

局部搜索与改进 MOPSO 的混合优化算法及其应用

王丽萍^{1,2} 吴秋花¹ 邱飞岳^{3,4} 吴裕市⁴

(浙江工业大学经贸管理学院 杭州 310023)¹ (浙江工业大学智能信息处理研究所 杭州 310023)²
(浙江工业大学现代教育技术研究所 杭州 310014)³ (浙江工业大学信息工程学院 杭州 310023)⁴

摘 要 为弥补粒子群后期收敛缓慢与早熟的不足,提出了一种局部搜索与改进 MOPSO 的混合优化算法(H-MOPSO)。该算法首先采用非均匀变异算子和自适应惯性权重,强化全局搜索能力;继而建立混合算法模型,并利用侧步爬山搜索算法对粒子群作周期性优化,使远离前沿的粒子朝下降方向搜索,而靠近前沿的粒子朝非支配方向搜索,加快粒子群的收敛并改善解集多样性。对标准测试函数的求解表明,该算法比 MOPSO、NSGA-II 和 MOEA/D 具有更好的多样性和收敛性。供应商优选问题的求解进一步验证了 H-MOPSO 的有效性。

关键词 多目标优化,粒子群算法,局部搜索,混合算法

中图分类号 TP18 文献标识码 A

Hybrid Optimized Algorithm Based on Improved MOPSO and Local Search and its Application

WANG Li-ping^{1,2} WU Qiu-hua¹ QIU Fei-yue^{3,4} WU Yu-shi⁴

(College of Business Administration, Zhejiang University of Technology, Hangzhou 310023, China)¹

(Institute of Intelligence Information Processing, Zhejiang University of Technology, Hangzhou 310023, China)²

(Institute of Educational Technology, Zhejiang University of Technology, Hangzhou 310014, China)³

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)⁴

Abstract In order to improve the weaknesses of the particle swarm's easily premature and slow convergence in late stage, the H-MOPSO, based on the integration of improved MOPSO and local search, was proposed. First of all, the non-uniform mutation operator and self-adaptive inertia weight were adopted to enhance its ability of global search. Then, the model of MOPSO hybrid with local search was established. According to the model, the local search algorithm based on hill climbing strategy with sidesteps was periodically used to optimize the swarm, making particles search along descent direction when they were away from Pareto front, and search along non-dominated direction while they were near Pareto front. Simulation results of benchmark functions show that H-MOPSO has better performance compared with MOPSO, NSGA-II and MOEA/D. The solving of supplier selection problem further validates its effectiveness.

Keywords Multi-objective optimization, Particle swarm optimization, Local search, Hybrid algorithm

1 引言

科学研究及工程实践中存在着大量多目标优化问题,该类问题通常不存在多目标均为全局最优的解,而存在一个非劣解集。一般地,多目标优化问题的形式可表示为^[1]:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \dots, f_r(x)) \\ \text{s. t. } &\begin{cases} g_k(x) \geq 0, & k=1, 2, \dots, p \\ h_l(x) = 0, & l=1, 2, \dots, q \\ x_m \in [x_m^L, x_m^U], & m=1, 2, \dots, n \end{cases} \end{aligned} \quad (1)$$

式中, $f_i(x)$ ($i=1, \dots, r$) 为目标函数, $g_k(x) \geq 0$ ($k=1, \dots, p$) 和 $h_l(x) = 0$ ($l=1, \dots, q$) 为目标函数所满足的等式与不等式约束。对于该模型,称解 u 为 Pareto 支配解 v , 当且仅当 $\forall i$

$\in \{1, \dots, r\}, f_i(u) \leq f_i(v)$, 且 $\exists i \in \{1, \dots, r\}, f_i(u) < f_i(v)$, 记为 $u \succ v$ 。称解集空间 X 中的任意向量 u 为 Pareto 最优解, 当且仅当 $\neg \exists v \in X$ 使得 $v \succ u$ 。 X 中所有 u 的集合 P 称为 Pareto 最优解集, 对应 $F(P)$ 即 Pareto 最优前沿 (Pareto Front)。

多目标进化算法 (Multi-Objective Evolutionary Algorithm, MOEA) 已被证明是求解此类问题的有效方法, 其中多目标粒子群算法 (Multi-Objective Particle Swarm Optimization, MOPSO)^[2] 相对于其他 MOEA, 如 NSGA-II^[3], 因计算简便、参数设置方便、收敛速度快而备受关注^[4,5]。

然而, 粒子群的搜索依赖于最优粒子, 因此容易导致算法后期收敛缓慢甚至陷入局部最优。为提高算法的性能, 使其

到稿日期: 2012-01-05 返修日期: 2012-04-21 本文受国家自然科学基金项目 (61070135), 国家社会科学基金项目 (10GBL095) 资助。

王丽萍 (1964—), 女, 教授, 博士生导师, 主要研究方向为决策优化、计算智能, E-mail: qfywlp@gmail.com; 吴秋花 (1987—), 女, 硕士生, 主要研究方向为商务智能; 邱飞岳 (1965—), 男, 教授, 硕士生导师, 主要研究方向为学习科学与媒体技术、智能系统; 吴裕市 (1987—), 男, 硕士生, 主要研究方向为人工智能。

