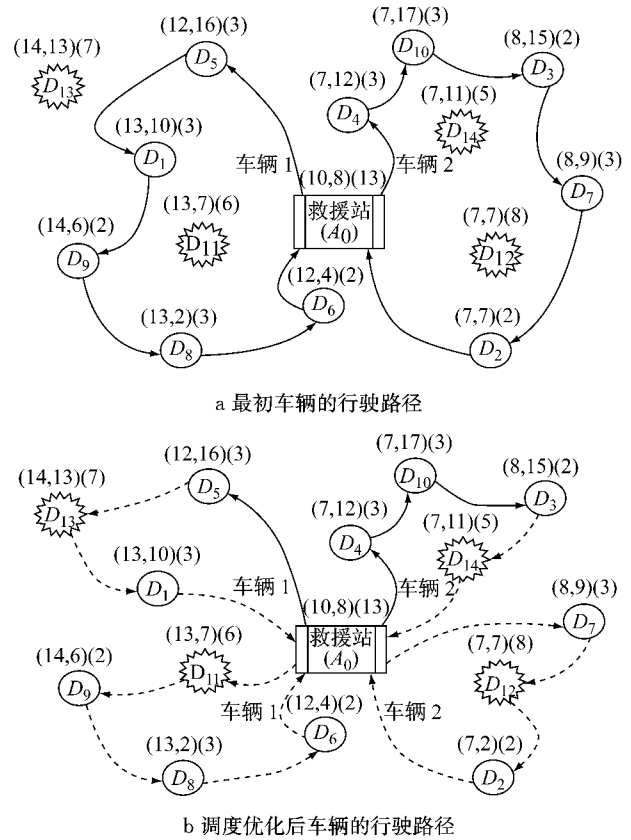


图 1 车辆行驶路径

Fig. 1 Travel route of vehicle

该模型的限制条件由车辆、路径、灾害点时间窗所约束, 于是可以建立模型的最优方程 z 如下:

$$\begin{aligned} \min z = & C_f \sum_{i \in M} \sum_{j \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} \theta_{ij,\tau} + \\ & \sum_{i \in \Psi \setminus G} \sum_{j \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} T_{ij,\tau} \theta_{ij,\tau} + \\ & W_e \sum_{i \in \Psi \setminus (G,M)} E_i + W_1 \sum_{i \in \Psi \setminus (G,M)} L_i \quad (1) \end{aligned}$$

限制条件如下:

$$\sum_{j \in \Psi \setminus (G,M)} \theta_{ij,t_0} \geq \sum_{j \in \Psi \setminus (G,M)} \sum_{\tau=t_0+1}^{t_n} \theta_{ij,\tau} \quad i \in M_0 \quad (2)$$

$$\sum_{j \in \Psi \setminus M} \theta_{ij,t_i,k} = 1 \quad i \in M_1, i \neq j \quad (3)$$

$$\sum_{i \in M_0} \sum_{j \in \Psi \setminus (G,M)} \theta_{ij,t_0} = \sum_{i \in \Psi \setminus (G,M)} \sum_{j \in G_0} \sum_{\tau=t_0+1}^{t_n} \theta_{ij,\tau} \quad (4)$$

$$\sum_{i \in \Psi \setminus (G,M)} \sum_{j \in G_1} \sum_{\tau=t_0}^{t_n} \theta_{ij,\tau} = 1 \quad (5)$$

$$Q(\alpha_{i,k} - 1) - \sum_{j \in H} d_j \alpha_{j,k} \leq \omega_i \quad i \in M, k \in K \quad (6)$$

$$\sum_{k \in K} (\alpha_{i,k} S_k) \geq \omega_i \quad i \in \Psi \quad (7)$$

$$d_j + \omega_i + Q\left(\sum_{\tau=t_0}^{t_n} \theta_{ij,\tau} - 1\right) \leq \omega_j$$

