

带硬时间窗模糊车辆路径问题的多目标优化

王连锋, 宋建社, 曹继平, 叶 庆

(第二炮兵工程大学科研部, 西安 710025)

摘 要: 针对带硬时间窗车辆路径问题的多重模糊性, 基于模糊可信性理论建立多目标模糊期望值模型, 提出求解该问题的自适应混合多目标粒子群优化算法。该算法根据相位空间的思想给出一种实数编码方式, 设计双存档机制, 分别存储演化过程中产生的非支配解和有益不可行解, 并引入自适应局部搜索、变异和粒子全局向导选择策略。仿真实验结果表明, 与多目标进化算法相比, 该算法可以获得更优的 Pareto 解集。

关键词: 车辆路径问题; 模糊可信性; 粒子群算法; 多目标优化; 约束; 时间窗; Pareto 最优解

Multi-objective Optimization for Fuzzy Vehicle Routing Problem with Hard Time Windows

WANG Lian-feng, SONG Jian-she, CAO Ji-ping, YE Qing

(Science Research Department, The Second Artillery Engineering University, Xi'an 710025, China)

【Abstract】 Aiming at the vehicle routing problem with hard time windows and multiple fuzzy characteristics, a multi-objective fuzzy expected model is designed based on fuzzy credibility theory, and an adaptive hybrid Multi-objective Particle Swarm Optimization(MOPSO) is proposed to solve the fuzzy vehicle routing model. The algorithm puts forward a particle encoding method according as phase-space, and designs a double archiving mechanism which stores the non-dominated solutions and excellent infeasible solutions separately. It also introduces adaptive strategies on local search, mutation and selection for particle global guide. The comparative experiments with multi-objective evolutionary algorithm verify that the method is capable of getting more excellent Pareto sets.

【Key words】 vehicle routing problem; fuzzy credibility; particle swarm algorithm; multi-objective optimization; constraint; time windows; Pareto optimal solution

DOI: 10.3969/j.issn.1000-3428.2013.04.003

1 概述

车辆路径问题是物流和供应链管理领域的热点问题, 也是经典的组合优化问题, 而随着其应用的推广和研究的深入, 路径、节点和车辆等基本要素表现出了模糊不确定性, 出现了模糊车辆路径问题(Fuzzy Vehicle Routing Problem, FVRP)。文献[1-3]分别对模糊时间窗、模糊需求和模糊旅行时间的情况进行了研究, 但在现实中(如应急物流、军事物流), FVRP 中往往不止存在一种模糊要素, 且考虑的因素也包括车辆数、行驶时间、节点效能、安全性和车辆容量约束等多个。因此, FVRP 往往是带约束和多重模糊性的多目标组合优化问题, 而目前对此类问题的研究较少^[4-8]。

文献[5]综述了多目标车辆路径问题, 指出求解此问题的方法主要有标量法和多目标进化算法(Multiple Objectives Evolutionary Algorithms, MOEA), 尤以后者为重, 因为前者存在权重难以确定的问题。文献[6-8]都以 MOEA 为主, 结合一定局部搜索策略对相应的问题进行求解, 且都采用常用的罚函数法对约束进行处理(拒绝法的实质是罚因子为无穷大), 而罚因子的选取也是一个难题。相比遗传进化算法, 粒子群优化(Particle Swarm Optimization, PSO)算法以其易实现、机制简单、参数少、高效并行搜索等优点被广泛地应用于各类复杂环境中优化问题的求解。文献[9]将 PSO 成功地应用在多目标优化领域, 提出了比较经典的带自适应网格技术的多目标粒子群优化(Multi-objective Particle Swarm Optimization, MOPSO)算法, 文献[10]利用 PSO 求解

基金项目: 国家自然科学基金资助重点项目(61132008)

作者简介: 王连锋(1985—), 男, 博士研究生, 主研方向: 智能计算; 宋建社, 教授、博士生导师; 曹继平, 讲师; 叶 庆, 博士研究生

收稿日期: 2012-07-09 **修回日期:** 2012-08-15 **E-mail:** alfenfeng185@sina.com

了多目标的车辆路径问题,但其对约束的处理仍然采用了罚函数的方法,文献[11]综述了进化计算中约束的处理方法,并指出将约束条件转化为目标函数形成多目标优化问题可以取得较好的效果。