能更好地解决实际问题,学者们提出引入局部搜索算法优化最优粒子,指导粒子群快速寻找均匀分布的 Pareto 最优解集。作为高效求解单目标问题的方法,序列二次规划、牛顿法等自然而然地被选作局部搜索策略^[6,7]。然而,此类基于梯度的方法要求目标函数严格连续,限制了其应用范围。为此,学者们提出采用爬山法^[8]、禁忌搜索^[9]、混沌搜索^[10]等启发式方法作为局部搜索策略。这些混合 MOPSO 的提出及对各种测试函数和实际问题的成功求解表明,局部搜索算法对粒子群搜索具有积极引导作用。但以上采用的局部搜索算法,仅考虑对 MOPSO 的收敛性或多样性作单方面改进,这容易引起另一方面性能的退化,如文献^[6,7]侧重于深度搜索,易陷入局部最优,文献^[8-10]着重于保持种群多样性,牺牲了算法的收敛性。

鉴于此,本文引入侧步爬山策略,提出一种基于侧步爬山策略的局部搜索算法与改进 MOPSO 的混合优化算法。该算法首先采用非均匀变异算子和自适应惯性权重策略,来强化全局搜索能力。并依据建立的局部搜索与粒子群优化混合模型,采用侧步爬山策略对粒子群做周期性优化,使其根据与前沿的距离朝下降方向或非支配方向搜索,加快收敛并改善其分布。通过对标准测试函数及供应商优选问题进行求解,证明了 H-MOPSO 的有效性。

2 局部搜索与改进 MOPSO 的混合优化算法

2.1 局部搜索算法

本文局部搜索算法采用了新颖的侧步爬山策略^[11],这是一种基于 Pareto 支配关系、依据解离前沿的远近构造下降方向或非支配方向的局部搜索策略。图 1 以二目标为例,描述了该局部搜索策略的原理。

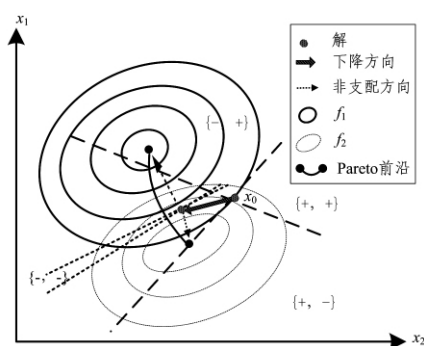


图 1 局部搜索策略原理

图 1 中, $\{-, -\}$ 表示粒子 x_0 的下降锥(descend cone),即该区域中任意一点 x 有关系式 $x \succ x_0$; $\{+, +\}$ 表示 x_0 的上升锥(ascend cone),即该区域中的点 x 均被 x_0 支配;当 x 处于区域 $\{-, +\}$ 中时,有 $f_1(x) < f_1(x_0)$ 且 $f_2(x) > f_2(x_0)$,称该区域为多样锥(diversity cone),反之同理。对于三维及以上情况,区间划分以此类推,除了下降锥 $\{-, \dots, -\}$ 与上升锥 $\{+, \dots, +\}$,其余均为多样锥。

采用该策略对粒子作局部搜索时:①初期粒子群普遍远离前沿,下降锥与上升锥占空间近一半,容易在下降锥中找到最优支配解,因此可利用下降方向,引导粒子群快速向前沿靠拢。为使算法可求解更普遍的实际问题,下降方向采用非梯度法计算,即在 $\{-, -\}$ 中 x_0 的下降方向 $d = x - x_0$;若在

$\{+, +\}$ 区域,则 $d = x_0 - x$ 。②若在有限 M_b 次局部搜索内搜索得到的均为非支配解,说明粒子群已非常接近前沿,即搜索过程的末期,多样锥远远大于下降锥及上升锥。此时转变搜索方向,以多样化为目标引导粒子群在前沿上的扩散,见图 1

中的非支配方向,表达式为: $\vec{d}_s = \sum_{i=1}^{M_b} ((x_i - x_0) / \|x_i - x_0\|)$,其中 $x_i (i=1, \dots, M_b)$ 为搜得的各个非支配解。③依据领域内搜索得到的非支配解数与 N_b 的关系,判断粒子与前沿的距离并自适应选择搜索方向:若非支配解数小于 N_b ,则按照①处理;若大于或等于 N_b ,则按照②处理。

基于上述思想,局部搜索算法具体流程如下:

步骤 1 令初始解 y_0 为粒子群中个体 x_0 ,初始化迭代次数 $i=0, \text{flag}=0$ 。

步骤 2 在 y_0 的 r 领域 $[y_0 - r, y_0 + r]$ 内,随机选取解 $y_1, i++$ 。

步骤 3 若 $y_1 \succ y_0$ (或 $y_0 \succ y_1$),令下降方向 $d = y_1 - y_0$ (或 $y_1 - y_0$),采用三点二次插值法计算最优步长 s ,并令 $x_{\text{new}} = y_0 + d \cdot s$ (或 $y_1 + d \cdot s$),转步骤 5;否则转步骤 4。

