

考虑安全约束的连退生产过程多目标操作优化

王显鹏^{1,3}, 杨立文^{1†}, 董志明^{1,2}, 张博¹

(1. 东北大学 信息科学与工程学院, 沈阳 110004; 2. 辽宁省制造系统与物流优化重点实验室, 沈阳 110004;
3. 辽宁省智能工业数据解析与优化工程实验室, 沈阳 110004)

摘要: 针对连退生产过程中经常出现的薄料带钢跑偏问题, 建立考虑安全约束的连退生产过程多目标操作优化模型, 并针对问题特点提出一种基于分类和多种群竞争协调的多目标进化算法(MOEA-CMCC). 在算法中引入具有不同进化策略的多个种群以增强搜索的多样性, 并在多种群之间引入竞争机制和信息共享的协调机制以提高算法的鲁棒性; 通过对外部档案集中的解进行分类并在类内进行局部搜索, 以保证外部档案集的分散性和算法的收敛速度. 基于Benchmark问题的实验结果表明, 所提出的算法具有较好的收敛性和分散性; 对连退操作优化问题的实验结果表明, 所提出的算法能够有效求解该问题.

关键词: 安全约束; 多目标优化; 连续退火生产过程; 操作优化

中图分类号: TP18

文献标志码: A

Multi-objective operation optimization of continuous production process with safety constraints

WANG Xian-peng^{1,3}, YANG Li-wen^{1†}, DONG Zhi-ming^{1,2}, ZHANG Bo¹

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. Liaoning Key Lab of Manufacturing Systems and Logistics, Shenyang 110004, China; 3. Liaoning Engineering Laboratory of Operations Analytics and Optimization for Smart Industry, Shenyang 110004, China)

Abstract: To deal with the running deviation of thin strip in the continuous annealing production process, a multi-objective operation optimization model with the consideration of safety constraints is established, and a multi-objective evolutionary algorithm is proposed, in which the classification and the multi-population with competition and coordination are incorporated. The mechanism of multi-population with different evolution strategies is established to enhance the search diversity, and the competition and cooperation based on information sharing is adopted to improve the robustness of the algorithm. The solutions in the external archive(EXA) are classified into different groups, and then local search is performed within each group, which can help to ensure the diversity of the non-dominated solutions in EXA and improve the convergence speed of the algorithm. Computational results based on benchmark problems show that the proposed algorithm has good convergence and diversity, and further results based on the practical operation optimization of continuous annealing illustrate the effectiveness of the proposed algorithm.

Keywords: safety constraint; multi-objective optimization; continuous annealing production process; operation optimization

0 引言

生产过程操作优化在流程工业自动化系统中位于生产调度层与过程控制层之间, 主要是根据企业当前的生产条件与生产调度层制定的生产计划, 通过算法求解最优的操作条件, 为过程控制层提供设定值. 许多学者对流程工业各领域中的操作优化问题

进行了研究^[1-3].

连续退火生产过程^[4]是钢铁企业冷轧厂的重要环节, 其功能为按照设定的退火工艺曲线依次经过预热炉、加热炉、均热炉、缓冷炉、快冷炉、过时效炉、终冷炉等对热轧厂出品的带钢进行热处理, 从而提高带钢的性能. 但其生产过程包含很多复杂的阶段, 并

收稿日期: 2017-09-18; 修回日期: 2017-11-13.

基金项目: 国家重点研发计划项目(2016YFB0901900); 国家自然科学基金项目(71790614, 71621061, 61573086); 教育部111创新引智基地项目(B16009).

责任编委: 刘德荣.

作者简介: 王显鹏(1980-), 男, 教授, 博士, 从事基于机器学习的工业生产过程建模、操作优化、进化算法等研究; 杨立文(1990-), 女, 硕士生, 从事工业生产过程操作优化、进化算法的研究.