$$i \in \Psi \setminus G, j \in \Psi \setminus M, i \neq j \quad (8)$$

$$\sum_{i \in \Psi \setminus \{G, M\}} \sum_{j \in G} \sum_{\tau=t_0}^{t_n} [\theta_{ij, \tau} (\tau + T_{ij, \tau})] \leq R \quad (9)$$

$$\sum_{i \in \Psi \setminus G} \sum_{\tau=t_0}^{t_n-1} \theta_{ij, \tau} = 1 \quad j \in \Psi \setminus \{G, M\}, i \neq j \quad (10)$$

$$\sum_{k \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} \theta_{kj, \tau} = \sum_{i \in \Psi \setminus G} \sum_{\tau=t_0}^{t_n} \theta_{ij, \tau} \quad j \in \Psi \setminus \{G, M\}, \\ i \neq j, j \neq k \quad (11)$$

$$\sum_{i \in \Psi} \sum_{k \in K} \alpha_{i, k} = 1 \quad (12)$$

$$\sum_{i \in \Psi} \sum_{k \in K} (k \alpha_{i, k}) = P_i \quad (13)$$

$$\sum_{k \in K} (k \alpha_{i, k} - k \alpha_{j, k}) \geq Q \left(\sum_{\tau=t_0}^{t_n} \theta_{ij, \tau} - 1 \right) \\ i \in \Psi \setminus G, j \in \Psi \setminus M, i \neq j \quad (14)$$

$$\sum_{k \in K} (k \alpha_{i, k} - k \alpha_{j, k}) \leq Q \left(1 - \sum_{\tau=t_0}^{t_n} \theta_{ij, \tau} \right) \\ i \in \Psi \setminus G, j \in \Psi \setminus M, i \neq j \quad (15)$$

$$E_i = \max(0, \lambda_i - \sum_{j \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} \tau \theta_{ij, \tau}) \quad i \in \Psi \setminus G \quad (16)$$

$$L_i = \max(0, \sum_{j \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} \tau \theta_{ij, \tau} - \lambda_i) \\ i \in \Psi \setminus G \quad (17)$$

$$\sum_{i \in \Psi \setminus G} \sum_{\tau=t_0}^{t_n} [\theta_{ij, \tau} (T_{ij, \tau} + \tau)] = \sum_{k \in \Psi \setminus M} \sum_{\tau=t_0}^{t_n} \tau \theta_{kj, \tau} \\ j \in \Psi \setminus \{G, M\}, i \neq j, j \neq k \quad (18)$$

式中: C_f 为车辆使用的固定成本; i, j 为灾害点; M 为所有车辆的出发节点; Ψ 为所有灾害及救援节点; τ, t_0, t_n 为时刻点; $\theta_{ij, \tau}$ 为从 τ 时刻开始救援车辆从灾害点 i 到达灾害点 j , $\theta_{ij, \tau} = \begin{cases} 1 & \text{在时刻 } \tau \text{ 完成路弧}(i, j) \\ 0 & \text{其他} \end{cases}$; G 为所有车辆的到达节点; $T_{ij, \tau}$ 为从 τ 时刻开始从灾害点 i 到达灾害点 j 的时间; W_e 为过早到达灾害点的惩罚成本; E_i 为在灾害点 i 的等待时间; W_l 为延迟到达灾害点的惩罚成本; L_i 为延迟到达灾害点 i 的时间; M_0 为车辆的出发节点 0; M_1 为车辆的出发节点 1; G_0 为车辆的到达节点 0; G_1 为车辆的到达节点 1; Q 为使用车辆的集合; $\alpha_{i, k}$ 为灾害点 i 由车辆 k 实施救援, $\alpha_{i, k} = \begin{cases} 1 & \text{当灾害点 } i \text{ 获得车辆 } k \text{ 救援} \\ 0 & \text{其他} \end{cases}$, k 为车辆; d_j 为灾害点 j 的需求量; ω_i 为车辆离开灾害点 i 后的载重量; K 为所有车辆总和; S_k 为车辆 k 的承载能力; R 为车辆路径总成本; P_i 为第 i 个灾害点被选上的概率; λ_i 为期望到达灾害点 i 的时间。