步骤 4 $\text{flag}++$,累计侧步方向 $\vec{d}_s += (y_1 - y_0) / \|y_1 - y_0\|$ 。若 $\text{flag} \geq N_b$,新解 $x_{\text{new}} = y_0 + \vec{d}_s$,转步骤 5;否则转步骤 2。

步骤 5 若 $i \leq M_b$,回到步骤 2;否则,输出 x_{new} 替换旧的粒子 x_0 。

2.2 MOPSO 及其改进

在多目标粒子群算法中,粒子群的更新公式为:

$$\begin{cases} v_i^{t+1} = w \cdot v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (p_g^t - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{cases} \quad (2)$$

式中, w 为惯性权重, c_1 和 c_2 为学习因子, r_1 和 r_2 为取值 $(0, 1)$ 的随机数。搜索过程中,第 t 代粒子群中的每个粒子根据前面得到的全局最优位置 P_g^t 和个人最优位置 P_i^t ,在学习因子的影响下调节方向与速度,得到新的速度向量 v_i^{t+1} ,并基于该速度向量与当前位置 x_i^t 得到下一位置 x_i^{t+1} 。粒子群搜索前沿解的过程正如一个鸟群搜索食物的过程。

2.2.1 自适应惯性权重

由更新公式(2)可知,惯性权重的大小影响着粒子的“飞行”速度与方向,从而影响算法的全局性与收敛性。Eberhart 等^[12]指出,当惯性权重小于 0.4 时,粒子群的收敛速度快,但易陷入局部最优;当惯性权重大时,如 $w \in [0.9, 1.2]$,全局搜索能力突出,但找到最优粒子需要的迭代次数增加,收敛速度减慢。鉴于此,本文中 w 采用从初始的 0.9 逐代递减至 0.4 的方式,见式(3),使粒子群起初着重全局搜索,得到均匀分布的粒子群后加速收敛。

$$w_t = (w_1 - w_2) \cdot \frac{M_t - t}{M_t} + w_2 \quad (3)$$

式中, t 为当前代数, M_t 为最大迭代次数, w_1 取 0.9, w_2 取 0.4。

2.2.2 非均匀变异算子

为避免粒子群陷入局部最优,引入非均匀变异算子^[13]作小概率 P_{mut} 扰动。设第 t 代有一个粒子 $a^t = (v_1, v_2, \dots, v_m)$,随机选中变量 v_k 进行变异,得到下一代粒子 $a^{t+1} = (v_1, \dots, v_k', \dots, v_m)$, v_k' 的非均匀变异形式见式(4)。

$$v_k' = \begin{cases} v_k + \Delta(t, UB - v_k), & \text{if } flip = 0 \\ v_k - \Delta(t, v_k - LB), & \text{if } flip = 1 \end{cases} \quad (4)$$

$$\Delta(t, y) = y \cdot (1 - r^{(1 - \frac{t}{T})^b})$$

式中, $flip$ 是取值为 0 或 1 的随机数, UB 和 LB 为 v_k 的上界与下界, $\Delta(t, y)$ 返回 $[0, y]$ 之间的随机值, r 服从 $U(0, 1)$, t 和 T 分别表示当前代数与最大代数, b 反映了代数对值的影响程度(文献[13]中取值为 5)。从式(4)可以看出, 随着代数 t 的增加, $\Delta(t, y)$ 靠近 0 的概率逐渐增大, 粒子群逐渐趋于稳定。

2.3 混合算法模型

将上述局部搜索算法与改进的 MOPSO 相结合时, 为使算法的局部搜索与全局搜索能力相平衡, 并兼顾运行效率, 需考虑几个方面^[14, 15]: 1) 局部改进的对象, 即粒子群的选择以及数量; 2) 局部改进的频率, 即局部搜索的周期; 3) 局部改进过程的强度, 即局部搜索半径及个体的局部改进次数。

由于粒子群易陷入局部最优, 因此在算法早期应充分利用其全局搜索能力, 使得到的初始粒子群分布均匀。在算法中后期, 粒子群算法收敛逐渐减慢, 无法进一步有效地提高解集质量, 此时引入侧步爬山策略加速算法收敛并改善解集分布。另一方面, 由于外部种群即精英粒子相对靠近前沿, 若将其作为局部优化对象, 更多地会得到非支配方向, 致使改进效果不明显, 因此局部搜索的对象宜采用内部种群。考虑到算法的效率, 基于侧步爬山策略的局部搜索过程以一定周期进行。

基于上述思想, 局部搜索与粒子群搜索的混合模型见图 2。数量为 N_p 的内部种群经初始化后, 首先依据基本公式进化, 辅之变异算子及惯性权重提高搜索的全局性。在进入算法中后期(当前迭代次数 \geq 最大迭代次数/3)时, 周期性地侧步爬山搜索, 使 $P_{bs} \cdot N_p$ 个粒子中, 一部分作快速下降搜索, 加速收敛, 另一部分个体朝着非支配方向搜索用于粒子群多样性的改进。两种搜索方式以固定周期交互进行, 从而保证算法的运行效率。

