

应急物资配送动态调度的粒子群算法

田 军, 马文正, 汪应洛, 王刊良

(西安交通大学 管理学院, 西安 710049)

摘 要 应急条件下的物资配送与调度面临着需求信息不准确、需求紧急程度差异和运输路网动态变化的复杂环境, 借助模糊数学中的三角模糊数描述应急物资需求量, 利用连续速度时间依赖函数模拟真实的动态路网交通状况, 并考虑不同需求点的需求紧急程度差异, 建立了针对性的应急物资配送动态调度的多目标数学模型; 通过设计粒子群优化算法, 采用“离散-连续向量混合编码”方案和加权整合的适应值函数导向机制, 结合连续更新的位置和速度操作策略, 建立了针对这类含有离散和连续变量组合的优化模型的快速高效求解算法; 最后, 结合两个实际的算例进行了数值实验与分析, 通过与用 Matlab 求得的解析解的比较, 证明算法收敛速度快、鲁棒性强, 从而为应急条件下的物资配送动态调度提供了有效和可靠的方法。

关键词 应急物资配送; 动态车辆路径规划; 多目标优化调度; 粒子群算法

Emergency supplies distributing and vehicle routes programming based on particle swarm optimization

TIAN Jun, MA Wen-zheng, WANG Ying-luo, WANG Kan-liang

(School of Management, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract Fuzzy demands, dynamic transportation network, and prioritized supplying and distribution are the typical characters of emergency supplies under eruption incident(s). Making use of the triangular fuzzy numbers of fuzzy mathematics to describe the emergency demands for supplies, and the speed of time-dependent function to simulate the real dynamics states of the road network, as well as thinking of the prioritized condition, the paper set up an optimization model for scheduling emergency supplies, which had a multi-objective and could choose a road to avoid congestion in special period of time, and strictly meet the needs of time requirement of the real supply orders. Then a Particle Swarm Optimization algorithm has been designed according to the characteristics of the model. A vector encoding method with “discrete - continuous variables encoding” for every particle was put forward, a Fitness Function was proposed according to the objective functions, a strategy for continues renewing the location and velocity of each particle was proposed. It could better solve the problems of discrete and continuous combinatorial optimization variables for the mathematical model of the emergency dispatch. With the help of an empirical analysis in combination with two examples, the algorithms was proved suitable and validity when the direct comparison of exact solutions obtained by the MATLAB was done.

Keywords emergency supplies distribution; dynamic vehicle routing programming; multi-criteria scheduling; particle swarm optimization

1 引言

在突发灾害的应急处置过程中, 应急物资的供应是最为重要的工作之一, 快速有效的物资调度与配送, 不仅是应急救援过程有效开展的前提和条件, 同时也是减少人民生命财产损失、防止次生灾害发生、保障社会

收稿日期: 2009-11-07

资助项目: 国家自然科学基金-(70971108)

作者简介: 田军 (1964-), 男, 博士, 副教授, 研究方向: 突发灾害应急管理, 物流与调度管理, 决策与优化, 管理信息系统与决策支持系统等; 马文正 (1984-), 男, 硕士研究生; 汪应洛, 男, 教授, 博士生导师, 中国工程院院士, 研究方向: 工业工程与工程管理; 王刊良 (1966-), 男, 教授, 博士生导师, 研究方向: 信息管理与电子商务。

稳定等持续发展的必要支撑。

Kemball-Cook^[1] 在 1984 年首先提出对于救援物资的运输和供应需要进行针对性的管理, 此后, 国内外学者对应急物资调度的问题进行了深入地探讨, 作了许多有意义的研究. Sheu^[2] 建立了一个基于混合模糊聚类方法的应急物资调配和配送模型, 用于紧急分配救援关键时期的应急物资; Linet^[3] 建立了一个需求时间延迟最小的数学模型, 用于处理应急派送的次序问题; Meishiang^[4] 研究建立了基于地理信息系的洪灾应急物资临时配送中心, 针对营救小分队的物资分配模型; Barbarosoglu^[5] 构建了最初批次应急物资调度计划的两阶段随机规划模型; 何建敏^[6] 以“时间最短、出行点数最少”为目标, 建立了两阶段多目标模型; 缪成^[7] 则构建了一个以“最小化车辆行驶费用、运输方式转换费用、及需求延迟满足目标函数增加值”为目标的数学规划模型。

Teodorovic^[8] 研究了单车场情况下, 具有模糊顾客需求的商业物流车辆路径问题, 利用三角模糊数描述需求信息的不确定性, 并运用 Gillett^[9] 提出的 Sweeping 算法进行求解. Fox^[10] 将路网动态性引入旅行商问题, 对应急物流的调度具有较好的指导意义. Malandraki^[11] 进一步研究了动态路网的函数表达问题, 通过建立行驶时间分段依赖函数, 构建基于动态时间参数计算的车辆路径选择问题, 并给出了车辆行驶时间分段依赖函数模型. 在此基础上, Soumia^[12] 发现可以用行驶速度函数代替行驶时间函数, 通过路段长度和行驶速度来计算行驶时间, 能够更好的模拟行驶时间的连续变化. 李宁^[13] 研究了带有时间窗约束的车辆路径问题, 即在客户要求的时间窗内将物资送达, 从而为处理物资配送的轻重缓急提供较好地借鉴, 为了使车辆尽量在时间窗内到达, 当车辆在时间窗之前到达时, 要让车辆进行等待或用惩罚机制进行调整。