式(1)是目标函数,用于最小化总成本,包括医疗救援车辆成本、路径成本和惩罚成本。之后的约束条件分为 3 类,第 1 类是关于医疗救援车辆的,由式(2)~(9)决定;第 2 类是关于灾害点特征的,由式(10)~(17)决定;第 3 类是关于路径特征的,由式(18)决定。式(2)和(3)表示医疗救援车辆在软时间窗的约束下从救援站或者其中一个灾害点开出;式(4)和(5)表示所有车辆都会回到救援站;式(6)~(8)表示车辆的承载能力;式(9)表示在救援计划完成前所有车辆都会返回到救援站;式(10)表示所有灾害点都会得到救助;式(11)和(12)表示每个灾害点仅由 1 部医疗救援车辆实施救助;式(13)决定了医疗救援车辆对任务分配的执行条件;式(14)和(15)表示 2 个相邻的灾害点由同一部救援车辆实施救助;式(16)和(17)用于计算由于违背软时间窗而带来的惩罚成本。方程式(13)决定路径的约束条件。

3 模型的求解

3.1 初始化

医疗救援路径通过一定的编码来分析,以便灾害点能够以一定的软时间窗顺序得到医疗救助,例如:有 6 个灾害点需要医疗救助,分别为 $D_1, D_2, D_3, D_4, D_5, D_6$,如果令 A_0 为救援站,而路径编码为 $(A_0, D_3, D_5, D_2, A_0, D_6, D_1, D_4, A_0)$,可以看出,在软时间窗的约束下有 2 条路径可选,安排 2 部车辆分别实施救援,车辆 1 救援路径为: $A_0 \rightarrow D_3 \rightarrow D_5 \rightarrow D_2 \rightarrow A_0$,车辆 2 救援路径为: $A_0 \rightarrow D_6 \rightarrow D_1 \rightarrow D_4 \rightarrow A_0$,如果存在 n 部救援车辆,则每条染色体将有 n 条链接。在初始化阶段,通过以下 3 个步骤产生可行的初始化结果。①分配灾害点到每个链接上,实质是分组问题,灾害点将按与救援站的物理区域进行分组,目标是使总体的车辆行驶时间最小化。例如有 2 部车辆 v_1, v_2 用于医疗救援,救援站为 A_0 ,将每个灾害点 $i(x_i, y_i)$ 按如下的原则进行分组:以救援站 A_0 为圆心设定 x, y 轴,将 x 轴正值区的灾害点 $i(x_i, y_i)$ 分配给 v_1 ,而 x 轴负值区的灾害点 $i(x_i, y_i)$ 分配给 v_2 ,此时的灾害点 $i(x_i, y_i)$ 与救援站的绝对距离为 $D(i, A_0) = [(x_i - x_{A_0})^2 + (y_i - y_{A_0})^2]^{-\frac{1}{2}}$ 。②对救援车辆的路径进行分配,利用 Clarke 等^[9]的节约矩阵 $E(i, j)$ 可以构建灾害点 i, j 与救援车辆的关联。首先假设在 T_0 时刻 2 个灾害点分配在同一个路径链接上,再比较成本节约值 $E(i, j) = D(A_0, i) + D(A_0, j) - D(i, j)$,在保证车辆承载能力的前提下,成本节约值

大的灾害点将得到保留. ③利用近邻启发算法解决车辆路径问题, 其原则是首先随机选择一个灾害点 i , 然后再选择另一个最近的灾害点 j 形成路径, 直至

所有的灾害点都覆盖到为止.

通过以上步骤可以实现问题的初始化, 参照图 1 中的数据, 可以得到具体初始化结果如图 2 所示.