本文在已有文献对模糊车辆路径问题和多目标约束优化问题研究的基础上,针对具有多重模糊性的 FVRP,依据模糊可信性理论建立了模糊期望值模型,并基于 MOPSO 提出了自适应混合多目标粒子群优化(Adaptive Hybrid Multiple Objectives Particle Swarm Optimization, AHMOPSO)算法。

2 带硬时间窗的模糊车辆路径问题

设某一物流中心有 v_0 和 n 个节点 $v_i, i=1,2,\dots,n$, 车辆由 v_0 出发,服务完 n 个节点后返回 v_0 , 节点 v_i 必须在时间窗 $[e_i, l_i]$ 内被服务,且一个节点必须且只能被一辆车服务。已知 v_0 有 K 辆具有相同容量 Q 的运输车, 节点 v_i 的需求量为三角模糊数 $q_i = (q_{i,1}, q_{i,2}, q_{i,3})$ 且允许部分服务。车辆在边 (v_i, v_j) 上的安全系数为 $p(i, j)$, 旅行时间也为三角模糊数 $t(i, j) = (t_1(i, j), t_2(i, j), t_3(i, j))$, 在 v_i 处的服务时间 $\delta_i = \lambda d_i$, λ 为单位物品服务所需时间。令 $R = (r_1, r_2, \dots, r_K)$ 表示车辆路线集, 其中, $r_k = (v_0^k = v_0, v_1^k, v_2^k, \dots, v_{n_k}^k, v_{n_k+1}^k = v_0)$; v_i^k 表示第 k 条路线中第 i 个节点; n_k 为路线 k 中节点数。

上述问题是一类抽象化的 FVRP, 它存在于应急管理、战场资源保障及危险品生产和运输等多种场合, 带有明显的弱经济性, 时间和安全性是此类问题考虑的 2 个重要因素。此时决策人员希望得到一种方案, 使得派出车辆全部返回的期望时间最小化、最危险线路的安全系数最大化, 以便准备下一批次的配送。本文即考虑此种情况下的 FVRP 建模与求解问题。

3 多目标模糊期望值模型的建立

在路线 r_k 中, 假设车辆服务完节点 v_i^k 后, 车辆剩余运载量为 $Q_i^k = Q - \sum_{j=1}^i q_j^k, i=1,2,\dots,n_k$ 。由模糊变量性质可知 Q_i^k 仍是三角模糊数。考虑到车辆运载能力受限和需求不确定性, 车辆在服务途中可能需要返回车场满载后继续服务, 此时记为一次服务失败。车辆在服务节点 v_i^k 时存在 2 种失败的可能性, 即当 $Q_i^k < 0$ 和 $Q_i^k = 0 (i \neq n_k)$ 时, 可能性分别为 $Pos\{Q_i^k < 0\}$ 及 $Pos\{Q_i^k = 0\}$ 。

3.1 车辆期望行驶时间计算

令 s_i^k 表示车辆 k 从节点 v_i^k 到下一节点 v_{i+1}^k 所行驶的时间, 由分析可知: 若 $0 < Q_{i-1}^k < q_i^k, s_i^k = 2t^k(i, 0) + t^k(i, i+1)$, 且 $Q_i^k = Q - q_i^k$; 若 $Q_{i-1}^k = q_i^k, s_i^k = t^k(i, 0) +$

$t^k(0, i+1)$, 且 $Q_i^k = Q$; 若 $Q_{i-1}^k > q_i^k, s_i^k = t^k(i, i+1)$, 且 $Q_i^k = Q_{i-1}^k - q_i^k$ 。

由此, 定义离散模糊变量 $s_i^k, i=1,2,\dots,n_k$:

$$\mu(s_i^k) = \begin{cases} pos(Q_i^k < 0), s_i^k = 2t^k(i, 0) + t^k(i, i+1) \\ pos(Q_i^k = 0), s_i^k = t^k(i, 0) + t^k(0, i+1) \\ pos(Q_i^k > 0), s_i^k = t^k(i, i+1) \end{cases} \quad (1)$$