从实际需求来看, 应急条件下的物资配送与调度存在三个方面的主要困难: 一是需求信息的不准确与不确定性, 物资需求不可能进行精确的计算和测量, 多数情况下是按照大致条件进行估算, 数据的误差和变化范围较大; 二是灾害过程可能对运输线路造成影响, 导致运输路网发生动态改变, 同时车辆的行驶速度也会受到天气、灾情、及交通流量等的限制, 具体车辆的运行时间难以精确掌控; 三是物资需求点的需求紧急程度不同, 给物资配送带来排序和调度上的困难. 因此, 应急物资的配送和调度比一般物资调配问题要复杂得多, 而且, 其建模和求解面临着强时间压力条件的严格限制, 时间和效率至为关键, 因此, 探索建模和求解的高效率算法具有重要意义。

2 应急物资配送的动态调度模型构建

2.1 问题描述

考虑一个应急物资配送中心 (大型仓库) 的情况 (如图 1), 当灾情发生后, 需要将存储的物资紧急配送到若干需求点, 假设仓库的物资储备充足, 但是配送车辆数目有限, 每辆车的最大载重量是固定的. 各需求点的地理位置坐标可以通过地理信息系统或相关数据库获得, 配送中心与各个物资需求点、以及各物资需求点之间的运输路线和距离数据也可以同时获取, 但是, 由于灾害突然降临并持续发展, 运输路网可能遭到破坏, 可行运输路线处于动态变化之中 (相应的数据可以及时获取). 各物资需求点的物资需求数量以及需求紧急程度, 可以通过现场估算并及时将信息传递到物资配送中心, 但是需求量的估算并不十分准确, 需求的紧急程度也处于变化之中. 配送中心遵循的调度规则是: 每个物资需求点的需求都必须得到满足, 需求紧急程度高的需求点要优先配送, 紧急程度相同时, 以运送距离为基准, 离配送中心近的需求点将优先获得配送。

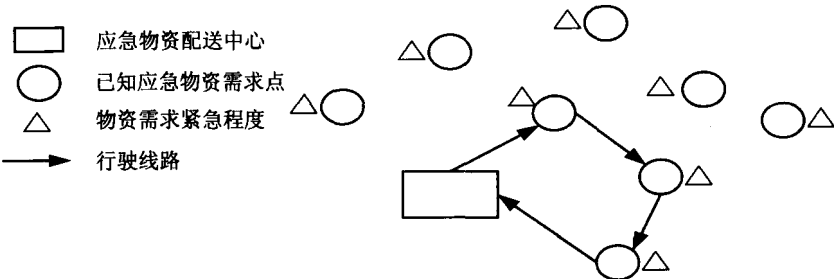


图 1 应急物资配送的动态调度示意图

2.2 不确定需求信息和车辆行驶速度的处理

1) 不确定需求量的处理. 应急物资的需求数量可以通过直接评估或者需求点申请得到, 通常可以被描述

为“2 吨左右”或“2 到 3 吨之间”等不确定性语言, 借鉴 Teodorovic 的思路^[8], 采用模糊方法来处理该类信息, 用三角模糊数 $\tilde{p}_i = (m_{1i}, m_{2i}, m_{3i})$ 表示应急需求点 i 的物资需求量 (如图 2 所示), 其中 m_{1i} 和 m_{3i} 表示三角模糊数的左、右边界, m_{2i} 表示偏好量 (即实际需求的评估量或者申报量). 需求量的隶属度函数为:

$$\mu_{\tilde{p}_i}(m_i) = \begin{cases} 0, & m_i < m_{1i} \\ \frac{m_i - m_{1i}}{m_{2i} - m_{1i}}, & m_{1i} \leq m_i < m_{2i} \\ \frac{m_{2i} - m_i}{m_{2i} - m_{3i}}, & m_{2i} \leq m_i < m_{3i} \\ 0, & m_i \geq m_{3i} \end{cases} \quad (1)$$

2) 车辆行驶速度的动态处理. Soumia^[12] 认为, 车辆行驶速度的变化不是跃升改变, 而是平滑改变, 他给出了运输路线早高峰、晚高峰连续行驶速度依赖函数 (见图 3). 理论上讲, 只要速度函数是连续变化的, 车辆的运行时间就能够计算 (采用积分方法). 在此, 可以借鉴 Soumia 的速度依赖函数来模拟应急条件下的车辆运输行驶速度的变化.