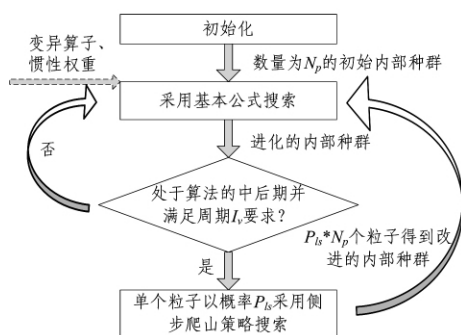


图 2 混合算法模型

2.4 H-MOPSO 算法流程

步骤 1 设置算法基本参数, 设定局部搜索参数, 初始化内部种群, 设置粒子的初始速度为 0, 个体最佳位置为初始位置, 令当前代数 $t = 1$ 。

步骤 2 评价内部种群个体的适应值, 更新个体最优的位置, 并基于拥挤距离排序法^[3]管理外部种群。若外部种群为空, 直接存入得到的非支配个体; 若外部种群非空, 将内部种群个体与外部种群个体相比较, 存入非支配个体并删除被

支配个体; 当外部种群个体数达最大容量 N_a 时, 选择较为拥挤的解替换, 并重新计算拥挤距离并排序。

步骤 3 若 $t < M_t/3$ (M_t 为最大迭代次数) 且 $t \% I_t \neq 0$ (取余操作, I_t 为周期长度), 则不满足局部搜索条件, 进入步骤 4, 否则转步骤 5。

步骤 4 从外部种群的前 20% 随机选取一个全局最优, 依据基本公式更新速度和位置, 转步骤 6。

步骤 5 内部种群个体以概率 P_{bs} 进行基于侧步爬山策略的局部搜索, 进入下一步。

步骤 6 产生一个 $0 \sim 1$ 间的随机数, 若小于变异概率 P_{mut} , 则利用变异算子对粒子进行局部变异。

步骤 7 重复步骤 2—步骤 6, 直到达到最大迭代次数, 输出外部种群。

2.5 计算复杂度分析

H-MOPSO 的主要运算工作在于粒子的外部种群维护及局部搜索过程。设优化问题目标个数为 M , 算法的内部种群规模为 N_p , 外部种群规模为 N_a , 则基于拥挤距离排序法维护外部种群的时间复杂度为 $O(N_a \cdot \log N_a)^{[2]}$ 。局部搜索过程中, 某个粒子进行局部搜索的次数最大为 M_{bs} 。每次局部搜索时, 若当前解与邻居解为支配关系, 可直接计算方向及步长, 时间复杂度分别为 $O(1)$ 和 $O(N_t \cdot M)$ (N_t 为寻找最优步长的尝试次数); 若沿非支配方向搜索, 计算方向及步长的时间复杂度均为 $O(1)$ 。可得, 每次局部搜索的最大时间复杂度为 $O(M \cdot N_t)$, 单个粒子经历完整局部搜索过程的时间复杂度为 $O(M \cdot N_t \cdot M_{bs})$ 。在整体算法中, 每迭代一次有 $P_{bs} \cdot N_p$ ($\leq N_p$) 个粒子进行局部搜索, 对应局部搜索过程的总时间复杂度为 $O(M \cdot N_t \cdot N_p \cdot M_{bs})$ 。综上, 在包含局部搜索过程时, 算法的总时间复杂度可记为: $O(N_a \cdot \log N_a) + O(M \cdot N_t \cdot N_p \cdot M_{bs})$ 。

与 NSGA-II ($O(M \cdot N_p^2)$) 相比, 一次迭代过程中, 两者的时间复杂度比值为:

$$\frac{O(N_a \cdot \log N_a) + O(M \cdot N_t \cdot N_p \cdot M_{bs})}{O(M \cdot N_p^2)} \quad (5)$$

由式(5)可知, 当两者具有相同大小的种群规模 ($N_a = N_p$) 时, 它们的时间复杂度关系与 N_t 、 M_{bs} 和 N_p 的取值大小有关。通常, N_t 取值为 $5 \sim 10$, M_{bs} 的取值为 $5 \sim 10$, N_p 的取值与具体问题有关, 问题越复杂其取值越大(本文中 $N_a = N_p = 100$), 于是有 $N_t \cdot M_{bs} \leq N_p$ 。另一方面, H-MOPSO 在引入局部搜索时采取了周期性方式(周期长度 $I_0 = 10$, 即每迭代 10 次进行一次局部搜索), 多数迭代过程的时间复杂度为 $O(N_p \cdot \log N_p) < O(M \cdot N_p^2)$, 这大大降低了整体算法的时间复杂度。因此, H-MOPSO 的时间复杂度小于 NSGA-II 算法。