†通讯作者. E-mail: ytyangliwen@163.com

且各阶段包含众多相互耦合的操作变量,使得操作难度较大,传统的人工经验方式难以满足实际生产的需要.因此,Guo等^[5]以最小化能耗为目标建立了连退加热过程的操作优化模型并使用PSO算法对其求解.Zhang等^[6]通过对连退生产过程的机理分析,建立了基于数据驱动的操作优化模型.Yang等^[7]建立了连退生产过程的多目标操作优化模型并对其进行求解,降低了能源消耗,提高了产品的质量和连退机组的产能.

针对多目标优化问题的求解,近年来出现了很多优秀的算法,如多策略改进的多目标粒子群优化算法^[8]、基于目标分解的高维多目标并行进化优化方法^[9]和NSGA-II^[10]等.文献[11-12]引入分类机制,实现了对新产生候选解的过滤.然而,上述算法并没有充分利用种群内个体携带的信息及其之间信息的共享,也没有充分利用分类机制来提高解的性能.

由上述分析可知,针对流程工业生产过程操作优化的研究主要集中在单目标操作优化,而对于多目标操作优化,大多是通过加权方式转化为单目标操作优化,并且很少考虑生产安全问题.在连续退火的生产过程中,安全性约束对于机组的连续生产和安全运行至关重要.在实际生产中,薄料带钢在连退均热炉中经常因为带钢张力的波动而导致瓢曲和跑偏,当跑偏达到一定程度时,纠偏辊无法完成纠偏而导致断带,需要整个连退生产线停机进行检查,在排查故障辊的过程中需要消耗大量的时间,从而给企业造成巨大的经济损失.为此,本文针对冷轧连退生产过程,建立考虑质量和连续运行等安全约束的多目标操作优化模型,并针对问题特点提出一种基于分类与多种群竞争协调的多目标进化算法(MOEA-CMCC).

1 考虑安全约束的连退多目标操作优化建模

1.1 问题描述

冷轧连退生产过程操作优化是根据待加工带钢的规格和当前的设备条件,在满足工艺约束、质量约束和保障连续安全生产的前提下,以提高产品质量、降低能源消耗、提高产能和对工艺流程的执行度为目标,确定连退各阶段的温度、轧制力、张力和带钢速度等控制参数的最优设定值.

1.2 问题模型

1.2.1 模型参数

模型参数 $S = (s_0, s_1, \dots, s_{24})^T$ 中各符号的含义如下:

s_0, s_1 : 带钢入口厚度和宽度;

s_2 : 出炉温度;

$s_3 \sim s_5$: 平均卷曲、精轧、终轧温度;

$s_6 \sim s_{12}$: 带钢C、S、P、Mn、N、Si、Al的含量;

s_{13} : CDCM延伸率;

s_{14}, s_{15} : 缓冷炉1区和2区温度;

s_{16}, s_{17} : 1#和2#冷炉冷却气体温度;

s_{18} : 终冷炉2区水温;

s_{19}, s_{20} : 均热炉内辊子的个数和半径;

s_{21} : 电机转轴上的转动惯量;

s_{22} : 带钢的弹性模量;

s_{23} : 辊子间带钢的长度;

s_{24} : 均热炉入口处的带钢张力 F_0 .

1.2.2 决策变量

决策变量 $X = (x_0, x_1, \dots, x_{73})^T$ 中各符号的含义如下:

x_0 : 带钢穿行速度的设定值;

$x_1 \sim x_5$: 加热炉1区~5区温度的设定值;

x_6, x_7 : 均热炉1区和2区温度的设定值;

$x_8 \sim x_{10}$: 1#冷炉1区~3区温度的设定值;

x_{11}, x_{12} : 1#过时效炉1和2区温度的设定值;

$x_{13} \sim x_{16}$: 2#过时效炉4个区温度的设定值;

x_{17} : 2#冷炉温度的设定值;

x_{18} : 终冷炉1区水温的设定值;

x_{19} : 平整机延伸率;

x_{20}, x_{21} : 1#和2#平整机轧制力;

$x_{22} \sim x_{24}$: 平整机入口、中间、出口张力;

x_{25} : 均热炉入口速度;

$x_{26} \sim x_{49}$: 均热炉内24个辊子速度的设定值;

$x_{50} \sim x_{73}$: 均热炉内24个辊子扭矩的设定值.

