

基于差分进化的金字塔演化策略求解一维下料问题



侯 改 何 朗 黄樟灿 王占占 谈 庆

武汉理工大学理学院 武汉 430070

(1797525526@qq.com)

摘 要 一维下料问题是组合优化中一类经典的 NP-hard 问题,被广泛用于机械制造及工程应用等领域。针对传统群智能算法在求解该类问题时难以平衡种群内个体及种群之间开采与探索、竞争与协作矛盾的问题,在金字塔演化策略(Pyramid Evolution Strategy, PES)的基础上,提出了求解一维下料问题的基于差分进化的 PES 算法。该算法充分利用 PES 算法的优势,很好地解决了以上两个矛盾;但是由于 PES 算法未考虑种群当前个体与最优个体之间的协作关系,因此其收敛速度较慢。为此,在 PES 算法的加速过程中引入差分进化的变异、交叉操作,并对产生的不可行试验个体进行修复,使得算法能够充分利用种群个体间的差异信息,这一方面有利于加快算法的收敛速度,另一方面可以实现对新个体附近区域的局部开采,有利于提高算法的求解精度。将提出的算法应用于 6 组一维下料算例,实验结果表明,与其他 8 种算法相比,所提算法在求解精度和收敛速度上均有更好的性能表现,验证了该算法求解一维下料问题的可行性和有效性。

关键词: 群智能算法;金字塔演化策略;一维下料问题;差分进化

中图分类号 TP301.6

Pyramid Evolution Strategy Based on Differential Evolution for Solving One-dimensional Cutting Stock Problem

HOU Gai, HE Lang, HUANG Zhang-can, WANG Zhan-zhan and TAN Qing

School of Science, Wuhan University of Technology, Wuhan 430070, China

Abstract One-dimensional cutting stock problem is a classic NP-hard problem in combinatorial optimization, which is widely used in mechanical manufacturing and engineering applications. It is difficult for traditional swarm intelligence algorithm to balance the contradictions of mining and exploration, competition and cooperation between individuals and populations with in a population when solving one-dimensional cutting stock problem. On the basis of pyramid evolution strategy, an algorithm based on pyramid evolution strategy and differential evolution is proposed in this paper. The algorithm makes full use of the advantages of the PES algorithm and solves these two contradictions well, but the PES algorithm does not consider the cooperative relationship between the current individual and the optimal individual, it may lead the convergence speed of the algorithm to be slowly. To this end, the mutation and crossover operations of differential evolution are introduced into the accelerating process of PES algorithm, and the invalid individuals are corrected by the repair operator. On the one hand, it is conducive to speed up the convergence speed of the algorithm, on the other hand, it can utilize the differences among individuals and realize the local mining of the area near the individuals in the population, which is beneficial to improve the accuracy of the algorithm. The proposed algorithm is applied to six examples. The experimental results show that the algorithm has high optimization accuracy and convergence speed compared with other eight algorithms, which verifies the feasibility and effectiveness of the algorithm for solving the one-dimensional cutting stock problem.

Keywords Swarm intelligent algorithm, Pyramid evolution strategy, One-dimensional cutting stock problem, Differential evolution

1 引言

一维下料问题(One-dimensional Cutting Stock Problem, 1DCSP)是机械、建筑等行业中的常见问题,即将给定数目的单一规格或多规格的线性库存原料切割成不同规格,满足一定数量需求的零件^[1]。如何有效提高原料的利用率并降低生产成本,是相关行业关注的热点。一维下料问题实质是多目

标整数规划问题,属于 NP-hard 问题。目前,国内外求解一维下料问题的方法主要有两类:确切算法和启发式算法。确切算法包括动态规划、割平面法、分支定界法等^[2-3]。这类算法只适用于小规模一维下料问题,对于较复杂的问题存在求解效率差、精度低的缺陷,因此启发式算法逐渐被用于求解一维下料问题。传统启发式算法^[4-8]可以较快地得到下料方案,但不能保证该下料方案符合实际的生产需求。近年来,随着生

收稿日期:2019-05-05 返修日期:2019-07-22 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61672391)

This work was supported by the National Natural Science Foundation of China (61672391).