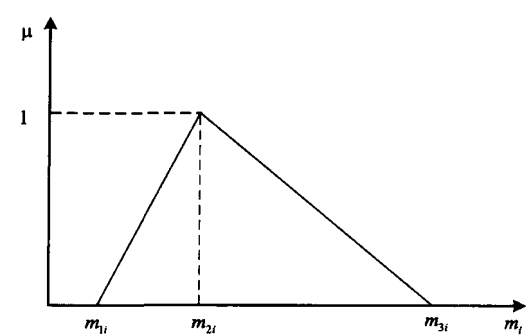


图 2 需求量的三角模糊数

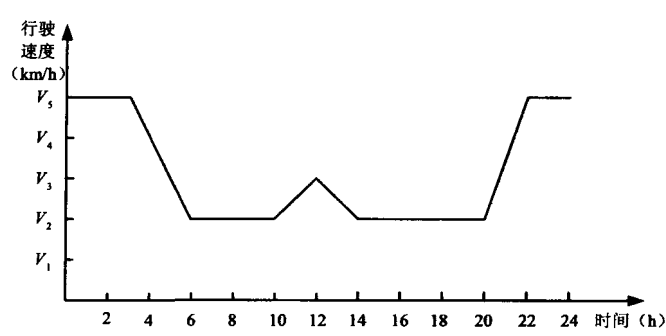


图 3 Soumia 连续行驶速度依赖函数

2.3 模糊需求、动态路网、按优先次序配送的应急物资调度优化模型

由于突发事件的弱经济性, 应急条件下车辆行驶成本将变得不再重要, 车辆行驶的时间将成为最为重要的因素, 因此最优调度策略应该是将车辆行驶总时间最短作为调度的主要目标. 与此同时, 配送数量 (卸货量) 应最大限度地满足各需求点的需求量, 所以将送货量尽量接近偏好需求量作为次要目标. 因此, 应急物资配送的目标函数可以描述为:

$$\min f_1 = \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N (t_{ij} x_{ijk}) + \sum_{i=1}^N T_i \quad (2)$$

$$\min f_2 = 1 - \frac{1}{N} \sum_{i=1}^N \mu_{\tilde{p}_i}(m_i) \quad (3)$$

受到的约束条件包括:

$$\sum_{i=1}^N m_i y_{ik} \leq Q, \quad \forall k = 1, 2, \dots, K \quad (4)$$

$$LT_i^r \leq ET_j^{r+1}, \quad i = 1, 2, \dots, N; j = 1, 2, \dots, N; r = 1, 2, 3 \quad (5)$$

$$\sum_{k=1}^K y_{ik} = 1, \quad i = 1, 2, \dots, N \quad (6)$$

$$\sum_{i=0}^N x_{ijk} - \sum_{j=0}^N x_{jik} = 0, \quad \forall k = 1, 2, \dots, K \quad (7)$$

$$\sum_{i=0}^N x_{ijk} = y_{jk}, \quad j = 1, 2, \dots, N; \forall k = 1, 2, \dots, K \quad (8)$$

$$\sum_{j=0}^N x_{ijk} = y_{ik}, \quad i = 0, 1, 2, \dots, N; \forall k = 1, 2, \dots, K \quad (9)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subseteq \{0, 1, \dots, N\}, 2 \leq |S| \leq N - 1; \forall k \quad (10)$$

$$m_{1i} \leq m_i \leq m_{3i} \quad (11)$$

$$y_{ik}, x_{ijk} = 1 \text{ 或 } 0 \quad (12)$$

式中:

x_{ijk}, y_{ik}, m_i 为决策变量. m_i 表示在需求点 i 的实际供应量 (卸货数量);

$$y_{ik} = \begin{cases} 1, & \text{应急需求点 } i \text{ 的需求由车辆 } k \text{ 完成,} \\ 0, & \text{其他.} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 从点 } i \text{ 行驶到点 } j, \\ 0, & \text{其他.} \end{cases}$$

N 为应急物资需求点的数目 (i, j 表示需求点编号, $i, j=0$ 表示配送中心);

Q 为车辆的最大装载量;

T_i 为应急物资在第 i 个需求点的卸货时间;

t_{ij} 为需求点 i 和 j 之间的车辆运行时间, 由运输距离和运输速度计算获得;

ET_i^r 为紧急程度相同 (同为 r) 需求点的最早服务时间;

LT_i^r 为紧急程度相同 (同为 r) 需求点的最晚服务时间;

r 为需求紧急程度, $r \in (1, 2, 3, 4)$, 分别表示特急、加急、紧急和一般四种水平.

约束条件说明: 公式 (4) 表示第 k 辆车一次运输中, 在各个卸货点的货物总量应该不大于其有效装载量; 公式 (5) 表示不同紧急程度的需求点之间交货时间的限制, 需求紧急程度高的优先配送, 即级别高的需求点的最晚服务时间应该早于级别低的需求点的最早服务时间; 公式 (6) 表示每一个需求点都必须有一辆车且仅有一辆车进行服务 (需求量多于一辆车载量时, 单独派车处理, 这与车辆调度优化没有关系, 在此简化处理), 所规划的车辆线路必须覆盖所有的需求点; (7) 是一种流量守恒限制式, 表示出入某需求点的车辆数守恒; 公式 (8) 和 (9) 表示如果有一辆车 k 被指派到某一条配送路线, 则必存在一条由配送中心出发再返回配送中心的车辆路线; 式 (10) 是为了消除不应有的支路, 避免形成没有通过配送中心的循环路线; 公式 (11) 严格来说应该在服务可靠性水平 α 下, 应急物资供应满足需求点 i 的物资需求量的指定区间, 即服务可靠性 $Pos\{m_{1i} \leq m_i \leq m_{3i}, i = 1, 2, \dots, N\} \geq \alpha$, 当 $\alpha=100\%$ 时, 就转化为当前的表达式.