1.2.3 数学模型

考虑安全约束的连退生产过程多目标操作优化模型如下:

$$\min F = [f_1(X, S), f_2(X), f_3(X), f_4(X, S)]. \quad (1)$$

$$\text{s.t. } x_{i,\min} < x_i < x_{i,\max}, \quad i = 0, 1, \dots, 73; \quad (2)$$

$$x_1 < x_2 < x_3 < x_4 < x_5; \quad (3)$$

$$a_1 < x_5 - x_1 < b_1; \quad (4)$$

$$a_2 \leq x_7 - x_6 \leq b_2; \quad (5)$$

$$a_3 \leq x_8 - x_9 \leq b_3; \quad (6)$$

$$a_4 \leq x_9 - x_{10} \leq b_4; \quad (7)$$

$$a_5 \leq x_{12} - x_{11} \leq b_5; \quad (8)$$

$$a_6 \leq x_{14} - x_{13} \leq b_6; \quad (9)$$

$$a_7 \leq x_{14} - x_{15} \leq b_7; \quad (10)$$

$$a_8 \leq x_{16} - x_{15} \leq b_8; \quad (11)$$

$$L_i \leq \psi_i(X, S) \leq U_i, \quad i = 1, 2, \dots, 6; \quad (12)$$

$$0 < \frac{F_1(X, S) - F_0}{F_0} \leq \sigma\%; \quad (13)$$

$$0 < \frac{F_{i+1}(X, S) - F_i(X, S)}{F_i(X, S)} \leq \sigma\%, \quad (14)$$

$$i = 1, 2, \dots, 23.$$

其中: $x_{i,\min}$ 、 $x_{i,\max}$ 分别为第 i 个决策变量的下限和上限; a_i 、 b_i 分别为连退生产过程中工艺约束的下限和上限; $\psi_1(X, S) \sim \psi_6(X, S)$ 分别为加热炉、均热炉、1#冷炉、1#过时效炉、2#过时效、2#冷炉出口带钢的温度; L_i 、 U_i 分别为各炉出口带钢温度的下限和上限; F_0 为均热炉入口处的带钢张力; $F_i(X, S)$ ($i = 1, 2, \dots, 23$) 为均热炉内第 i 个辊子与第 $i + 1$ 个辊子之间的带钢张力; $F_{24}(X, S)$ 为出口处的带钢张力。

式(3)~(11)为连退生产过程中加热炉、均热炉、1#冷炉、1#过时效炉、2#过时效炉炉内的工艺约束,以保证各炉内温度的设定满足工艺要求;式(12)为连退生产过程的产品质量安全约束,用来保证各炉出口处带钢的质量能够满足工艺要求;式(13)和(14)为连退生产过程的生产安全约束,即生产过程中需要保证各辊子入口和出口处的带钢张力满足 $F_i < F_{i+1}$,以避免薄料带钢出现打滑跑偏。其中,每个辊子处的带钢张力使用文献[4]所提出的机理模型求解获得。

2 基于分类与多种群竞争协调的多目标进化算法(MOEA-CMCC)

2.1 基于多种群的进化策略

不同的交叉算子具有不同的搜索行为和进化方向,针对不同问题其搜索性能也各不相同。因此,在本文提出的算法(MOEA-CMCC)中引入多个种群,并为每个种群分配不同的交叉算子,动态地调整种群的大小,从而能够通过多种群机制保证解的分散性,并通过多种交叉算子提高算法求解不同类型问题时的鲁棒性。所采用的交叉和变异算子如下。