通信作者:何朗(helang@whut.edu.cn)

产技术的发展和下料问题应用规模的扩大,许多解决一维下料问题的群智能算法被提出,如遗传算法^[9]、蚁群算法^[10]、粒子群算法^[11]等。近几年的研究还倾向于通过改进群智能算法提高一维下料问题的求解质量,改进算法更容易得到性能较优的解。例如,Cheng等^[12]通过改善鱼群的觅食行为和聚群行为,同时设定聚群行为拥挤度因子的上下限,提出一种改进的人工鱼群算法。Shen等^[13]结合遗传算法和模拟退火算法,提出一种基于模拟退火的广义粒子群优化算法。Li等^[14]构造了一种通过“择优插入”的局部搜索算法来辅助遗传算子的混合遗传算法。上述改进的群智能算法中,前两者侧重于改进算法的探索能力,后者侧重于改进算法的开采能力,它们虽然对算法的开采或探索能力进行了一定的改进,但并没有完全解决算法开采与探索之间的矛盾,而且没有考虑种群内个体及种群间竞争与协作的矛盾。

针对当前群智能演化算法信息传递的目的性不明确的问题,Tan^[15]提出一种金字塔演化策略,PES有明确的分工机制与晋升机制,能够较好地平衡开采与探索、竞争与协作的矛盾,使得算法具有良好的跳出局部最优的能力。PES算法由于未考虑种群当前个体与最优个体之间的协作关系,因此收敛速度较慢。基于此,本文提出一种基于差分进化的PES算法(Pyramid Evolution Strategy with Differential Evolution, DEPEs),该算法利用当前个体和最优个体之间的协同交互作用来加快算法的收敛速度,并实现对新个体附近区域的局部开采。PES算法虽然在求解函数优化^[15]、混合整数规划^[16]和彩色图像颜色量化^[17]问题方面有着较好的效果,但是目前还未被应用到一维下料问题等离散组合优化领域。本文设计DEPEs算法来求解一维下料问题,拓宽了PES算法的应用领域。

2 1DCSP的数学模型

一维下料问题是经典的组合优化问题,本文综合考虑单一规格和多规格原材下料的具体特征,建立一维下料问题的通用数学模型。给定 m 种长度分别为 $L_i(i=1, \dots, m)$ 的线性原材,每种原材的数量为 $b_i(i=1, \dots, m)$,现需从所供原材上切割出 n 种零件,零件长度为 $l_j(j=1, \dots, n)$,对应数量为 $d_j(j=1, \dots, n)$,目标是寻找最优的下料方案,使得所用线性原材总量最少,并且除最后一根原材的余料尽可能多之外,其余下料原材的余料要尽可能少。一维下料问题的数学模型如下:

$$\begin{aligned} \min z = & \left(\sum_{k=1}^K x_k L_{g(k)} - \sum_{i=1}^n l_i d_i - l \right) \\ \text{s. t. } & \begin{cases} \sum_{i=1}^n a_{ki} l_i \leq L_{g(k)}, & k=1, \dots, K \\ \sum_{k=1}^K a_{ki} x_k = d_i, & i=1, \dots, n \\ x_k \leq b_{g(k)}, & k=1, \dots, K \\ a_{ki} \geq 0, & a_{ki} \in Z \end{cases} \end{aligned} \quad (1)$$

其中, K 为下料方案所使用的切割方式数; x_k 表示下料方案中第 k 类切割方式的使用数量; $g(k)$ 表示第 k 种切割方式对应使用原材的编号, $g(k) \in \{1, \dots, m\}$; a_{ki} 表示第 k 种切割方式切割出第 i 类零件的数量; l 表示最后一根下料原材的余料。