3 仿真实验及其分析

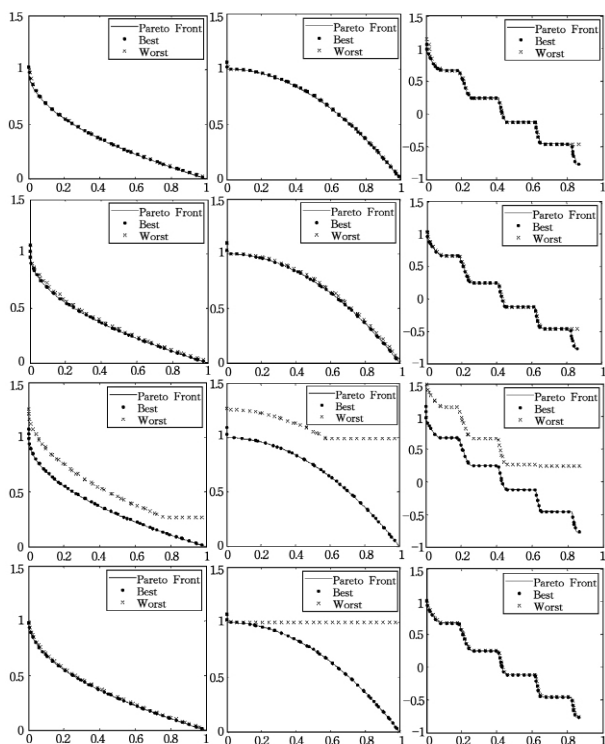
选择常用二维测试函数 ZDT1—ZDT3^[1], 选取代表性多目标进化算法 MOPSO^[2]、NSGA-II^[3] 和 MOEA/D^[16], 采用 SAS 前沿 (Summary Attainment Surface)^[17]、世代距离^[1] (Generational Distance, GD) 和间距指标^[1] (Spacing Metric, SP) 来对比各算法性能, 每个算法对每个测试函数分别做独

立试验 30 次。本文所有实验的平台为 Intel(R) Core(TM) i5 CPU, 2.27GHz, 2G 内存, 实验环境为 Visual Studio 2005 C++。

各种算法种群大小设定为 100, $M_t = 100$, 两种 MOPSO 的变异概率为 0.2, $c_1, c_2 = 1$, 此外, H-MOPSO 中, 局部搜索参数分别取值: $P_{ls} = 0.9$, $I_v = 10$, $M_{ls} = 5$, $r = 0.3$, $N_{ls} = 3$ 。MOEA/D 和 NSGA-II 中的参数 $\eta_c = 15$, $\eta_m = 20$, 交叉概率为 0.9, 变异概率为 $1/a$ (a 为问题的变量个数)^[3,16]。

3.1 SAS 前沿对比

SAS 前沿是对算法多次求解得到的前沿作统计构造所得, 具有统计学意义。30 次实验得到的各算法最好和最差 SAS 如图 3 所示。



从上至下分别为 MOEA/D、NSGA-II、MOPSO 和 H-MOPSO, 从左至右分别为 ZDT1、ZDT2、ZDT3

图 3 算法求解测试函数得到的 SAS 前沿

可知, MOEA/D 在各个二维测试函数的求解上, 效果较好, 具有鲁棒性; NSGA-II 在 ZDT2 的求解上, 不如 MOEA/D, 但整体性能稳定; 而 MOPSO 得到的两条 SAS 差距较大, 说明粒子群快速收敛的不稳定性, 且从最差 SAS 可以看到, 粒子群较容易陷入局部最优; 相对地, H-MOPSO 在 ZDT1 和 ZDT3 问题上的求解效果与 MOEA/D、NSGA-II 相差不大, ZDT3 的 SAS 尾部解集更接近前沿, 说明侧步爬山策略及其混合策略有效改进了 MOPSO 的收敛性与全局性。

3.2 指标评价结果

各种算法 GD 和 SP 指标的最大值、最小值、中值、平均值和标准差见表 1。其中最优值以粗体标记。

从表 1 数据可见, 在 ZDT1—ZDT3 这 3 个问题的求解上, H-MOPSO 相比 MOPSO 有明显的改进, 不论在最小、最大还是均值和标准上, GD 指标各值和 SP 指标均值、最大值及标准差均有数量级的提升, 表明侧步爬山策略的引入使多目标粒子群算法求解得到的前沿与标准前沿距离大大缩小,

解集的分布性也有提高, 算法具有更好的稳定性。

此外, 相比 NSGA-II 和 MOEA/D, H-MOPSO 的 GD 指标均值与标准差也不相上下, 且在 GD 最小值、最大值及中值上, H-MOPSO 更优。同样地, H-MOPSO 在 SP 指标各值上也比这 3 种算法小, 说明 H-MOPSO 在求解这几个测试函数时均有优势。该组数据显示, 本文提出的混合优化算法 H-MOPSO 具有良好的收敛性、分布性与稳定性。