1) 交叉算子。

在MOEA-CMCC算法中设置5个种群(P_i 表示第 i 个种群),被分配的交叉算子分别为文献[13]中列出的BLX- α 、SBX、SPX、PCX和DE。

2) 变异算子。

对于 P_i 的第 j 个解 $X_{ij} = \{x_{ij0}, x_{ij1}, \dots, x_{ijd-1}\}$ 的每一个维度 x_{ijk} ($k = 0, 1, \dots, d - 1$),首先在 $[0, 1]$ 之间产生一个随机数,如果该随机数小于变异概率 p_m ,则根据文献[14]中的多项式变异算子对 x_{ijk} 进行变异操作;否则保持 x_{ijk} 不变。

2.2 多种群竞争协调

在算法的搜索过程中,总的种群规模 N 始终为 $n_{\text{subPop}} \cdot N_i$ (n_{subPop} 表示种群的个数, N_i 表示第 i 个种群的大小)。在初始阶段,每个种群的规模是相等的。种群之间的竞争性体现在每间隔 n_{complete} 次迭代,算法将动态调整每个种群的规模。例如,当 P_i 的规模需要增大时,用分配给 P_i 的交叉算子产生新个体并加入到 P_i 中,其中一个父代来自于当前 P_i 中的个体,剩余所需的父代在外部档案集EXA中随机选择。EXA中的个体可能是由其他种群得到的,因而携带了其他种群的信息,这种父代解的选择方法实现了种群之间信息的共享。算法1给出了多种群竞争协调的过程。种群规模的动态调整公式为

$$N_{\text{new}i} = \frac{n_i}{|\text{EXA}|} N. \quad (15)$$

其中: n_i 为EXA中来自 P_i 的个体数, $|\text{EXA}|$ 为当前EXA中包含的个体数; $N_{\text{new}i}$ 为 P_i 调整后的规模。

算法1 多种群竞争协调过程。

- 1) for $i = 0 : n_{\text{subPop}} - 1$ do
- 2) 按照式(15)计算 P_i 的新规模。
- 3) end for
- 4) 找到新规模中小于 N_{\min} 的种群,将找到的种群的新规模重新设置为 N_{\min} 。
- 5) 找到拥有最大新规模的种群 P_k 。
- 6) 计算 $N_{\text{new}i}$ 的总和,记为total。
- 7) if total < N do
- 8) $N_{\text{new}k} = N_{\text{new}k} + (N - \text{total})$ 。
- 9) else if total > N do
- 10) $N_{\text{new}k} = N_{\text{new}k} - (\text{total} - N)$ 。
- 11) end if
- 12) for $i = 0 : n_{\text{subPop}} - 1$ do
- 13) while 当前规模 N_i 大于新规模 $N_{\text{new}i}$ do
- 14) 在 P_i 中随机选择一个个体删除。
- 15) $N_i = N_i - 1$ 。
- 16) end while
- 17) while 当前规模 N_i 小于新规模 $N_{\text{new}i}$ do
- 18) 在 P_i 中随机选择一个个体并且在EXA中随机选择一个或两个个体作为父代,用 P_i 的交叉算子交叉产生新的个体,并添加到 P_i 。
- 19) $N_i = N_i + 1$ 。
- 20) end while
- 21) end for。

2.3 分类局部搜索

为了进一步提高EXA的质量,首先根据个体在目标空间的距离将EXA分为 n_{subEXA} 个子集 (EXA $_i$)

表示第 i 个子集),在目标空间中相邻的个体被归到同一个子外部档案集中;然后在类内进行局部搜索.在目标空间距离相近的类内进行搜索,能加快算法的收敛速度;同时,目标空间中不同区域的类又能够保证解的分散性.算法2给出了类内局部搜索的过程.

算法2 类内局部搜索.