其中, $t^k(i, j)$ 表示节点 v_i^k 到 v_j^k 之间所需的旅行时间, 由于 $t^k(i, j)$ 为模糊变量, 因此 s_i^k 为双重模糊变量。令 $\xi = 2t^k(i, 0) + t^k(i, i+1), \eta = t^k(i, 0) + t^k(0, i+1), \varsigma = t^k(i, i+1), \xi, \eta$ 和 ς 同为可能性空间 $(\Theta, P(\theta))$ 上的模糊变量, 根据文献[12]可知, 当且仅当 $E(\xi) > E(\eta)$ 时有 $\xi > \eta$, 其中, E 为期望值算子。据此, 可设 $s_{i,1}^k = \min\{\xi, \eta, \varsigma\}, s_{i,3}^k = \max\{\xi, \eta, \varsigma\}, s_{i,2}^k = \text{mid}\{\xi, \eta, \varsigma\}$, 隶属度分别为 $\mu_{i,1}^k, \mu_{i,2}^k, \mu_{i,3}^k$, 其中, mid 表示取中间值算子; $\mu(s_i^k)$ 可根据三角模糊变量的特征计算出。则由双重模糊变量的期望值定义^[12]可知:

$$\begin{aligned} \omega_{i,1}^k &= \frac{1}{2} \left(\mu_{i,1}^k + \max_{1 \leq j \leq 3} \mu_{i,j}^k - \max_{1 < j \leq 3} \mu_{i,j}^k \right) \\ \omega_{i,2}^k &= \frac{1}{2} \left(\max_{1 \leq j \leq 2} \mu_{i,j}^k - \max_{1 \leq j < 2} \mu_{i,j}^k + \max_{2 \leq j \leq 3} \mu_{i,j}^k - \max_{2 < j \leq 3} \mu_{i,j}^k \right) \\ \omega_{i,3}^k &= \frac{1}{2} \left(\max_{1 \leq j \leq 3} \mu_{i,j}^k - \max_{1 \leq j < 3} \mu_{i,j}^k + \mu_{i,3}^k \right) \end{aligned}$$

综上, 车辆 k 的期望值行驶时间为:

$$E(r_k) = \sum_{i=0}^{n_k} \sum_{j=1}^3 \omega_{i,j}^k E(s_{i,j}^k) \quad (2)$$

3.2 车辆期望安全通过率计算

与分析车辆的期望行驶时间原理一样, 令 p_i^k 为车辆 k 从节点 v_i^k 可到达下一节点 v_{i+1}^k 的安全系数, 根据前文分析可知它也为一个离散模糊变量, 且有:

$$\mu(p_i^k) = \begin{cases} pos(Q_i^k < 0), p_i^k = [p^k(i, 0)]^2 p^k(i, i+1) \\ pos(Q_i^k = 0), p_i^k = p^k(i, 0) p^k(0, i+1) \\ pos(Q_i^k > 0), p_i^k = p^k(i, i+1) \end{cases} \quad (3)$$

参照 3.1 节的方法可以求出 $p_{i,j}^k$ 和 $\omega_{i,j}^k, j=1,2,3$, 则车辆 k 的安全系数的期望值为:

$$E(P^k) = \prod_{i=0}^{n_k} \sum_{j=1}^3 \omega_{i,j}^k p_{i,j}^k \quad (4)$$

3.3 模型建立

设 $x_{ijk} = 1$ 表示车辆 k 计划从 v_i 沿边 (v_i, v_j) 到 v_j , 否则, $x_{ijk} = 0$; s_{ijk} 表示车辆 k 从 v_i 行驶到 v_j 所需时间; t_i 表示车辆到达 v_i 的时间(若存在服务失败, 则表示