3 模型求解的粒子群算法设计

3.1 算法选择依据

由于问题本身比较复杂, 包含模糊信息和众多处于动态变化中的参数, 目标函数和约束条件构成的解空间难以满足用数学解析方法求解的条件 (可行域连续、可微并且为凸集等条件), 所以用传统的数学规划方法求解受到了限制, 因此需要探索适应性强的智能优化方法 (数值模拟方法) 求解; 另一方面, 在应急条件下的物资配送和调度问题受到严苛的时间压力, 对模型求解的效率有更高的要求, 能够针对参数的变化作出快速的求解, 但对于求解结果的精确性要求相对要宽松一些, 因此, 求解模型的算法选择应遵循收敛速度快、占用资源少、鲁棒性强的原则. 从仿生学衍生出来的粒子群算法恰好能够满足该类问题的需要, 其特点是能够避免类似进化算法的复杂遗传操作, 简单易实现, 且收敛速度快. 因此, 选择粒子群算法进行求解.

3.2 粒子群算法简介

粒子群优化算法 (PSO, particle swarm optimization) 是美国心理学家 Kennedy 和电气工程师 Eberhart 受鸟类觅食行为启发而提出的一种基于群体智能理论的新兴技术^[14-15], 它采用简单的速度 - 位移模型, 在适应值函数信息导向下实现基于种群的全局搜索, 其特有的记忆功能可以跟踪搜索路径并动态调整搜索策略. 算法的数学描述为: 在 n 维空间中有 M 个粒子, 每个粒子的位置记为 $X_i = [X_{i1}, X_{i2}, \dots, X_{in}]$, 每个位置对应一个与优化目标函数相关的适应度函数 Fit_i , 每个粒子以各自的速度 $V_i = [V_{i1}, V_{i2}, \dots, V_{in}]$ 发生运动. 对于单个粒子 i 在“飞行”历史中的过去最好位置记为 $P_i = [P_{i1}, P_{i2}, \dots, P_{in}]$, 群体所有粒子经历过的历史最好位置为 $P_g = [P_{g1}, P_{g2}, \dots, P_{gn}]$, 各粒子的速度和位置按如下公式进行更新 (“飞行”)^[13]:

$$V_{id}(t+1) = wV_{id}(t) + c_1r_1(P_{id}(t) - X_{id}(t)) + c_2r_2(P_{gd}(t) - X_{id}(t)) \quad (13)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (14)$$

其中下标 d 表示粒子的维度 ($d \in \{1, 2, \dots, n\}$), w 为惯性权因子, c_1 和 c_2 都是正的加速常数, r_1 和 r_2 是 $[0, 1]$ 之间均匀分布的随机数. 另外, 为了减少粒子离开搜索空间的可能性, 通过设置粒子的速度区间 $[V_{\min}, V_{\max}]$ 和位置范围 $[X_{\min}, X_{\max}]$, 对粒子的移动进行适当限制.

3.3 应急物资调配模型的粒子群算法设计

3.3.1 编码方案设计

如何为粒子找到一个合适的表达方法, 使粒子的位置与问题的解相对应, 是实现算法的关键. 借鉴遗传算法中处理同时含有离散和连续变量组合优化问题的编码方式^[19], 把每个粒子的位置向量编码分为两部分, 即 [离散变量: 连续变量]. 离散变量对应于 N 个物资需求点、 K 辆车的调度问题, 粒子的位置编码为矩阵的前 $N + k - 1$ 列, 其排列次序表示配送顺序; 连续变量部分对应实数向量, 粒子的位置为矩阵的后 N 列, 分别表示 N 个需求点的供应量 (卸货量).

3.3.2 适应度函数设计

问题模型存在两个目标函数, 在求解中需要整合成为一个适应值函数. 但是由于两个目标的量纲不一致, 而且量级存在很大差异, 所以整合过程中首先需要采用归一化方法将两个目标整合成为一个综合目标. 设计采用如下方式进行归一化整合:

$$Z = \rho_1 \frac{TT - TT_{\min}}{TT_{\max} - TT_{\min}} + \rho_2 \left[1 - \frac{1}{N} \sum_{i=1}^N \mu_{\bar{p}_i}(m_i) \right] \quad (15)$$

式中: ρ_1 、 ρ_2 为权重, 表示两个目标的相对重要性, $\rho_1 + \rho_2 = 1$, 实际使用时可以根据现场情况进行设计; TT 表示车辆行驶总时间 (此处 $TT = f_1$), TT_{\min} 表示当前种群中最小车辆行驶总时间, TT_{\max} 表示当前种群中最大车辆行驶总时间. TT_{\min} 和 TT_{\max} 也都是通过 f_1 计算出来的.