在该模型的约束条件中,第1个约束表示线性原材的尺寸约束;第2个约束表示零件的需求约束;第3个约束表示线性原材的用量约束;第4个约束表示变量的取值范围约束。

3 金字塔演化策略

PES算法^[15]是一种基于金字塔演化策略的群智能演化算法。该算法根据适应度值将种群按照一定的数量比例划分为探索层、传递层I、传递层II和开采层,同层个体之间存在竞争关系,层与层之间是协作关系。

如图1所示,PES算法中各层之间分工明确。探索层由适应度值低的个体组成,分布于金字塔结构的底层,侧重于探索工作。开采层由适应度值高的个体组成,分布于金字塔结构的顶层,侧重于开采工作。传递层由适应度值中等的个体组成,分布于金字塔结构的中间部分,主要负责3项任务:1)兼顾开采与探索工作,通过层内竞争挖掘同层中的潜在优秀个体;2)通过层间协作,将从底层探索到的优秀个体传递至顶层,实现优秀个体自下至上的动态流动;3)对从底层传递上来的潜在优秀个体采取保护培养措施,避免其与该层的其他个体进行直接竞争而被淘汰。

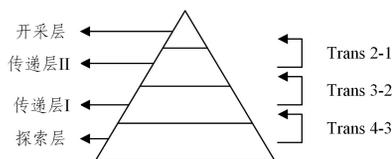


图1 金字塔结构^[15]

Fig. 1 Pyramid structure^[15]

PES算法采用分层策略平衡优化过程中的开采与探索,采用层内竞争与层间协作策略平衡竞争与协作。金字塔结构中各层分工合作,层层递进,能够有效地提高算法的寻优效率。

4 基于差分进化的PES算法

PES算法主要针对连续函数优化问题而提出,因而不能直接应用于一维下料问题的求解。考虑到DE算法可以引导种群中的个体向全局最优个体逼近,且具有较强的收敛性能^[18],本文在PES算法的加速过程中引入DE算法的变异、交叉操作,使得算法在不断跳出局部最优的同时可以加快收敛。

4.1 种群初始化

一维下料问题的每个候选下料方案都对应一个双层编码结构,因此其不能直接应用基本PES算法的种群初始化方法。基于此,本文设计两阶段法生成初始下料方案:第一阶段采用双层编码结构生成原始解;第二阶段通过修正算法^[14]将原始解修正为下料问题的初始解,得到初始下料方案。

双层编码结构的设计如下:将初始解对应的双层编码结构表示为一个二元组 (X, Y) ,其中 $X \in [1, m]^{D_1}$ 和 $Y \in [1, n]^{D_2}$ 是随机产生的实值变量,分别构成原材使用序列和零件切割序列的搜索空间。本文中 D_1 和 D_2 分别取 $\sum_{i=1}^m b_i$ 和 $\sum_{i=1}^n d_i$; m 和 n 分别表示原材和零件的种类编号,并且每类编号在编码列中出现的频次分别为 b_i 和 d_i ,其中 b_i 和 d_i 分别表示每类原材的供应数量和每类零件的需求数量。

任意一个候选解 S 可由目标函数 f 进行评估:

$$f = \frac{1}{\sum_{k=1}^K x_k L_{g(k)} - l} \quad (2)$$

其中, K 为候选解 S 使用的切割方式数; x_k 表示第 k 类切割方式的使用数量; $g(k) \in \{1, 2, \dots, m\}$ 表示第 k 种切割方式对应使用原材的编号; l 表示最后一根下料原材的余料。

4.2 种群分层

基于双层编码结构的初始种群是随机产生的,种群中个体间信息传递的目的性不明确,而分层策略能够使个体间的职责分工更加明确。

1)对于初始种群 X_0 ,计算种群中个体的适应度值,将 X_0 按照适应度值进行升序排序:

$$F_0 = \{f(x_1), f(x_2), \dots, f(x_N)\} \quad (3)$$

$$[F, I] = \text{sort}(F_0) \quad (4)$$

$$X = X_0(I) \quad (5)$$

其中, N 为种群规模; F_0 是初始种群 X_0 对应的适应度值集合; F 是 F_0 经过升序排列后得到的适应度值集合; X 是 X_0 按照 F 进行升序排列后的种群; I 是 X 在初始种群 X_0 中对应的位置索引。