第2次到达时间), w_i 表示车辆在 v_i 处的等待时间, $w_i = \max(e_i - t_i, 0)$ 。则 FVRP 的多目标模糊期望值模型为:

$$\min f_1 = \max_k \left[E(r_k) + \sum_{i=0}^n \sum_{j=0}^n x_{ijk} E(w_j + \delta_j) \right] \quad (5)$$

$$\max f_2 = \min_k E(P^k) \quad (6)$$

$$\text{s.t.} \quad \sum_{j=1}^n q_{j3} \sum_{i=0}^n x_{ijk} \leq 2Q, \forall k \quad (7)$$

$$E(x_{ijk}(t_i + w_i + \delta_i + s_{ijk} - t_j)) \leq 0, \forall i, j, k \quad (8)$$

$$e_j \leq E \left[\sum_{k=1}^K \sum_{i=0}^n x_{ijk} (t_j + w_j) \right] \quad l_j - E\delta_j, \forall j \quad (9)$$

$$\sum_{k=1}^K \sum_{j=1}^n x_{ijk} \leq K, i=0 \quad (10)$$

$$\sum_{i=1}^n x_{i0k} = \sum_{j=1}^n x_{0jk} \leq 1, \forall k \quad (11)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1, \forall i \quad (12)$$

$$\sum_{k=1}^K \sum_{i=0}^n x_{ijk} = 1, \forall j \quad (13)$$

$$t_0 = w_0 = \delta_0 = 0 \quad (14)$$

$$x_{ijk} = 0 \text{ or } 1, \forall i, j, k \quad (15)$$

其中, 式(5)和式(6)为目标函数, 分别为最小化车辆全部返回的期望时间及最大化最危险线路的安全系数; 式(7)保证了每辆车在服务途中最多返回中心一次; 式(8)和式(9)为时间窗约束; 式(10)为总的使用车辆数约束; 式(11)使得车辆从中心出发并最终返回中心; 式(12)和式(13)保证了每个节点仅有一辆车服务; 式(14)和式(15)声明了变量。

考虑到对数函数的单调性, 此处对式(6)进行对数变换为 $\ln(E(P^k))$, 则式(6)变化为:

$$\min f_2 = -\min_k [\ln E(P^k)] \quad (16)$$

这样放大了不同路径方案在该目标上的差异性, 也便于与式(5)一起进行 Pareto 比较。

4 自适应混合多目标粒子群算法

本文采用粒子群算法和变邻域启发式算法的混合求解策略, 先对路线进行实数编码, 再基于相位空间的思想对粒子进行解码, 采用双存档机制存储非支配解和有益不可行解, 并设计了自适应的局部搜索、变异和向导选择策略。

4.1 粒子编码与更新

第 j 个粒子记为 $X^j = (X_1^j, X_2^j, \dots, X_n^j)$, 其中 $i=1, 2, \dots, n$, $j=1, 2, \dots, M$, M 为群体规模, $X_i^j \in [0, \Theta)$, $\Theta > 0$ 。

若 $TL_c \leq X_i^j < TL_{c+1}$, 则节点 v_i 将由第 c 辆车进行服务。其中:

$$TL_c = \Theta(c-1)/K \quad (17)$$

同理, 粒子的速度 $V^j = (V_1^j, V_2^j, \dots, V_n^j)$, 其中, $V_i^j \in [-\Theta, \Theta)$ 。如果 $X_i^j \notin [0, \Theta)$ 或 $V_i^j \notin [-\Theta, \Theta)$, 则有:

$$X_i^j = \text{mod}(X_i^j, \Theta) \quad (18)$$

$$V_i^j = \text{mod}(V_i^j, 2\Theta) - \Theta \quad (19)$$

其中, $\text{mod}(x, y)$ 表示将 x 对 y 进行取余运算, 如 $\text{mod}(-5, 20) = 15$, $\text{mod}(25, 20) = 5$ 。同一线路内的服务顺序根据 x_i^j 由小到大进行排列。