(x, y)	(10,8)	(13,10)	(7,2)	(8,15)	(7,12)	(12,16)	(12,4)	(8,9)	(13,2)	(14,6)	(7,17)	(13,7)	(7,7)	(14,13)	(7,11)
需求量	13	3	2	2	3	3	2	3	3	2	3	6	8	7	5
时间窗	[0,∞]	[13,25]	[17,38]	[13,23]	[9,16]	[8,15]	[16,38]	[16,31]	[17,35]	[16,30]	[12,20]	[12,28]	[16,36]	[12,18]	[15,26]
灾害点	A_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}

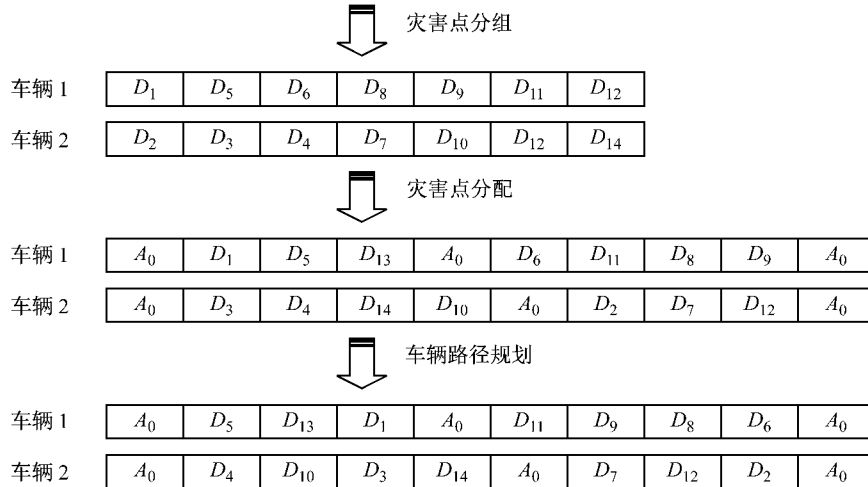


图 2 混合遗传算法(HGA)初始化过程

Fig.2 Initialization process of hybrid genetic algorithm

3.2 遗传迭代交换

此时引入最优搜索运算, 对每 2 个灾害点进行遗传交换比较, 如果产生的结果比初始状态好, 则初始状态将被取代从而形成父代, 直至所有的灾害点的遗传交换比较完成, 最终产生新的父代. 运算过程将消耗一定的时间, 而迭代交换运算可以改进父代与子代之间的链接, 需要通过以下 5 个步骤实现: ①从父代中随机选择 2 个遗传基因; ②交换 2 个遗传基因的位置从而形成子代; ③互换 2 个遗传基因的相邻基因从而形成更多的子代; ④评估所有子代并找到最好的一个; ⑤如果最好的子代比父代好, 则替换父代并重复步骤①, 否则停止. 图 3 展示这种迭代交换过程.

3.3 救援时间评价

这里需要最小化车辆的救援时间, 由于每辆车的救援完成时间各不相同, 所以必需以最晚完成救援的车辆作为救援结束的时刻. 令 T_{va} 为车辆 v 所

耗费的所有时间, $T_{va} = \sum_{i=1}^n T_{v_i,a}$, n 为救援车辆的数量, 而 $T_{vm}(X_w)$ 为车辆 v_i 所耗费的最长时间, X_w 代表第 w 个染色体, 则有 $T_{va} = \sum_{v=1}^{n_v} \{t(D_{n_c}, A_0) + \sum_{i=1}^{n_c} t(D_{i-1}, D_i)\}$, $T_{vm}(X_w) =$