在综合目标的基础上, 建立适应度函数如下:

$$Fit_i = \min \left(\frac{1}{Z} \right) + R \times \left(\sum_{k=1}^K \max \left(\sum_{i=1}^N m_i - Q, 0 \right) + \sum_{k=1}^K \max (LT_i^r - ET_j^{r+1}, 0) \right) \quad (16)$$

此处第一项是将综合目标的倒数计入适应值函数, 原因在于, Z 值非常小, 求解过程中在坐标轴上显示的点的位置较低, 差异不明显, 而采用倒数的方式则可以更好地显示出差异. 第二和第三项是把约束条件的前两项 (即公式 (4) 和 (5)) 整合到适应值函数中去, 其中: R 是一个比较大的正数, 是为了强化该项的影响. 约束条件的后几项 (公式 (6) 到 (12)) 没有整合进适应值函数, 是考虑到这些约束条件是对变量本身取值的限制 (即对变量定义域的限制), 在算法求解过程中, 可以通过两种途径加以解决, 一是进行编码范围的限制, 参见 3.1 节最后部分; 二是算法求解过程中, 在更新粒子状态之前首先判断更新后得到的新位置是否符合约束条件, 对于产生的不满足定义域约束条件的解, 首先进行剔除, 然后再继续后续步骤. 这种处理方式可以简化适应值函数表达, 从而给求解过程带来方便.

3.3.3 位置与速度操作方案

粒子的速度用于改变粒子的位置, 可以看作是对粒子位置的一种变换. 在离散、连续组合优化问题中, 粒子的状态由整数和实数混合排列表示, 实数部分可以继续用原来的速度定义, 而整数部分速度定义和连续问题不同, 则需要重新定义. 根据 Clerc^[16] 对离散问题编码操作的研究, 参照吴斌提出的变换思路^[17], 可以采用下面的操作定义粒子的位置和速度:

① **位置间的加法** 定义 $XO(x_i, x_j)$ 为粒子状态 $X(x_1, x_2, \dots, x_n)$ 上的交换, 表示 x_i 和 x_j 两数作 2-Opt 运算, 即将 X 向量的第 i 维和第 j 维交换位置. 例如: $X = (1, 6, 9, 7, 4)$, $XO(1, 6)$, 则 $X' = X \oplus XO = (6, 1, 9, 7, 4)$. 符号 “ \oplus ” 用来表示位置间的加法.

② **粒子的速度** 粒子的速度可以用一个或者多个交换数的排列来表示, 即 $V = (XO_1, XO_2, \dots, XO_n)$.

③ **位置与速度的加法** 在更新粒子的位置时, 粒子的新状态按照交换数的顺序将 X 的元素作 2-Opt 运算, 即 $X_{new} = X \oplus V$, 例如: $X = (1, 2, 3, 4, 5)$, $V = ((1, 3), (5, 2))$, 则 $X' = X \oplus V = (3, 5, 1, 4, 2)$. 其中粒子的速度具有方向性, 不满足交换律.

④ **速度的加法** 两个速度 V_1 、 V_2 的和定义为两个速度中交换数的并集, 且不改变交换数的顺序, 其中速度的和不具备交换性, 即 $V_1 \oplus V_2$ 与 $V_2 \oplus V_1$ 产生的结果不同, 这是由速度的方向性决定的.

⑤速度的数乘 速度的数乘表示为 $V_2 = c \cdot V_1$, 此处用符号 “.” 来表示速度的数乘. 其中 c 是一个常数 ($0 < c \leq 1$), 则 $V_2 = c \cdot V_1 = (XO_1, XO_2, \dots, XO_{c\|V\|})$, 即从 V 中选取前 $c\|V\|$ 个交换数作为新的速度. 其中 $\|V\|$ 表示粒子的长度, 指速度中包含的交换数的个数.

⑥位置间的减法 两个位置相减的结果是一个速度, 即 $V = X_2 \odot X_1$, 符号 “ \odot ” 表示位置间的减法. 在操作过程中, 逐一比较 X_1 和 X_2 的各分量 $x_{1,i}$ 和 $x_{2,i}$, 如果两者相同, 则继续比较下一组分量; 如不相同, 将两个分量作为一个交换数, 同时将 X_2 中的分量按照这个交换数做 2-Opt, 则 V 的第 i 个分量为: $v_i = ((x_{1,i}, x_{2,i}))$

3.3.4 粒子速度和位置的更新公式

通过对粒子位置和速度的特殊定义, 将 “ \oplus ”、“ \odot ”、“.” 代入到 PSO 速度和位置更新公式中, 得到离散 PSO 优化操作下的粒子速度和位置的更新公式:

$$V_{id}(t+1) = w \cdot V_{id}(t) \oplus c_1 \cdot r_1 \cdot (P_{id}(t) \odot X_{id}(t)) \oplus c_2 \cdot r_2 \cdot (P_{gd}(t) \odot X_{id}(t))$$

(17)

$$X_{id}(t+1) = X_{id}(t) \oplus V_{id}(t+1)$$

(18)

离散、连续组合优化问题中, 每个粒子向量速度和位置的更新分为两部分, 离散变量部分采用 (17)、(18) 式进行更新; 连续变量部分利用 (13)、(14) 式进行更新.

3.3.5 基于粒子群算法求解应急物资调度问题求解流程

用粒子群算法求解应急物资调度问题的流程如图 4 所示.