2)按照递减的分层比例 D ,对以上所获得种群 X 中的个体进行分层:

$$X = \{X_4, X_3, X_2, X_1\} \quad (6)$$

$$n_i = N \times D_i \quad (i=1, 2, 3, 4) \quad (7)$$

$$n_i < n_{i+1} \quad (i=1, 2, 3) \quad (8)$$

其中, X_1, X_2, X_3, X_4 分别对应开采层、传递层 II、传递层 I 和探索层; n_i 为第 i 层的个体数目; D_i 为第 i 层的分层比例。式(7)和式(8)是各层个体在数量上的约束。

4.3 开采与探索

一维下料问题编码的特殊性,使得 PES 算法已有的开采与探索规则并不可行。本文根据遗传算法的交叉操作,设计如下的开采与探索规则来对 X 中的个体进行更新。

1)设第 i 层的最优个体 $x_{i,\text{best}}$ 为父代 1,第 j 个个体 $x_{i,j}$ 为父代 2,按照该层搜索半径 r_i 选取 $x_{i,\text{best}}$ 中的若干基因位,并将其按照一个随机顺序重排后遗传给子代的相应基因位,同时将父代 1、父代 2 中的对应基因置空。记当前的子代为 $x_{i,\text{mid}}$ 。

2)对于 $x_{i,\text{mid}}$ 中的空缺基因位,设置追随概率为 p_f ,则子代的更新公式如下:

$$x_{i,\text{new}} = x_{i,\text{mid}}(r_i, p_f) \quad (9)$$

即当 $\text{rand} > p_f$ 时,子代追随该层最优个体,遗传父代 1 的对应基因;否则,遗传父代 2 的对应基因,将子代 $x_{i,\text{mid}}$ 更新为子代 $x_{i,\text{new}}$ 。采用这种开采与探索规则,不仅可以使子代保留最优个体的优良基因,而且可以控制子代跟随该层最优个体的随机性,从而维持种群的多样性。

种群中个体开采与探索的能力随着其所处位置的不同而改变。从探索层至开采层,开采能力逐步加强,探索能力逐步减弱,个体搜索的随机性不断被抑制,因此搜索半径要不断减小。

设各层搜索半径为 $R = \{r_1, r_2, r_3, r_4\}$,其具体形式如下:

$$r_i = \text{ceil}(r_{i+1} \times \omega), i=1, 2, 3 \quad (10)$$

$$r_i < r_{i+1}, i=1, 2, 3 \quad (11)$$

其中, $\text{ceil}(\cdot)$ 为向上取整函数, r_i 代表第 i 层的搜索半径, ω 为半径收缩因子。式(11)是对每层个体搜索范围的限制。

4.4 加速过程

为更好地对种群中的个体进行初步筛选,对经过开采与探索得到的新个体进行加速操作。该操作充分考虑了种群当前个体与最优个体之间的协作关系,并以此来加快算法的收敛。加速操作采用差分进化^[19]的变异、交叉策略来实现。

1)变异策略。本文采用变异策略 DE/current-best/1/bin 对每层产生的新个体 x_i^t 按式(12)进行变异操作。

$$v_i^{t+1} = x_i^t + \lambda \times (x_{\text{best}} - x_i^t) + F \times (x_{i_1}^t - x_{i_2}^t) \quad (12)$$

其中, $i \neq r_1 \neq r_2, x_{r_1}, x_{r_2} \in X$; t 表示当前迭代次数; λ 和 F 为变异算子,控制偏差向量放大的比例。

2)交叉策略。交叉操作的原理是利用当前种群中个体的部分分量和变异个体的对应分量,按照某种规则进行交换,生成试验个体 u_i^{t+1} ,具体操作如式(13)所示:

$$u_i^{t+1} = \begin{cases} v_i^{t+1}, & \text{if } \text{rand} < CR \text{ or } j = j_{\text{rand}} \\ x_i^t, & \text{otherwise} \end{cases} \quad (13)$$