- 1) for $i = 0 : n_{\text{subEXA}} - 1$ do
- 2) for $j = 0 : |\text{EXA}_i|/n_{\text{subEXA}} - 1$ do
- 3) if EXA_i 中包含的个体数等于1 do
- 4) 用 EXA_i 中的个体复制并产生一个新个体.
- 5) 在 $[0, d - 1]$ 之间随机产生一个整数,将新个体在该数对应的位置处的值设置为在其上下界范围内随机产生的数.
- 6) else
- 7) 根据文献[13]中自适应选择交叉算子的方法选择一个交叉算子.
- 8) 在 EXA_i 中随机选择不同的个体作为父代,根据上一步选择的交叉算子进行交叉操作以产生新个体.
- 9) end if
- 10) 评价新个体.
- 11) 将新个体添加到集合 A .
- 12) end for
- 13) end for
- 14) 用集合 A 中的个体更新 EXA .

2.4 算法流程

在MOEA-CMCC中采用Deb等^[10]提出的方法更新 EXA .另外,设置了 EXA 中的最大规模 n_{EXA} ,一旦 EXA 的规模超过 n_{EXA} ,便删除最拥挤的个体.算法3给出了MOEA-CMCC的整体流程.

算法3 MOEA-CMCC的整体流程.

- 1) 设定迭代次数 $g = 0$;初始化每个种群 $P_i = \{X_{i0}, X_{i1}, \dots, X_{iN_i-1}\}$, $N_i = N/n_{\text{subPop}}$ 为每个种群包含的个体数,并为每个种群分别分配一个不同的交叉算子;将初始种群中的非支配解存于 EXA .
- 2) while 没有达到终止条件 do
- 3) if g 是 n_{class} 的倍数并且 EXA 的个数 > 80 do
- 4) 将当前 EXA 记录于 tempEXA .
- 5) 将 tempEXA 分为 n_{subEXA} 个子外部档案集.
- 6) end if
- 7) 根据算法2进行类内局部搜索.

- 8) if g 是 n_{complete} 的倍数 do
- 9) 根据算法1进行多种群的竞争协调.
- 10) end if
- 11) for $i = 0 : n_{\text{subPop}} - 1$ do
- 12) for $i = 0 : N_i - 1$ do
- 13) 交叉操作:按照当前种群 P_i 被分配的交叉算子,用当前个体 X_{ij} 与在 EXA 中随机选择一个或两个个体作为父代进行交叉操作以产生新个体 X_{new} .
- 14) $X_{ij} = X_{\text{new}}$.
- 15) 按2.1节中的方法对 X_{ij} 进行变异操作.
- 16) 评价 X_{ij}
- 17) end for
- 18) 用 P_i 更新外部档案集 EXA .
- 19) end for
- 20) $g = g + 1$.
- 21) end while
- 22) 输出获得的可行的非支配解.

3 数值实验与分析

3.1 实验环境与参数设置

本文提出的MOEA-CMCC算法使用C++实现,所有实验均是在CPU为Intel Core i5-3550,内存4.0 GB的个人电脑上进行.

MOEA-CMCC算法的相关参数设置如下:各交叉算子的参数采用文献[13]中给出的建议值;种群的个数 $n_{\text{subPop}} = 5$,种群的最小规模 $N_{\text{min}} = 5$,所有种群总的规模 $N = 100$;EXA的规模 $n_{\text{EXA}} = 100$,子EXA的个数 $n_{\text{subEXA}} = 5$;变异概率 $p_m = 1/d$;EXA分类间隔的迭代次数 $n_{\text{class}} = 15$,多种群间竞争协调间隔的迭代次数 $n_{\text{complete}} = 25$.算法的终止条件为达到目标函数的最大计算次数 F_{max} ,对于两目标优化问题,设置 $F_{\text{max}} = 30\,000$;对于三目标和四目标优化问题,设置 $F_{\text{max}} = 50\,000$.