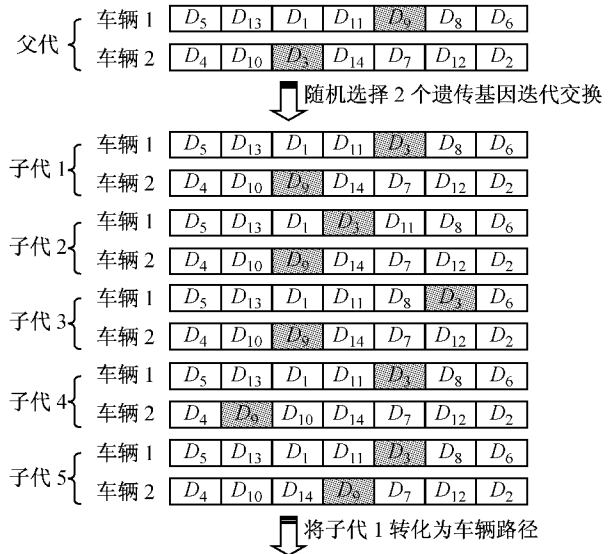


图 3 混合遗传算法(HGA)迭代交换过程

Fig.3 Iterative exchange process of hybrid genetic algorithm

$\max(T_{v_1,a}, T_{v_2,a}, \dots, T_{v_n,a})$, 式中: n_v 为所有救援车辆数目; $t(i, j)$ 为从灾害点 i 到 j 所耗费时间, $t(i, j) = \frac{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{V}$, V 为车辆行驶速度; n_c 为在一条救援路径上灾害点的数量.

3.4 染色体选择

这里需要选择一些染色体用于遗传运算,采用轮盘赌法选择健康的染色体,越健康的染色体被选上的机会越大,也就是说,染色体被选上的机率与其健康程度成正比。染色体选择通过以下步骤实现:

- ①计算所有染色体的健康程度 $U_{fit} = \sum_{\omega=1}^{p_{size}} T_{vm}(X_{\omega})$, p_{size} 为被选染色体的规模;
- ②计算每个染色体 X_{ω} 被选择上的概率 $p_{\omega} = [U_{fit} - T_{vm}(X_{\omega})] / [U_{fit}(p_{size} - 1)]$, $\omega = 1, 2, \dots, p_{size}$;
- ③计算每个染色体 X_{ω} 的累积概率 $P_{\omega} = \sum_{i=1}^{\omega} p_i$;
- ④在范围 $(0, 1]$ 中产生任意的随机数 λ , 如果 $P_{\omega} \geq \lambda > P_{\omega-1}$, 则染色体 X_{ω} 将被选择。

3.5 遗传运算

遗传交叉算子可以找到更好的解决方案,而遗传变异算子可以扩展更大的搜索空间。下文用交叉算子、启发式变异和反转突变来产生改进的子代染色体。

3.5.1 顺序交叉

通过遗传交叉每次可产生 2 个子代染色体,可以通过以下步骤实现:①从第 1 组父代中随机选择 1 组基因子串;②复制该基因子串到原子串对应的位置,产生 1 组新的子串;③删除第 2 组父代中相应位置的基因,该位置的基因形成序列;④从左到右将原子串中空出位置按前步的序列顺序填满基因,从而形成一个新的子代;⑤通过交换 2 组父代的位置,重复步骤①~④的过程可产生另外一个子代。具体的遗传交叉过程如图 4 所示。

3.5.2 启发式变异

可以通过以下步骤实现:①从父代中随机选出 3 个遗传基因;②产生被选择基因所有可能的邻里排列,并生成所有的子代。具体的启发式变异过程如图 5 所示。

3.5.3 反转突变

逆算子作为一种变异运算仅仅用来增加样本的多样性,而不是加强样本的质量。具体的反转突变过程如图 6。

4 数据仿真分析

2008 年发生在我国四川汶川地区的里氏 8.0 级特大地震给灾区人民带来严重的生命和财产损失,此时需要各个医疗救援站积极响应,使医疗急救物资迅速运输到位。而各个救援站的车辆资源是有限的,需要合理调度,故此以非常规突发灾害事件为案