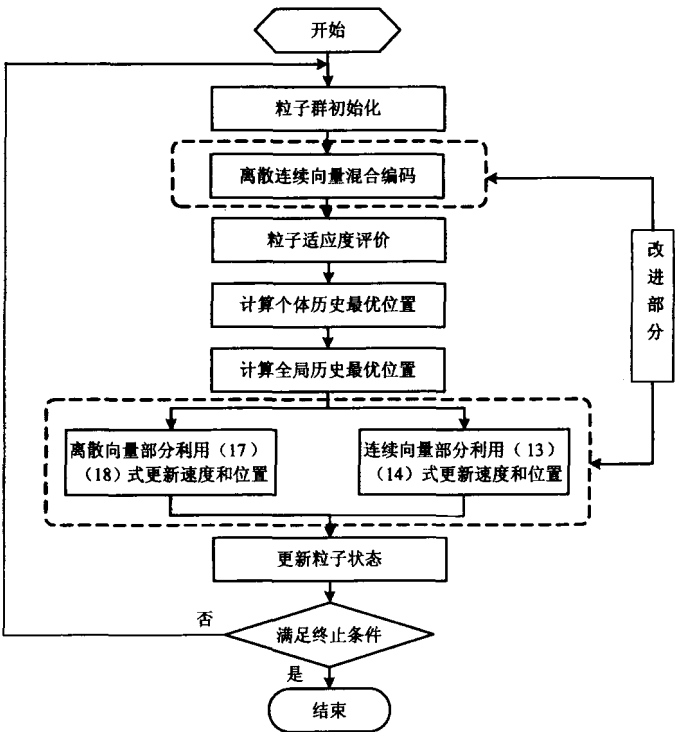


图 4 应急物资配送动态调度问题求解的粒子群算法流程图

3.4 粒子群算法复杂度分析

种群规模为 K , 迭代次数为 L , 问题规模为 N (需求点数量) 的应急物资调度问题, 其计算复杂度分析如下: 每次迭代时, 都要经过解码, 适应度计算, 个体、群体最优状态选择和速度计算, 状态更新. 整个算法的时间复杂度为:

$$O(K, L, N) = L * (O(K * N^2) + O(K * N) + O(1) + O(K) + O(2K * N^2 + K * N^2))$$

$$\approx L * (O(4K * N^2)) \approx L * K * O(N^2).$$

从上式可知, 粒子群算法的计算量与研究问题规模的平方成正比, 与种群规模和迭代次数成正比. 算法的计算量和其它启发式算法在同一个量级水平上.

4 算法试验与结果分析

4.1 算例选择与实验参数设计

采用文献 [17] 和文献 [18] 提供的算例 (规模不相同) 分别对本文设计的粒子群求解算法进行检验. 各需求的紧急程度采用计算机模拟方式随机给出, 假设不同等级间道路的速度依赖函数都符合 Sourmia 速度依赖, 三个主要速度参数 (参见图 3) 分别为: $V_2=20\text{Km/h}$, $V_3=30\text{Km/h}$, $V_5=40\text{Km/h}$.

文献 [17] 的问题是某物资调度中心, 8 个物资需求点, 由容量为 8 吨的同种类运输车辆配送的情况, 物资需求的特征和需求数量如表 1 所示; 文献 [18] 是北京发生 “SARS” 疫情时, 一批应急医疗物资空运到北京首都机场, 然后需要紧急将这些物资调运到城区的 16 家大型医院, 需求特征和数量由表 2 给出, 其中 g_i 是模糊需求量、 T_i 表示卸货时间、 r 表示物资需求紧急程度.

表 1 算例 1 的应急物资需求特征和数量

应急需求点	A1	A2	A3	A4	A5	A6	A7	A8
g_i (吨)	(1.5,2,2.2)	(1.2,1.5,2)	(3.5,3.5,3.7)	(2.5,3,3.2)	(1.2,1.5,1.7)	(2.5,3,3.2)	(1.8,2.5,2.7)	(2.5,3,3.3)
T_i (小时)	0.5	0.25	1	0.75	0.5	0.8	0.5	0.8
r (等级)	1	2	1	1	1	2	2	3

表 2 算例 2 的应急物资需求特征和数量

应急需求点	A1	A2	A3	A4	A5	A6	A7	A8
g_i (吨)	(1.5,2.7,3.5)	(2,2.5,3.2)	(3.2,3.5,3.8)	(2.2,2.8,3.5)	(1.2,1.5,1.9)	(2.5,3,3.2)	(1.8,2.1,2.4)	(2.5,2.8,3)
T_i (小时)	0.5	0.25	1	0.75	0.5	0.8	0.5	0.8
r (等级)	3	3	1	2	3	3	3	2

应急需求点	A9	A10	A11	A12	A13	A14	A15	A16
g_i (吨)	(2.2,2.5,2.7)	(3.1,3.4,3.8)	(1.1,2,2.8)	(3.3,4.1,4.5)	(2.8,3,3.4)	(2.5,3,3.4)	(1.2,1.5,1.8)	(2.1,2.3,2.6)
T_i (小时)	0.5	0.7	0.2	1	0.8	0.75	0.2	0.6
r (等级)	2	3	3	2	3	3	1	3

粒子群算法的参数设置如下: 粒子群的粒子个数为 50 个, 迭代次数 500, $c_1 = c_2 = 1.4$, 惯性权因子 w 采用 Shi^[20] 建议的线性递减权值策略:

$$w^{(t)} = (w_{ini} - w_{end})(T_{max} - t)/T_{max} + w_{end}$$

(19)

式中: T_{max} 为最大迭代次数, w_{ini} 为初始惯性权重, w_{end} 为进化到最大代数时的惯性权重.