其中, CR 为交叉算子, rand 表示 $[0, 1]$ 内的随机数, j_{rand} 表示编码列中的一个随机维度。

4.5 修复算子

一维下料问题的实值双层编码方式,使得经过差分进化变异、交叉操作产生的试验个体 u_i^{t+1} 在很大程度上为不可行解。本文定义处理不可行解的修复算子 (Repair Operator, RO) (以双层编码中的零件序列为例) 如下。

1)将 u_i^{t+1} 的对应编码按式(14)进行排序操作。

$$[u, \text{index}] = \text{sort}(u_i^{t+1}) \quad (14)$$

其中, u 为排序后的试验个体, index 为 u 在 u_i^{t+1} 中对应的位置索引。

2)将 u 中的零件编码按照每类零件的需求数目 d_j ($j=1, \dots, n$) 依次划分为 n 个部分,每部分赋值为相应的零件类别,并将 u 中元素按照位置索引 index 对应到 u_i^{t+1} 中,从而得到修复后的试验个体 u_i^{t+1} 。

$$u_i^{t+1} = u(\text{index}) \quad (15)$$

4.6 层间传递

层间传递涉及输送层和接收层的相互协作,采用锦标赛选择机制^[20]从输送层选择相应数目的个体传递至接收层,并对传递上来的潜在优秀个体按照该层的开采与探索规则进行孵化培养,避免其与该接收层的其他个体进行直接竞争而被淘汰,充分挖掘次优个体的潜力,从而实现层与层之间的相互协作。

4.7 整合新种群

采用保留精英的轮盘赌策略从每层产生的新个体、加速个体、传递个体以及孵化个体中选择相等数目的个体来更新该层。按照以上更新方式整合的新种群中可能会出现相同的个体,为保持种群的多样性,对于相同个体只保留一个,其余个体采用随机产生的个体代替。

基于差分进化的 PES 算法的流程如图 2 所示。

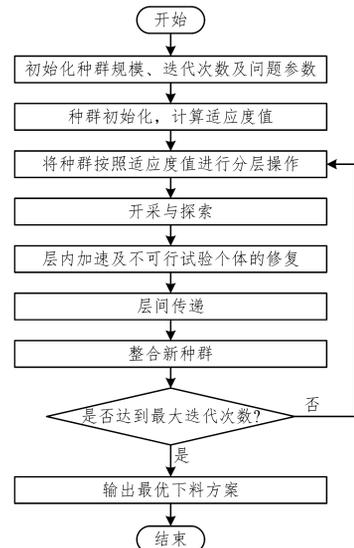


图 2 DEPES 算法的流程图

Fig. 2 Flowchart of DEPES algorithm

5 仿真实验

5.1 算例及实验设置

为评估 DEPES 算法求解一维下料问题的性能,本文对 6 组一维下料问题的标准算例^[5-8,13]进行仿真实验。这些算例均从实际的一维下料问题中抽象而来,具有很强的代表性。第一类算例 F_1 ^[13], F_2 ^[5]和 F_3 ^[8]是单一规格原材下料的情形,第二类算例 F_4 ^[7], F_5 ^[6]和 F_6 ^[5]是多规格原材下料的情形。通过这两类算例可以全面检验 DEPES 算法的性能,具体实验参数如表 1 所列。

表 1 实验参数

Table 1 Experimental parameters

算例	N	D	ω	R	F	λ	CR	G
F_1	40	[0.1,0.2,0.3,0.4]	0.5	[3.5,10,20]	0.5	0.7	0.1	1000
F_2	40	[0.1,0.2,0.3,0.4]	0.5	[3.5,10,20]	0.5	0.7	0.1	1000
F_3	40	[0.1,0.2,0.3,0.4]	0.5	[3.5,10,20]	0.5	0.7	0.1	1000
F_4	40	[0.1,0.2,0.3,0.4]	0.5	[3.5,10,20]	0.5	0.7	0.1	1000
F_5	40	[0.1,0.2,0.3,0.4]	0.5	[3.5,10,20]	0.5	0.7	0.1	1000
F_6	40	[0.1,0.2,0.3,0.4]	0.5	[4.8,15,30]	0.5	0.7	0.1	1000