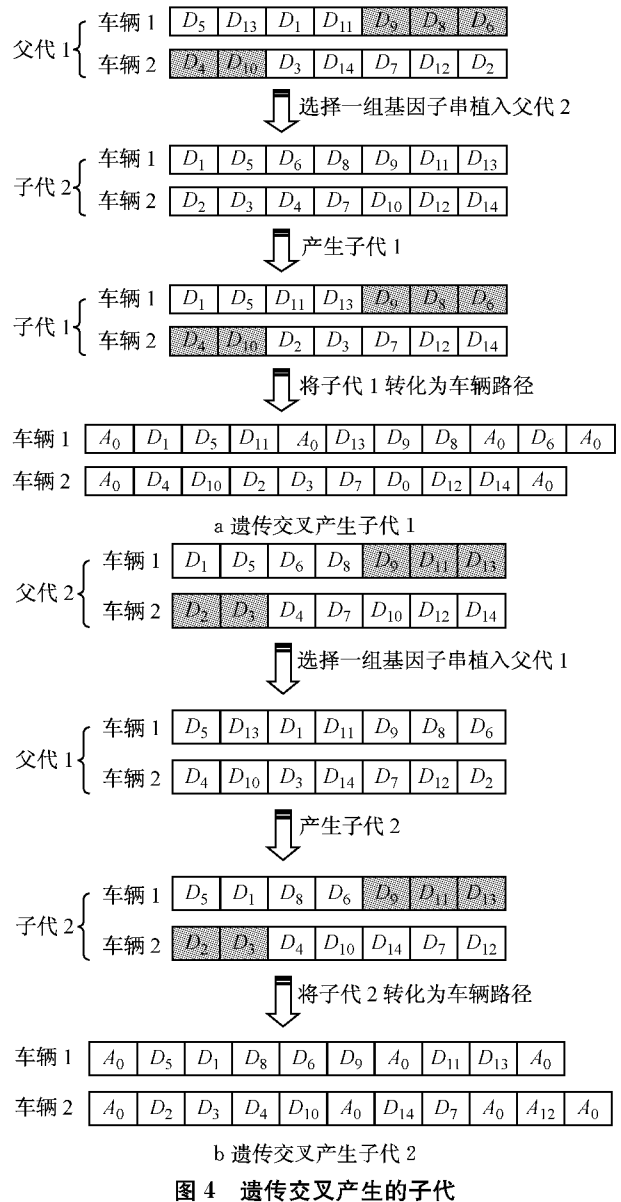


图 4 遗传交叉产生的子代
Fig. 4 The offspring of genetic crossover

例,依据前面的数学模型通过数据仿真分析 HGA 算法的有效性并与 Azi 等^[10]的精确算法(exact algorithm, EA)进行比较,说明其在处理带软时间窗的路径搜索问题方面的优势。EA 算法可以精确到 20 个灾害点的情形,而灾害点大于 20 个时 EA 算法的计算时间会很长,在这里取某救援站所属区域有 20 个灾害点的情形进行比较分析, HGA 算法采用 MATLAB 7. 1 编程、Intel Core P8400/2. 26GHz 处理器运算,而 EA 算法利用 AMPL Plus 1. 5/CPLEX 求解。在整个仿真测试过程中,假设有 2 部救援车辆用于医疗救助,各车辆的承载能力与灾害点的坐标及软时间窗都采用计算机随机生成,而 HGA 的初始数据也通过计算机随机生成, HGA 迭代次数为 200 次,车辆行驶的时间间隔为 2 个单位,