表 1 各算法的 GD 与 SP 指标值对比

(GD, SP)		ZDT1	ZDT2	ZDT3
MOPSO	Min	(0.003003, 0.001337)	(0.011073, 0.000291)	(0.005041, 0.002965)
	Max	(0.027134, 0.032519)	(0.015156, 0.020891)	(0.108312, 0.102648)
	Median	(0.015068, 0.016928)	(0.013115, 0.010594)	(0.056676, 0.052806)
	Mean	(0.008910, 0.003278)	(0.012925, 0.001529)	(0.028455, 0.022013)
	Std. Dev	(0.021349, 0.005646)	(0.043050, 0.003766)	(0.040526, 0.024884)
H-MOPSO	Min	(0.000000, 0.001606)	(0.000000, 0.000553)	(0.000091, 0.003787)
	Max	(0.019022, 0.002912)	(0.004076, 0.001499)	(0.019614, 0.004752)
	Median	(0.009511, 0.002259)	(0.002038, 0.001026)	(0.009852, 0.004270)
	Mean	(0.001453, 0.002290)	(0.000969, 0.000790)	(0.004818, 0.004301)
	Std. Dev	(0.003639, 0.003611)	(0.000210, 0.000260)	(0.000295, 0.000247)
MOEA/D	Min	(0.000012, 0.000619)	(0.000011, 0.000632)	(0.000091, 0.003725)
	Max	(0.045244, 0.049883)	(0.011419, 0.016896)	(0.044333, 0.030558)
	Median	(0.022628, 0.025251)	(0.005715, 0.008764)	(0.022212, 0.017142)
	Mean	(0.001151, 0.004728)	(0.000889, 0.001555)	(0.005723, 0.006993)
	Std. Dev	(0.001147, 0.011271)	(0.000333, 0.003156)	(0.000866, 0.005869)
NSGA-II	Min	(0.000001, 0.001247)	(0.000001, 0.000622)	(0.000023, 0.003954)
	Max	(0.046605, 0.051318)	(0.041760, 0.079765)	(0.033327, 0.031703)
	Median	(0.023303, 0.026283)	(0.020880, 0.040193)	(0.016675, 0.017829)
	Mean	(0.001453, 0.005154)	(0.001216, 0.004548)	(0.004840, 0.005647)
	Std. Dev	(0.001115, 0.009918)	(0.001479, 0.014475)	(0.000776, 0.004993)

4 供应商优选与订货量分配应用

供应商选择是供应链管理中的重要组成部分, 常需要考虑多种因素并在质量、交货可靠性、价格、提前期等多个目标之间权衡, 是一个典型的多目标优化问题。常用的解法主要是借助加权法、罚函数等将评价指标进行转化, 进而采用单目标方法求解^[18]。本文考虑直接用多目标进化算法求得 Pareto 最优解集作为决策者的候选方案, 以避免因目标转化而丢失可行方案。

4.1 问题描述与模型建立

模型假设有: ①在每个供应商处仅采购一种产品。②考虑价格折扣。③所有供应商都不存在产品供货短缺现象。

基于以上假设,供应商优选与订货量分配的基本模型为:

$$\begin{aligned} \text{Min } Z_1 &= \sum_{i=1}^n \sum_{j=1}^{m(i)} c_{ij}^1 x_{ij} \\ \text{Min } Z_2 &= \sum_{i=1}^n \sum_{j=1}^{m(i)} c_{ij}^2 x_{ij} \\ &\vdots \\ \text{Min } Z_k &= \sum_{i=1}^n \sum_{j=1}^{m(i)} c_{ij}^k x_{ij} \end{aligned}$$

$$\text{s. t. } \begin{cases} \sum_{i=1}^n \sum_{j=1}^{m(i)} x_{ij} = D \\ \sum_{j=1}^{m(i)} x_{ij} \leq S_i \\ b_{ij-1} y_{ij} \leq x_{ij} < b_{ij} y_{ij} \\ y_{ij} = \begin{cases} 0, & \text{if } x_{ij} = 0 \\ 1, & \text{if } x_{ij} > 0, \sum_{j=1}^{m(i)} y_{ij} \leq 1 \end{cases} \\ x_{ij} \geq 0, i=1, \dots, n, j=1, \dots, m(i) \end{cases}$$

(6)

各符号含义为: i :供应商序号, $i=1, \dots, n$; n :可供选择的供应商数量; j :折扣区间, $j=1, \dots, m(i)$; $m(i)$:供应商 i 的折扣区间数; b_{ij} :供应商 i 的第 j 个折扣区间的数量水平; c_{ij}^k :供应商 i 在第 j 个折扣区间与第 k 个目标函数的相关系数; D :总需求; S_i :供应商 i 所能提供的最大供货量; y_{ij} :供应商 i 是否在第 j 个折扣区间采购的标记; x_{ij} :采购者从供应商 i 的第 j 个折扣区间的订货量。

4.2 模型的求解与分析

以最小化总成本、延迟交货数和不合格产品数为 3 个目标,以文献[18]中的数据为例,采用 H-MOPSO 与 MOPSO 对该问题进行求解,等式约束采用 ϵ 约束处理机制^[19]。除了搜索半径 $r=0.15$,算法其余各参数的设置同测试函数部分。得到的前沿对比见图 4,其中的拟 Pareto 前沿由种群大小为 500 的 MOPSO 运行 500 代得到。

拟 Pareto 前沿上的部分解见表 2。以第 3 组为例,则各供应商订货量为:

$$\begin{aligned} X &= [x_{14}, x_{23}, x_{34}, x_{44}, x_{54}, x_{64}] \\ &= [156, 20, 276, 296, 156, 175] (\text{万}) \end{aligned}$$

即第二个供应商以第三区间的价格折扣采购 20 万,其余供应商均以第四区间的价格折扣进行采购。相比文献[18]中基于权重得到的最优方案 $Z=[224, 814, 23, 001, 13, 509]$,总成本有所提高,但延迟交货数和不合格产品数均降低了。

表 2 供应商优选方案范例集(单位:万)

Id	Total cost	Total number of delivery lateness	Total number of defective units
1	228.819	18.240	14.907
2	227.264	19.720	14.654
3	226.456	20.880	13.936
4	225.799	21.930	13.285
5	224.451	24.580	13.739