典型取值 $w_{ini} = 0.9$, $w_{end} = 0.4$.

最小车辆数的确定:

$$K = \left\lfloor \sum_{i=1}^N (m_i^L/Q) \right\rfloor + 1$$

(20)

式 (20) 中, $\lfloor \cdot \rfloor$ 表示不大于括号内数字的最大整数. 在应急物资调度过程中, 可用车辆越多, 调度时间越短.

4.2 试验结果及分析

4.2.1 算法结果的精确度分析

设计采用的仿真硬件环境为 Pentium IV/2.4GHz/512 RAM, 仿真软件平台为 Windows XP 操作系统, Matlab 6.5 编译软件.

需求量在模糊区间上随机给出, 当目标函数取最小值时, 解得供货量 (卸货量) 的值; 但用精确算法求解时处理随机数比较困难, 为此假设需求量都取三角模糊数中的偏好需求量, 此时需求量为一常数 (否则无法用数学解析方法求解). 用 Matlab 编程直接求出的解析解和用粒子群算法求得的数值解的效果如表 3 所示, 其中粒子群算法求得解为随机运行 30 次的平均值.

表 3 解析解和粒子群算法求得解的比较

所求得解的类型	优化结果	计算时间 (秒)
直接求得的解析解	0.267	28280.5
粒子群算法求得的数值解	0.276	439.1

粒子群算法求得数值解与精确解析解相比误差为 3.4%, 但是所用时间不到其 1/64.

4.2.2 问题规模对算法性能的影响

在种群规模和迭代次数不变的情况下, 目标是最小化行驶时间和最小化偏离偏好需求函数的值. 用粒子群算法分别求解算例 1、算例 2, 所得结果如表 4 所示, 进化曲线如图 5、图 6 所示. 其中优化结果和计算时间为随机运行 30 次的均值.

针对本文研究的问题, 当问题规模是原来的 2 倍时, 用 Matlab 直接求解的时间复杂度是 $O(N!)$, 所用计算时间以阶乘方式增长, 算例 2 的计算时间为: $\frac{21!}{10!} \times 28280.5 \approx 1.1 \times 10^{14}$ (小时), 无法在有限的时间内算出结果. 而用粒子群算法的求解时间仅增加为原来的 4 倍, 是完全可以求解的. 该结论与前面分析得出的粒子群算法时间复杂度为 $O(N^2)$ 是一致的. 同时, 通过算例 1 和算例 2 的进化曲线来看 (图 5 和 6), 粒子群算法的收敛速度和收敛性能是比较理想的.