粒子更新公式采用基本的迭代公式^[9], 惯性权重自适应线性递减 $w(t) = 0.9 - 0.5t/T$, t 为当前迭代次数, T 为算法最大迭代次数。

4.2 自适应向导选择和约束处理

根据式(7)~式(10), 可计算出每辆车违反容量、时间窗及车辆数的程度, 记三者之和为该车辆总的不满足约束的程度 $f'(r_k)$, 令粒子 X 的不可行度记为 $f_3(X) = \sum_{k=1}^K f'(r_k)$ 。基于此, 本文采用如下约束处理^[4]及向导选择方式:

(1) 将粒子群 \mathcal{N} 分为可行解集 $\mathcal{N} = \{X | f_3(X) = 0\}$ 与不可行解集 $\bar{\mathcal{N}} = \{X | f_3(X) > 0\}$ 。

(2) 通过 Pareto 占优的概念把可行解集 \mathcal{N} 分为非支配解集 P_s 和支配解 \bar{P}_s 。将 P_s 存储在外部非支配解档案 \mathbb{R} 中, \mathbb{R} 的更新采用自适应栅格技术^[9], 但不设上限。

(3) 将不可行解集 $\bar{\mathcal{N}}$ 分为有益解集 Q_b 和非有益解集 \bar{Q}_b 。将 Q_b 存储在规模为 \bar{N} 的有益非可行解档案 \mathbb{Z} 中, 若 \mathbb{Z} 中个体数超过了 \bar{N} , 则根据 $f_3(X)$ 由小到大的顺序, 保留前 \bar{N} 位个体。 Q_b 中的个体 X 应当不被 P_s 中任一位个体支配。

(4) 对每一个粒子, 产生随机数 $rand$, 若 $rand < N_0/(N_0 + \bar{N})$, 根据文献[9]的方法从 \mathbb{R} 中选择全局向导; 否则, 从 \mathbb{Z} 中随机选择全局向导。为加快收敛速度, 逐渐增加可行解参与进化的比例, \bar{N} 随迭代次数进行自适应调整, 满足 $\bar{N}(t) = N_0 - tN_0/T$ 。个体最优向导的更新为: 若新的个体支配个体最优向导, 则个体最优向导更新为新的个体; 若两者互不支配, 则随机个体最优向导随机选择两者之一; 其他情况则保持不变。

4.3 自适应局部搜索策略

局部搜索策略对强约束优化问题的求解质量有较大的作用^[13]。考虑到本文研究问题中 Pareto 解的数量往往没有连续问题那样多, 因此, 寻找更多的非支配解便成为了算法的主要任务。基于上述思想, 本文根据 Pareto 解的数量变化情况进行自适应局部搜索, 以便获得更多的非支配解。当 \mathbb{R} 中 Pareto 解的数量 n_{pa} 连续 M' 代没有变化时, 即对当前的非支配解集进行局部搜索。

局部搜索主要采用 2opt、Split^[7]和 Iterated Greedy^[14] 3 类算子。具体步骤如下:

(1)对 \mathbb{R} 中每一非支配解,判断其中路径的条数 κ , 若 $\kappa < K$, 转步骤(2); 否则, 转步骤(3)。

(2)产生随机数 $rand$, 若 $rand < 0.5$, 则选择目标 1(f_1) 最大的线路进行 Split 操作, 并对所有线路进行 2opt 优化; 否则, 选择目标 2(f_2) 最大的线路进行 Split 操作, 并对所有线路进行 2opt 优化。

(3)产生随机数 $rand$, 若 $rand < 0.5$, 对所有线路执行 2opt 优化; 反之, 转步骤(4)。

(4)进行 Iterated Greedy 操作。

在 2opt 算子中, 停止规则为发现支配原解或与原解互不支配的可行解; 在 Iterated Greedy 算子中, 取出 u 个节点, $u \in [\lceil 0.1n \rceil, \lceil 0.2n \rceil]$, 再按 F'_k 值由小到大的顺序对路径排列, 形成新路径集 $R' = \{r'_k, k=1, 2, \dots, K'\}$, $K' \leq K$, $F'_k = f_1(r'_k) \times f_2(r'_k)$, 最后将之前取出的单元可行地回插到 R' 中^[15]。