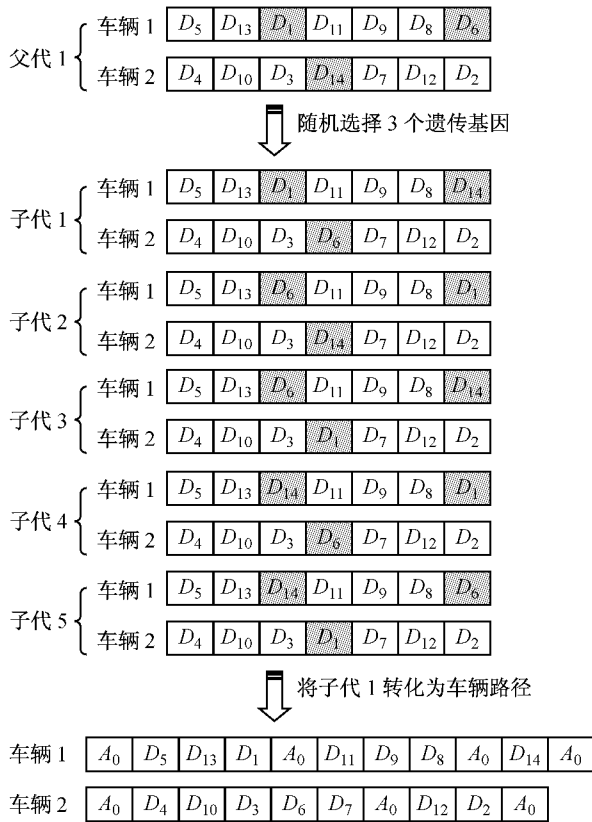


图5 启发式变异过程

Fig.5 Heuristic mutation process

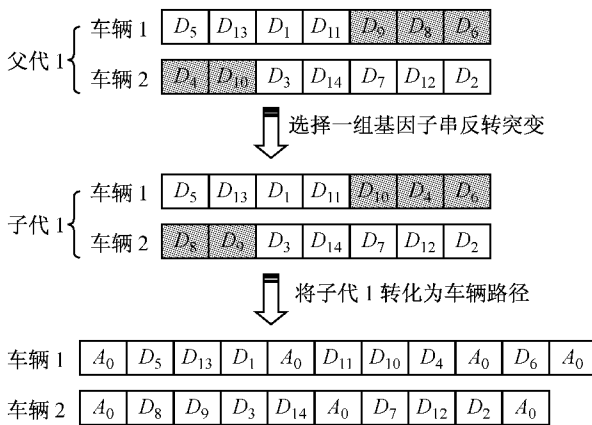


图6 反转突变过程

Fig.6 Inversion mutation process

超出时间窗的惩罚成本 W_e 和 W_1 各取 20 个单位。

首先比较 HGA 和 EA 算法的总成本缺口率 $C_{gap} = (C_{HGA} - C_{EA}) / C_{EA} \times 100\%$, 其中 C_{HGA} , C_{EA} 分别为 HGA 和 EA 算法的总体成本消耗; 运算时间的比率 $R_{EH} = T_{EA} / T_{HGA}$, 其中 T_{EA} , T_{HGA} 分别为 EA 和 HGA 算法的运算时间, 于是可以计算比较 2 种算法的总成本消耗和运算时间, 由于灾害点小于 4 的比较意义不大, 故随机采样数据计算从 4 个灾害点开始, 如表 1 所示。

从表 1 可以看出, 当灾害点的数量小于 7 个时, HGA 和 EA 算法的总成本是相当的, 而随着灾害点数量的增大, HGA 和 EA 算法之间的总成本缺口发生变化, 在 20 个灾害点时, 总成本缺口达到了 4.2%, 在 10 个灾害点以内时, HGA 和 EA 算法的运算时间差别不太大, 而随着灾害点数量的增大, HGA 算法体现出明显的时间优势, EA 算法的运算时间将大量增加, 而在 20 个灾害点时, EA 算法运算所耗费的时间是 HGA 算法的 133.3 倍。

表 1 HGA 和 EA 算法的总成本及时间消耗比较

Tab.1 Comparison of the total cost and time by HGA and EA algorithm

灾害点数量	C_{HGA}	T_{HGA}/s	C_{EA}	T_{EA}/s	$C_{gap}/\%$	R_{EH}
4	324.5	0.2	324.5	0.6	0	3.0
5	316.2	0.4	316.2	1.5	0	3.8
6	313.6	0.7	313.6	21.7	0	31.0
7	273.1	0.8	273.1	6.1	0	7.6
8	283.5	1.2	279.6	11.2	1.4	9.3
9	472.3	2.9	471.7	53.6	0.2	18.5
10	475.9	4.8	473.6	66.1	0.5	13.8
11	493.7	5.6	482.5	126.3	2.3	22.6
12	497.5	6.5	491.4	196.4	1.3	30.2
13	508.4	7.2	495.3	228.7	2.6	31.8
14	517.2	8.1	506.7	316.9	2.1	39.1
15	521.8	8.6	510.4	409.3	2.3	47.6
16	558.6	9.8	547.1	559.9	2.1	57.1
17	592.1	10.9	581.5	728.9	1.8	66.9
18	650.8	12.1	639.6	1 209.2	1.7	99.9
19	729.7	12.4	702.9	1 448.9	3.8	116.8
20	772.5	14.7	741.3	1 959.4	4.2	133.3