5.2 实验结果的比较与分析

DEPES 算法、PES 算法和 DE 算法分别运行 30 次后,6 个问题对应最优下料方案的原材平均利用率、平均利用率(除末根原材外)及最后一根下料原材的余料长度如表 2 所列。

表 2 各算法的实验结果

Table 2 Experimental results of each algorithm

算例	算法	平均利用率/%	末根原材的利用率/%	平均利用率(除末根下料原材)/%	最后一根下料原材的余料
F_1	DEPES	97.67	50.51	99.82	8909
	PES	97.67	52.11	99.74	8621
	DE	97.67	52.82	99.71	8492
	AGPSO	97.67	52.62	99.58	8529
F_2	DEPES	90.00	20.00	100.00	2.4
	PES	90.00	20.00	100.00	2.4
	DE	90.00	20.00	100.00	2.4
	文献[5]算法	90.00	40.00	97.14	1.8
F_3	DEPES	97.40	93.00	97.71	70
	PES	97.40	93.00	97.71	70
	DE	97.40	93.20	97.70	68
	HHA	97.40	94.30	97.62	57
F_4	DEPES	97.53	81.54	98.16	24
	PES	97.35	85.39	98.01	21
	DE	96.56	88.46	97.01	15
	GRH	97.39	65.71	98.39	36
F_5	DEPES	95.57	77.34	96.11	2485
	PES	95.22	70.87	95.97	3485
	DE	95.04	68.67	95.84	3435
	文献[6]算法	91.43	70.87	92.11	3485
F_6	DEPES	99.63	97.13	99.73	430
	PES	99.43	87.31	99.70	1269
	DE	99.06	64.60	99.71	5310
	GA	99.38	100.00	99.48	60

从表 2 可以看出,对于问题 $F_1 - F_6$, DEPES 算法均能在算法迭代 1000 次以内找到最优下料方案。对于单一规格原材下料问题 $F_1 - F_3$, 从下料方案对应的平均利用率这一指标来看, DEPES 算法均已达到已知最优; 从平均利用率(除末根下料原材)和最后一根下料原材的余料来看, DEPES 算法在保证平均利用率不变的情况下,可以使其他原材(除末根原材外)的利用率达到最大,并且能够使下料方案的余料尽可能集中到最后一根下料原材中,以方便后续下料。对于多规格原材下料问题 F_5 和 F_6 , 从平均利用率、末根原材的利用率、

平均利用率(除末根下料原材)和最后一根下料原材的余料这 4 个指标综合来看, DEPES 算法所得下料方案优于 DE 算法、PES 算法和已有文献中的最优结果。

对于多规格原材下料问题 F_4 , 虽然从平均利用率(除末根下料原材)和最后一根下料原材的余料的角度来看, DEPES 算法略差于基于贪婪舍入的启发式算法(GRH), 但是其在原材的平均利用率方面有较好表现, 这也表明了 DEPES 算法具有很强的跳出局部最优的能力。究其原因, GRH 算法的贪婪性导致算法过于强调当前的局部最优, 从而无法达到全局最优, 本文所提算法可以弥补这一缺陷。综合来看, DEPES 算法针对一维下料问题的求解性能优于 DE 算法和 PES 算法, 而且与自适应的广义粒子群算法、随机启发式算法、混合启发式算法、基于贪婪舍入的启发式算法、多阶段启发式算法以及遗传算法相比, 其对单一规格和多规格原材下料问题的平均利用率(除末根下料原材)分别平均提高了 1.09% 和 1.45%。因此, 对于大部分算例, DEPES 算法均能够有效提高下料方案的求解精度, 具有明显的性能优势。