4.4 自适应变异

引入变异算子可增强粒子的多样性, 是解决粒子群算法易早熟的主要方法之一。研究发现高斯变异算子不仅能有效避免算法早熟收敛, 且具有一定的局部搜索能力, 有利于找到更好的非支配解^[15], 因此, 本文采用高斯变异算子。

过大的变异概率 P_{mu} 使得算法难以收敛, 而过小的 P_{mu} 又不足以维持种群的多样性。此处引入文献[8]中多样性的度量 ψ , 提出一种自适应变异策略。

在一般情况下, P_{mu} 在算法初期取得较大的值, 以便于找到更多的非劣解, 算法后期取较小值, 有利于算法的收敛。而算法初期, 由于随机初始化大部分粒子, ψ 较大, 在算法后期, 所有粒子都比较接近 Pareto 前沿, ψ 较小。因此, 可直接将 ψ 作为 P_{mu} 使用。

4.5 AHMOPSO 算法流程

综合上述对算法关键环节的描述, 本文设计的自适应混合多目标粒子群算法的流程如下:

步骤 1 设种群规模为 N , 随机产生所有粒子的速度及 $0.8N$ 个粒子的位置, 剩余 $0.2N$ 利用插入法产生。在插入过程中, 每一条新路径的起点及待插入点都是从剩余未插入点集中随机产生的, 以此尽可能地提高可行粒子的多样性。

步骤 2 计算粒子的适应度值, 初始化个体向导 $pb_i(t)$, 按照 4.2 节的方法初始化 \mathbb{R} 、 \mathbb{Z} 、 N_0 和 \bar{N} , 令迭代次数 $t=0$, n_{pa} 连续未变化代数 $t_{pa}=0$ 。

步骤 3 令 $t:=t+1$, 开始循环迭代。

步骤 4 更新惯性权重, 根据 4.2 节选择全局向导 $gb_i(t)$, 更新粒子速度与位置, 按 P_{mu} 变异粒子。

步骤 5 更新 $pb_i(t)$, 若 $i < n$, 转步骤 4。

步骤 6 确定 P_s 和 Q_b , 并更新 \mathbb{R} 和 \mathbb{Z} 。

步骤 7 更新 t_{pa} , 若 $t_{pa} > M'$, 对 \mathbb{R} 进行局部搜索并更新 \mathbb{R} 与 \mathbb{Z} 。

步骤 8 计算变异概率 P_{mu} , 更新 \bar{N} 。

步骤 9 若 $t < T$, 返回步骤 3; 否则, 算法终止, 输出 \mathbb{R} 。

4.6 算法分析

从算法的描述中可以看出, AHMOPSO 本质上是并行地运行在 2 个群体(种群 \mathbb{S} 与 \mathbb{R})上的随机演化过程, 这 2 个演化过程既相互独立, 又相互关联, 充分利用了可行域与不可行域 2 个部分解集。同时, 基于相位空间思想的粒子解码方式也在一定程度上降低了粒子的收敛速度, 自适应局部搜索及变异算子的引入可以在一定程度上使算法跳出局部最优, 在保证收敛的基础上, 也提高解的多样性, 较好地平衡了收敛的速度和精度。

5 仿真实验及结果分析

对本文中所讨论的问题目前还没有标准测试集, 这里通过改造 Solomon 算例获得实验数据。对每个 benchmark, 若车辆容量为 Q' , 节点 v_i 需求量为 d_i , 时间窗为 $[e'_i, l'_i]$, 到节点 v_j 之间的距离为 $D(i, j)$, 其最优解对应的车辆数为 K_0 (如已知 C101 算例最优解的车辆数为 10), 则本文算例设定车辆数 $K = K_0 + 1$, 节点 v_i 的需求量为三角模糊数 $q_i = (d_i(1-0.5rand), d_i, d_i(1+0.5rand))$, 服务时间为 $\delta_i = 3q_i$, 时间窗 $e_i = e'_i$, $l_i = l'_i + \delta_i$, 车辆容量 $Q = 0.75Q'$ (由 q_i 及式(7)确定), 单元 v_i 与 v_j 之间行驶时间为三角模糊数 $t(i, j)$, 安全系数为 $[0.2, 1]$ 之间的随机数, 其中, $t_1(i, j) = D(i, j)(1-0.5rand)$; $t_2(i, j) = D(i, j)$; $t_3(i, j) = D(i, j)(1+0.5rand)$; $rand$ 为 $[0, 1]$ 之间的随机数。算法种群规模为 100, 迭代次数 $T=100$, \mathbb{R} 的规模 $N_0=100$, 最大允许未改变次数 $M'=10$, \mathbb{S} 的规模 $N=100$, 每一个目标函数在 \mathbb{R} 中栅格数 $ngrid=10$, 栅格膨胀系数 $\alpha=0.1$, 全局向导选择压力系数 $\beta=10$, 删除压力系数^[9] $\gamma=2$, 惯性权重 $\omega=0.7298$, 学习因子 $c_1=c_2=2.05$, 高斯变异采用标准正态分布。算法利用 Matlab2010b 编程, 并在 2.33 GHz/3.25 GB 的计算机上运行。