随着 HGA 算法迭代次数的增大医疗救援车辆的行驶时间进一步降低, 从而优化总成本消耗和救援运输时间. 下面分析 HGA 算法在 20 个灾害点的情况下, 迭代运算 200 次, 交叉率取 0.4, 突变率取 0.2, 随机取 3 对染色体进行遗传交叉, 其中 3 个染色体作遗传变异和反转突变, 于是每次迭代将产生 18 个子代, 这样重复迭代 200 次可以得到如图 7 的结果。

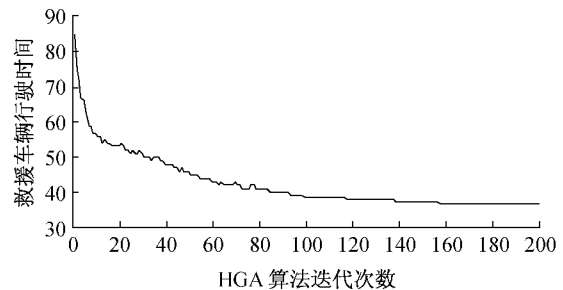


图7 HGA 算法 200 次迭代收敛

Fig.7 200 iterations converge of HGA algorithm

通过图 7 可以看出,随着 HGA 算法迭代次数的增加,救援车辆的行驶时间的优化趋于稳定,经过 200 次的迭代运算后,救援车辆的实际行驶时间从最初的 85 个单位缩减到 37 个单位.

5 结语

面对突发性非常规灾害事件,医疗急救物资的及时运输显得尤为重要,将有益于减少突发事件下的伤亡.针对软时间窗约束下的多辆医疗急救车辆的 VRPTW 问题建立了数学模型,得到最优总成本消耗下对各个灾害点救援的车辆行驶路径,同时有效缩短救援车辆的实际行驶时间.通过与 EA 算法进行数据仿真计算比较,结果表明 HGA 算法在总体成本消耗上与 EA 算法相当,但运算时间方面具有明显的优势.

参考文献:

- [1] Desaulniers G, Lavigne J, Soumis F. Multi-depot vehicle scheduling problems with time windows and waiting costs[J]. *European Journal of Operational Research*, 1998, 111(3): 479.
- [2] Qureshi A G, Taniguchi E, Yamada T. An exact solution approach for vehicle routing and scheduling problems with soft time windows[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2009, 45(6): 960.
- [3] Ghoseiri K, Ghannadpour S F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm [J]. *Applied Soft Computing*, 2010, 10(4): 1096.
- [4] Taillard E D, Laporte G, Gendreau M. Vehicle routing with multiple use of vehicles [J]. *Journal of the Operational Research Society*, 1996, 47: 1065.
- [5] Campbell A M, Savelsbergh M. Efficient insertion heuristics for vehicle routing and scheduling problems[J]. *Transportation Science*, 2004, 38: 369.
- [6] Marinakis Y, Marinaki M, Dounias G. A hybrid particle swarm optimization algorithm for the vehicle routing problem [J]. *Engineering Applications of Artificial Intelligence*, 2010, 23(4): 463.
- [7] Tan K C, Lee L H, Ou K. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints [J]. *Engineering Applications of Artificial Intelligence*, 2001, 14(6): 825.
- [8] Park Y B. A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines[J]. *International Journal of Production Economics*, 2001, 73(2): 175.
- [9] Clarke G, Wright J W. Scheduling of vehicles from a central depot to a number of delivery points[J]. *Operations Research*, 1964, 12: 568.
- [10] Azi N, Gendreau M, Potvin J Y. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles[J]. *European Journal of Operational Research*, 2010, 202(3): 756.