根据 30 次运算结果绘制出平均收敛曲线^[1],见图 5,可明显看出粒子群向前沿的收敛速度有所加快,证明混合搜索过程对粒子群的收敛性具有积极效用。对 SP 指标作统计,见表 3,各个值基本提高 10%及以上,可见 H-MOPSO 得到的解集具有更好的多样性。对运行时间作统计,可见运行时间各值基本减少 25%以上,表明算法的运算效率并没有因为增加局部搜索过程而有所降低,经过对数据文件进行分析,认为这是由于采用侧步爬山搜索策略后,外部种群的规模得到降

低,从而减少了外部种群维护的时间。

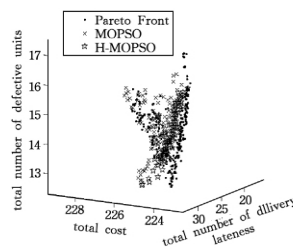


图 4 供应商优选模型的前沿对比

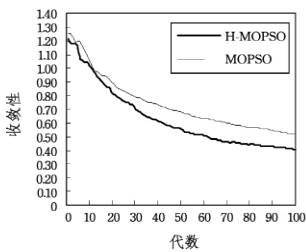


图 5 收敛特征图

表 3 算法求解供应商优选模型的 SP 指标值及运行时间(秒)

		Min	Max	Mean	Median	Std. Dev
SP	MOPSO	0.210466	0.943244	0.401696	0.392056	0.165119
	H-MOPSO	0.209291	0.710016	0.369031	0.352046	0.117630
	提高程度	0.56%	24.73%	8.13%	10.21%	28.76%
Time	MOPSO	0.290	0.950	0.515	0.520	0.143
	H-MOPSO	0.215	0.600	0.392	0.395	0.088
	提高程度	25.86%	36.84%	23.88%	24.04%	38.46%

以上实验说明,H-MOPSO 可以有效地解决此类混合整型多目标优化问题,为决策者提供更高质量的候选解集。性能对比也表明,H-MOPSO 在算法的收敛性、多样性及运算效率方面均优于 MOPSO。

结束语 本文提出一种新的混合多目标粒子群算法 H-MOPSO。首先,通过引入非均匀变异算子与递减的惯性权重策略对 MOPSO 进行改进,降低了其陷入局部最优的概率;并通过建立局部搜索与粒子群优化的混合模型,将基于侧步爬山策略的局部搜索算法周期性引入 MOPSO,有效地改善了算法的收敛性与多样性。经过对标准测试函数集的求解表明,相比 MOPSO、NSGA-II 和 MOEA/D, H-MOPSO 具有更好的收敛性与多样性。对供应商优选与订货量分配模型的求解也证明了该算法的有效性。下一步将研究如何提高 H-MOPSO 对高维问题的求解能力。

参 考 文 献

[1] 郑金华. 多目标进化算法及其应用[M]. 北京:科学出版社, 2007:1-179

[2] Coello C A C, Lechuga M S. MOPSO: a proposal for multiple objective particle swarm optimization[C] // Proceedings of the 2002 Congress on Evolutionary Computation. Piscataway: IEEE Press, 2002:1051-1056

[3] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multi objective Genetic Algorithm: NSGA-II[J]. IEEE transactions on evolutionary computation, 2002, 6(2): 182-197

[4] Coello C A C, Reyes-Sierra M. Multi-Objective Particle Swarm Optimizers: A Survey of The State-of-the-Art[J]. International Journal of Computational Intelligence Research, 2006, 2(3): 287-308

[5] 曾劲涛, 李金忠, 唐卫东, 等. 多目标微粒群优化算法及其应用研究进展[J]. 计算机应用研究, 2011, 28(4): 1225-1229

[6] Ono S, Nakayama S. Multi-Objective Particle Swarm Optimization for Robust Optimization and Its Hybridization with Gradient Search[C] // Proceedings of the 2009 Congress on Evolutionary Computation. Trondheim: IEEE Press, 2009: 1629-1636

(下转第 200 页)

$$\{c,d,e,f\}^+ = \{1,2\}, \{c,d,e,f\}^{++} = \{c,d,e,f\};$$

$$\{b,d,e,f\}^+ = \{1,3\}, \{b,d,e,f\}^{++} = \{b,d,e,f\};$$

$$\{a,b,c,f\}^+ = \{4,5\}, \{a,b,c,f\}^{++} = \{a,b,c,f\};$$

$$\{a,b,c\}^+ = \{4,5,6\}, \{a,b,c\}^{++} = \{a,b,c\};$$

$$M^+ = \emptyset, M^{++} = M; \emptyset^+ = G, \emptyset^{++} = \emptyset.$$

即 $(1, bcdef), (4, abcef), (12, cdef), (13, bdef), (45, abc-f), (456, abc), (\emptyset, M), (G, \emptyset)$ 。

(5) 通过定理 5 得到补背景 (G, M, I^c) 的并不可约元为:

$(1, bcdef), (4, abcef), (12, cdef), (13, bdef), (45, abc-df), (456, abc)$

(6) 通过定理 1 得到补背景的所有概念, 画出表 1 所对应的补背景概念格, 如图 2 所示。

类似地, 对于复杂形式背景, 也可以通过求其补背景的补背景的并不可约元的方法来获得原始概念格。

结束语 本文提出了一种如何从原背景的概念获取补背景的交不可约元(并不可约元)的方法, 针对这种方法给出了它的一个应用——如何求解复杂的形式背景所对应的概念格。由于复杂的形式背景的补背景比较简单, 因此从简单的补背景出发去求解, 找到补背景所对应的概念格, 然后从补背景所有概念的外延(内涵)出发, 得到原背景的交不可约元(并不可约元), 最后利用交不可约元(并不可约元)的性质, 能更好更快地得到原背景的概念格。本文的研究进一步揭示了原背景上的概念格与补背景上的概念格之间的关系, 是对文献[8,9]的完善。

参 考 文 献

[1] Wille R. Restructuring Lattices Theory: An Approach on Hierarchies of Concepts [M]// Riaral I, ed. Ordered Sets. Reidel, Dordrecht, 1982; 445-470

[2] Carpineto C, Romano G. Concept Data Analysis: Theory and Application [M]. John Wiley & Sons, Ltd, 2004

[3] Godin R. Incremental concept formation algorithm based on Galois lattices [J]. Computational Intelligence, 1995, 11(2): 246-267

[4] Bao Ho T. Discovering and using knowledge from unsupervised data [J]. Decision Support System, 1997, 21(1): 29-42

[5] Belohlavek R. fuzzy closure operators [J]. Journal of Mathematical Analysis and Applications, 2001(262): 473-489

[6] Yao Y Y. Concept lattices in rough set theory [C]// Dick S, Kurgan L, Pedrycz W, et al., eds. Proceedings of 2004 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2004). IEEE Catalog Number: 04TH8736, June 2004; 796-801

[7] Yao Y Y. A comparative study of formal concept analysis and rough set theory in data analysis [J]. Lecture Notes in Artificial Intelligence, 2004, 3066: 59-68

[8] 王霞. 概念格的约简理论与方法研究[D]. 西安: 西安交通大学, 2008

[9] 汪胜楠. 面向对象(属性)概念格的构建方法研究[D]. 西安: 西北大学, 2011

[10] Wei Ling, Wang Sheng-nan, Zhao Wei. Methods to construct several kinds of concept lattices [C]// Proceedings of 2010 International Conference on Machine Learning and Cybernetics. IEEE, 2010; 91-96

[11] Ganter B, Wille R. Formal Concept Analysis [M]. Mathematical Foundations. New York: Springer-Verlag, 1999

[12] Davey B A, Priestley H A. Introduction to lattices and order [M]. Cambridge: Cambridge University Press, 2002

(上接第 178 页)

[7] Jia Shu-jin, Zhu Jun, Du Bin, et al. Indicator-Based Particle Swarm Optimization with Local Search [C]// Proceedings of the Seventh International Conference on Natural Computation. Shanghai: IEEE Press, 2011: 1180-1184

[8] Tsou C S, Fang H H, Chang H H, et al. An Improved Particle Swarm Pareto Optimizer with Local Search and Clustering [C]// Proceedings of the 6th International Conference on Simulated Evolution And Learning. Hefei: Springer Press, 2006, 4247: 400-407

[9] Zhang Guo-hui, Shao Xin-yu, Li Pei-gen, et al. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem [J]. Computers & Industrial Engineering, 2009, 56(4): 1309-1318

[10] 贾兆红, 陈华平, 孙耀晖. 多目标粒子群优化算法在柔性车间调度中的应用 [J]. 小型微型计算机系统, 2008, 29(5): 885-889

[11] Lara A, Sanchez G, Coello C A C, et al. HCS: A New Local Search Strategy for Memetic Multi objective Evolutionary Algorithms [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(1): 112-132

[12] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]// Proceedings of the 2000 Conference on Evolutionary Computation. San Diego: IEEE Press, 2000; 84-88

[13] Tripathi P K, Bandyopadhyay S, Pal S K. Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients [J]. Information Science, 2007, 177: 5033-5049

[14] Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling [J]. IEEE transactions on evolutionary computation, 2003, 7(2): 204-223

[15] Nguyen Q H, Ong Y S, Krasnogor N. A Study on the Design Issues of Memetic Algorithm [C]// Proceedings of the 2007 Congress on Evolutionary Computation. Singapore: IEEE Press, 2007; 2390-2397

[16] Zhang Qing-fu, Li Hui. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [J]. IEEE transactions on evolutionary computation, 2007, 11(6): 712-730

[17] Joshua K. A summary-attainment-surface plotting method for visualizing the performance of stochastic multi objective optimizers [C]// Proceedings of the 5th International Conference on Intelligent Systems Design and Applications. 2005; 552-557

[18] Wang Tai-Yue, Yang Yih-Hwang. A fuzzy model for supplier selection in quantity discount environments [J]. Expert Systems with Applications, 2009, 36: 12179-12187

[19] Liang J J, Suganthan P N. Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism [C]// Proceedings of the 2006 Congress on Evolutionary Computation. Vancouver: IEEE Press, 2006; 9-16