为说明算法中自适应搜索的有效性, 修改 C101 算例, 记为 C101'算例, 将带自适应搜索(Adaptive Search Method, ASM)的 AHMOPSO 中局部搜索部分(包括自适应局部搜索、变异和向导选择, 全局最优向导只从 \mathbb{R} 选取)去掉(No Adaptive Search Method, NASM), 即 MOPSO, 并运算一次, 记录 \mathbb{R} 中每代非支配解集的 f_1 和 f_2 的平均值, 与 AHMOPSO 比较, 如图 1 和图 2 所示。从两图的总体趋势可以看出, 自适应搜索环节可显著提高算法的收敛精度,

其中,ASM 曲线的尖峰部分也直观说明了自适应局部搜索对扩大解的范围的影响。

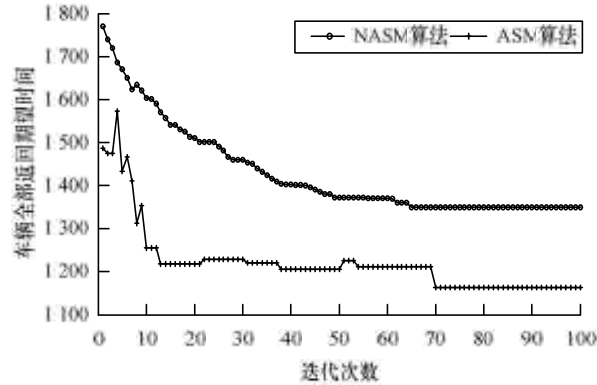


图1 f_1 的平均值与迭代次数的关系

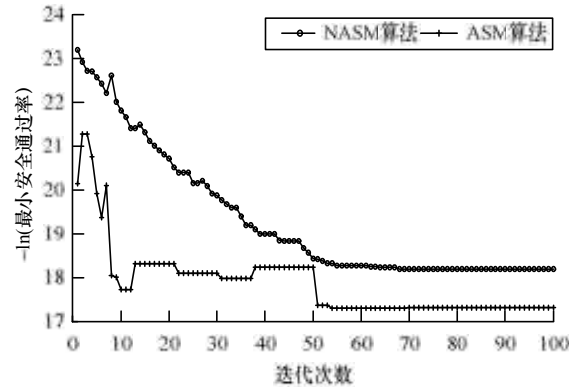


图2 f_2 的平均值与迭代次数的关系

为说明 AHMOPSO 的有效性, 将其与文献[6]提出的多目标进化算法 MOEA 相比, 种群与迭代次数都为 100, 交叉算子为最佳成本交叉算子^[6], 交叉概率为 0.7, 变异算子为 Partial swap 算子^[7], 变异概率为 0.4。对选取的部分 Benchmark 算例, 问题规模都为 100, 将 2 种算法分别运算 10 次, 统计结果如表 1 所示。