表 4 用粒子群算法求解算例 1 和算例 2 的比较

算例类型	优化结果	计算时间 (秒)
算例 1	0.367	441.2
算例 2	0.286	1696.5

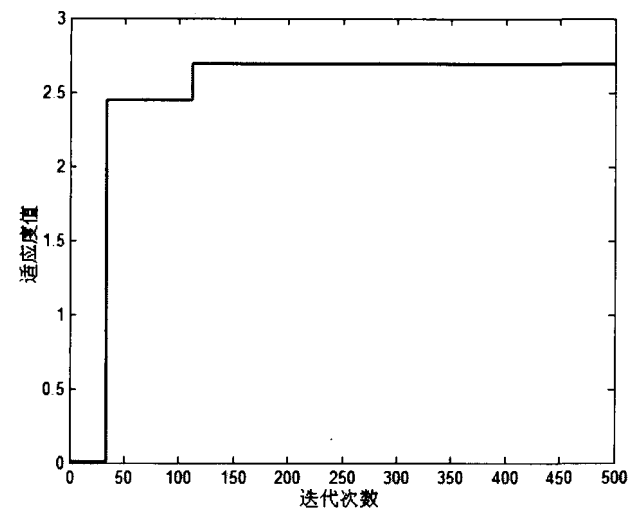


图 5 算例 1 的进化曲线

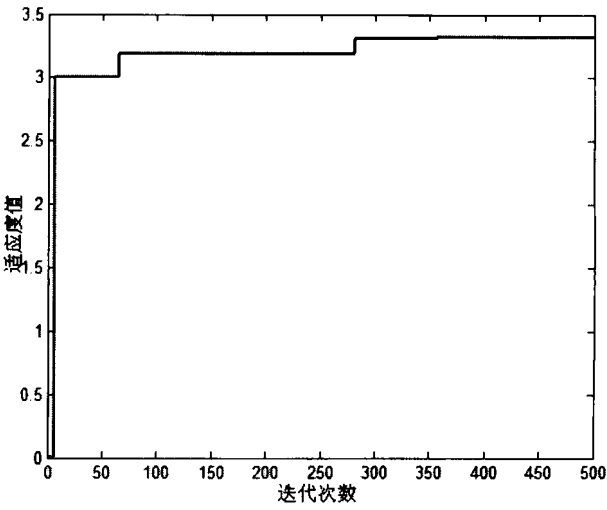


图 6 算例 2 的进化曲线

5 主要结论

论文从模糊需求信息、动态路网和车辆行驶速度变化、以及带有优先级别供应等三个角度刻画应急条件下物资配送调度的动态复杂环境条件, 在借鉴已有研究成果的基础上, 构建了具有更好反映客观实际需要的物资配送动态调度优化模型, 并根据模型本身的特点和现实条件下的强时间压力需求, 设计采用带有群体智能的粒子群算法进行求解, 通过“离散 - 连续向量混合编码”的粒子编码方案和相应的速度位置变更策略, 建立了利用粒子群算法求解的流程和步骤, 然后结合两个算例进行了数值实验与分析, 通过与 Matlab 直接求得精确解的比较, 证明了算法的有效性, 在运算时间和可靠性等方面达到了较好的效果. 但通过实验也发现, 尽管粒子群算法收敛速度快, 但有时容易陷入局部最优解, 尤其当问题规模较大时, 局部解的存在在一定程度上影响到算法质量, 这方面问题的有效解决仍然是将来需要继续探讨的方向之一.

参考文献

[1] Kembull-Cook D, Stephenson R. Lesson in logistics from Somalia[J]. Disaster, 1984, 8(1): 57-66.
[2] Sheu J B. An emergency logistics distribution approach for quick response to urgent relief demand in disasters[J]. Transportation Research Part E: Logistics and Transportation Review, 2005, 43(6): 687-709.
[3] Linet O, Ediz E, Beste K. Emergency logistics planning in natural disasters[J]. Annals of Operations Research, 2004, 129(7): 217-245.

- [4] Meishiang Y, Chen J W. A scenario planning approach for the flood emergency logistics preparation problem under uncertainty[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2007, 43(6): 737–754.
- [5] Barbarosoglu G, Ozdamar L, Cevik A. An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations[J]. *European Journal of Operational Research*, 2002, 140(1): 118–133.
- [6] 何建敏, 刘春林, 尤海燕. 应急系统多出救点的选择 [J]. *系统工程理论与实践*, 2001, 21(11): 19–21.
He J M, Liu C L, You H Y. Selection of multi-depot in emergency systems[J]. *Systems Engineering — Theory & Practice*, 2001, 21(11): 19–21.
- [7] 缪成, 许维胜, 吴启迪. 大规模应急救援物资运输模型的构建与求解 [J]. *系统工程*, 2006, 24(11): 6–12.
Miao C, Xu W S, Wu Q D. A transportation model and solution of large-scale emergency relief commodities[J]. *Systems Engineering*, 2006, 24(11): 6–12.
- [8] Teodorovic D, Radivojevic G. A fuzzy logic approach to dynamic dial-a-ride problem[J]. *Fuzzy Set and Systems*, 2000, 116(1): 23–33.
- [9] Gillett B L. *Introduction to Recent Advances in Vehicle Routing Methods*[M]. Amsterdam: North Holland Publishing, 1984.
- [10] Fox K R, Gavish B, Graves S C. An n -constraint formulation of the time dependent traveling salesman problem[J]. *Operation Research*, 1980, 28(4): 1018–1021.
- [11] Malandraki C, Daskin M S. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms[J]. *Transportation Science*, 1992, 26(3): 185–199.
- [12] Soumia I, Michel G, Potvin J Y. Vehicle dispatching with time-dependent travel times[J]. *European Journal of Operational Research*, 2003, 144(2): 379–396.
- [13] 李宁, 邹彤, 孙德宝. 带时间窗车辆路径问题的粒子群算法 [J]. *系统工程理论与实践*, 2004, 24(4): 130–135.
Li N, Zou T, Sun D B. Particle swarm optimization for vehicle routing problem with time windows[J]. *Systems Engineering — Theory & Practice*, 2004, 24(4): 130–135.
- [14] Kennedy J, Eberhart R C. *Particle swarm optimization*[C]//*Proceeding of IEEE International Conference on Neural Networks, IV*, Piscataway, NJ: IEEE Service Center, 1995: 1942–1948.
- [15] 王凌, 刘波. *微粒群优化与调度算法* [M]. 北京: 清华大学出版社, 2008.
Wang L, Liu B. *Particle Swarm Optimization & Scheduling Algorithms* [M]. Beijing: Tsinghua University Press, 2008.
- [16] Clerc M. *Discrete Particle Swarm Optimization — Illustrated by the Traveling Salesman Problem*[M]. <http://www.mauriceclerc.net>, 2000.
- [17] 刘云霞. *动态车辆调度问题分析及算法设计* [D]. 成都: 西南交通大学, 2004.
Liu Y X. *Dynamic vehicle routing problem analysis and its solving algorithm design*[D]. Chengdu: Southwest Jiaotong University, 2004.
- [18] 唐伟勤, 张隐, 张敏. 大规模突发事件应急物资调度中的车辆路径问题 [J]. *物流技术*, 2008, 27(12): 60–62.
Tang W Q, Zhang Y, Zhang M. Vehicle routing problem of dispatching emergency materials in large-scale emergency events[J]. *Logistics Technology*, 2008, 27(12): 60–62.
- [19] Salmen A, Ahmad I, Al-Madani S. Particle swarm optimization for task assignment problem[J]. *Microprocessors and Microsystems*, 2002, 26(8): 363–371.
- [20] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]// *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, 2000: 1945–1950.