5.3 算法性能分析

DEPES 算法综合利用了 DE 变异、交叉策略和 PES 的基本框架, 通过与 DE 和 PES 进行实验对比, 可以验证 DEPES 算法的有效性。同时, 通过比较 3 种算法在 6 个算例上随机独立运行 30 次的平均收敛曲线, 可以直观地了解算法的收敛速度和求解性能。

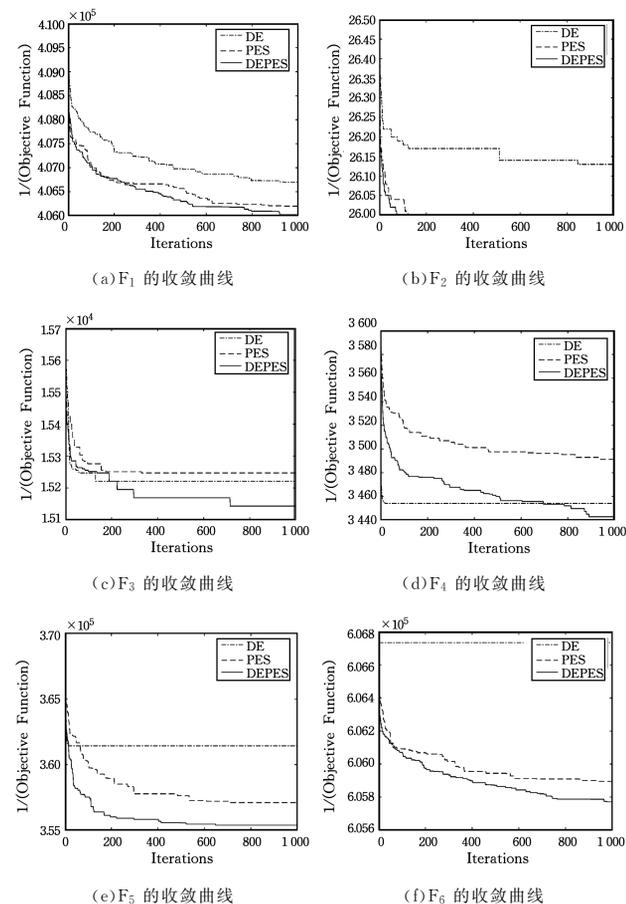


图 3 3 种算法在 6 个算例上的收敛曲线

Fig. 3 Convergence curves of three algorithms on six examples

从图 3 可以明显看出,DEPES 算法在 6 个测试算例上都表现出了较好的性能;其次是 PES 算法;性能表现较差的是 DE 算法,并且随着下料规模的扩大,DE 算法更容易陷入局部最优。由图 3(e)、图 3(f)可以明显看出,DE 算法在 F_5 、 F_6 上的收敛曲线呈一条直线,造成这种现象的原因是:在算法早期,种群中出现了适应值远远超过当前种群平均适应值的个体,当所有个体在收敛前都没有找到一个更佳的后代个体时,进化就会陷入停滞状态;与此同时,随着迭代次数的增加,种群的多样性迅速降低,这也导致 DE 算法较早收敛于局部最优解。从根本上来说,DEPES 算法一方面通过充分利用 PES 的优势,缓解了种群内部个体及种群之间开采与探索、竞争与协作的矛盾;另一方面,在 PES 算法的加速过程中引入差分进化的变异、交叉操作,促进了算法在种群中个体附近区域的局部开采,这些策略均有利于提高算法的求解性能。

结束语 本文提出了一种基于差分进化的 PES 算法,将差分进化的变异、交叉操作引入 PES 算法的加速过程中,并将其用于求解一维下料问题。在 6 个测试算例上的运行结果表明:DEPES 算法在收敛速度、求解精度方面均有明显改善,是一种求解一维下料问题的有效方法。将来可以结合并行计算来提高算法的运行效率,并考虑将 DEPES 算法应用于多维下料问题。