表1 AHMOPSO 算法与 MOEA 算法的仿真结果比较

算例	MOEA 算法				AHMOPSO 算法			
	n	opt	Δ	time	n	opt	Δ	time
C101'	17	4.952 8	0.884 4	175.121 4	23	3.983 4	0.813 4	204.125 6
C102'	18	9.054 6	0.888 8	146.238 4	24	5.788 7	0.845 9	131.842 9
C203'	26	7.378 9	0.853 5	284.621 7	39	6.896 4	0.780 2	339.674 2
R105'	6	24.479 2	0.894 1	526.234 2	11	6.055 6	0.823 6	630.395 1
R106'	21	13.428 1	0.875 3	479.245 4	30	14.036 3	0.807 2	556.135 3
R208'	16	16.734 8	0.854 6	703.579 2	28	7.488 1	0.787 8	832.596 1
RC101'	9	23.930 0	0.476 2	462.970 5	8	11.643 1	0.485 1	459.485 6
RC203'	24	25.770 7	0.793 4	579.790 0	37	14.452 4	0.691 2	702.359 4

在表 1 中, n 表示 Pareto 解的个数; opt 为相对最短距离指标, 该指标类似于逼近性指标, 因为没有真实的 pareto 前沿, 所以这里采用此指标, 详见文献[10], 一般为越小越好; Δ 为 Pareto 解的覆盖度^[9], 为越大越好型; $time$

为运算时间。从中可以看出, AHMOPSO 虽然在大部分算例上比 MOEA 计算时间要长, 但它获得 Pareto 解的质量还是明显要高一些。

为更直观地表现上述结论, 以 C101' 为例, 给出了 2 种算法所得 Pareto 前沿的比较, 如图 3 所示。

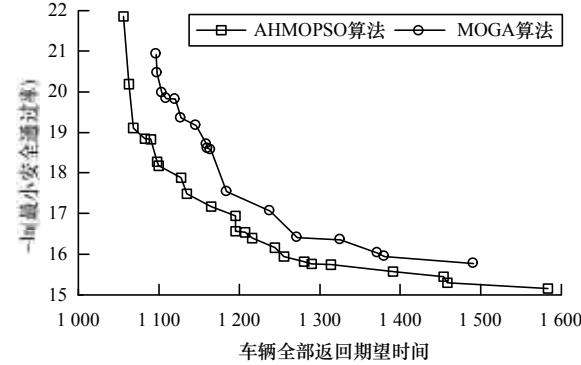


图3 C101'算例中的 Pareto 前沿比较

6 结束语

本文以带硬时间窗的模糊车辆路径问题为研究对象, 利用模糊可信性理论建立了多目标模糊期望值模型, 并设计了求解该模型的自适应混合多目标粒子群算法。首先基于相位空间思想给出一种实数编码方式, 然后引入自适应局部搜索与变异策略, 最后为增强算法在可行与不可行解边界的搜索能力, 设计了自适应双存档机制, 仿真实验表明了该算法可以有效求解所建立的多目标模糊期望值模型。今后的工作将主要围绕带软时间窗的多车场多目标模糊车辆路径问题展开, 重点研究高效约束处理技巧、快速非支配解集构造方法等。

参考文献

[1] Tang Jiafu, Pan Zhengdong, Fung R Y K, et al. Vehicle Routing Problem with Fuzzy Time Windows[J]. Fuzzy Sets and Systems, 2009, 160(5): 683-695.

[2] Cao Erbao, Lai Mingyong. The Open Vehicle Routing Problem with Fuzzy Demands[J]. Expert Systems with Applications, 2010, 37(3): 2405-2411.

[3] 张建勇, 李 军. 具有模糊旅行时间的 VRP 的一种混合遗传算法[J]. 管理工程学报, 2006, 20(4): 13-16.

[4] 焦李成, 尚荣华, 马文萍, 等. 多目标优化免疫算法、理论和应用[M]. 北京: 科学出版社, 2010.

[5] Jozefowicz N, Semet F, Talbi E G. Multi-objective Vehicle Routing Problems[J]. European Journal of Operational Research, 2008, 189(2): 293-309.

[6] Ombuki B, Ross B, Hanshar F. Multi-objective Genetic Algorithm for Vehicle Routing Problem with Time Windows[J]. Applied Intelligence, 2006, 24(1): 17-30.

(下转第 17 页)