参考文献

- [1] CHERRI A C, ARENALES M N, YANASSE H H, et al. The one-dimensional cutting stock problem with usable leftovers—A survey [J]. *European Journal of Operational Research*, 2014, 236(2): 395-402.
- [2] DELORME M, IORI M, MARTELLO S. Bin packing and cutting stock problems: Mathematical models and exact algorithms [J]. *European Journal of Operational Research*, 2016, 255(1): 1-20.
- [3] ALIANO F A, MORETTI A C, PATO M V. A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem [J]. *Journal of the Operational Research Society*, 2018, 69(1): 91-107.
- [4] CUI Y, SONG X, CHEN Y, et al. New model and heuristic solution approach for one-dimensional cutting stock problem with usable leftovers [J]. *Journal of the Operational Research Society*, 2017, 68(3): 269-280.
- [5] ZHU S L. The research on optimization algorithms for one-dimensional cutting stock problems [D]. Wuhan: Huazhong University of Science and Technology, 2013.
- [6] GRADIŠARM, CESAR M, TOMAT L. One-dimensional cutting stock optimisation by suborders [J]. *Tehnički Vjesnik-Technical Gazette*, 2018, 25(2): 474-480.
- [7] POLDI K, ARENALES M. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths [J]. *Computers & Operations Research*, 2009, 36(6): 2074-2081.
- [8] GUAN W L, GONG J, XUE H T. A hybrid heuristic algorithm for one-dimensional cutting stock problem [J]. *Machinery Design & Manufacture*, 2018(8): 237-239.
- [9] TIEN N D. A genetic algorithm approach for large-scale cutting stock problem [M] // *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2018: 796-805.
- [10] EVTIMOV G, FIDANOVA S. Ant colony optimization algorithm for 1D cutting stock problem [M] // *Advanced Computing in Industrial Mathematics*. Cham: Springer, 2017: 25-31.
- [11] ASVANY T, AMUDHAVEJ J, SUJATHA P. One-dimensional cutting stock problem with single and multiple stock lengths using DPSO [J]. *Advances and Applications in Mathematical Sciences*, 2017, 17(1): 147-163.
- [12] CHENG C, BAO L. An improved artificial fish swarm algorithm to solve the cutting stock problem [C] // *International Symposium on Neural Networks*. Cham: Springer, 2018: 165-172.
- [13] SHEN X J, YANG J C, YING W Q, et al. Adaptive general particle swarm optimization for one-dimension cutting stock problem [J]. *Journal of South China University of Technology (Natural Science Edition)*, 2007, 35(9): 113-117.
- [14] LI B, HE F. Improved hybrid genetic algorithm for solving one-dimensional cutting stock problem [J]. *Journal of Inner Mongolia University (Natural Science Edition)*, 2014, 45(3): 245-250.
- [15] TAN Q. Swarm intelligent evolution strategy based on pyramid structure [D]. Wuhan: Wuhan University of Technology, 2018.
- [16] TANG H H, PENG S J, WANG Z Z. Swarm intelligent evolution strategy based on pyramid structure for solving mixed integer programming problems [J]. *Application Research of Computers*, 2020, 37(5).
- [17] LI H, JIANG D Y, HUANG Z C, et al. Method for solving color images quantization problem [J]. *Journal of Computer Applications*, 2019, 39(9): 2646-2651.
- [18] LI X, MA S, HU J. Multi-search differential evolution algorithm [J]. *Applied Intelligence*, 2017, 47(1): 231-256.
- [19] ZOU H F, XIE C W, ZHOU Y P, et al. Group search optimization with opposition-based learning and differential evolution [J]. *Computer Science*, 2018, 45(S1): 124-129.
- [20] AL-BETAR M A, AWADALLAH M A, FARIS H, et al. Bat-inspired Algorithms with Natural Selection mechanisms for Global optimization [J]. *Neurocomputing*, 2018, 273: 448-465.



HOU Gai, born in 1993, postgraduate. Her main research interests include swarm intelligent computation and so on.



HE Lang, born in 1974, Ph.D, associate professor. His main research interests include evolutionary computation and so on.