3.2 测试函数与性能评价指标

采用30个经典的Benchmark问题进行对比实验,包括ZDT系列^[15]、LZ09系列^[16]、UF系列^[17]、DTLZ系列^[18].同时选用IGD^[19]指标评价算法的性能,它能够同时评价算法的收敛性和分散性,IGD值越小,表明算法的性能越好.在实验中,对于每个测试问题,每种算法都独立运行30次,使用其统计结果进行评价.

3.3 与代表性进化算法的对比实验

将本文提出的MOEA-CMCC与其有代表性的3种进化算法NSGA-III^[20]、AbYSS^[21]和HMOEA^[13]

进行对比实验.其中,3种算法的实现来源于网站 <http://jmetal.sourceforge.net/>的jMetal5.0软件包,算法中参数的取值为文献中给出的建议值.MOEA-CMCC算法的平均运行时间如表1所示.

表1 MOEA-CMCC的平均运行时间 s

Problems	Times	Problems	Times	Problems	Times
LZ09_F1	1.32	ZDT3	2.39	UF1	1.47
LZ09_F2	0.98	ZDT4	1.97	UF2	2.02
LZ09_F3	1.23	ZDT6	2.48	UF3	1.62
LZ09_F4	0.63	DTLZ1	3.20	UF4	1.90
LZ09_F5	1.27	DTLZ2	35.52	UF5	0.15
LZ09_F7	0.74	DTLZ3	11.91	UF6	0.42
LZ09_F8	0.58	DTLZ4	15.44	UF7	1.85
LZ09_F9	0.99	DTLZ5	6.23	UF8	33.53
ZDT1	2.68	DTLZ6	9.24	UF9	32.50
ZDT2	2.69	DTLZ7	5.12	UF10	31.42

基于实验结果,使用置信水平为95%的独立T检验来评价不同算法之间的性能差异,结果如表2所示.其中:“+”和“-”分别表示MOEA-CMCC显著优于或与显著劣于其他算法,加粗字体为各算法取得的最好解.从结果中可以看出:针对30个测试问题,MOEA-CMCC在14个测试问题上求得了最好的结果;在16个测试问题上显著优于NSGA-III;在21个测试问题上显著优于AbYSS;在13个测试问题上显著优于HMOEA.

图1和图2给出了部分测试函数通过各算法运行30次求得的最小IGD值所对应的Pareto前沿.从中可以看出,本文提出的算法求得的解更加接近真实的Pareto前沿,并且在目标空间中的分布也更加均匀.

表2 不同进化算法的平均IGD结果

Problems	Mean IGD			
	NSGA-III	AbYSS	HMOEA	MOEA-CMCC
LZ09_F1	8.23e-02 ⁺	4.11e-02 ⁺	1.78e-02 ⁺	1.27e-02
LZ09_F2	1.20e-01 ⁺	1.51e-01 ⁺	1.90e-01 ⁺	7.72e-02
LZ09_F3	6.90e-02	8.28e-02	8.60e-02 ⁺	6.69e-02
LZ09_F4	5.72e-02 ⁺	5.34e-02	6.95e-02 ⁺	5.32e-02
LZ09_F5	5.08e-02	6.98e-02 ⁺	6.72e-02 ⁺	5.13e-02
LZ09_F7	2.98e-01	3.53e-01 ⁺	2.47e-01	2.55e-01
LZ09_F8	2.37e-01	3.25e-01 ⁺	2.55e-01	2.57e-01
LZ09_F9	1.51e-01 ⁺	1.37e-01 ⁺	1.24e-01 ⁺	9.22e-02
ZDT1	3.97e-03 ⁺	3.85e-03	3.93e-03 ⁺	3.86e-03
ZDT2	3.94e-03	3.92e-03	3.99e-03	3.98e-03
ZDT3	6.38e-03 ⁺	1.56e-02 ⁺	1.27e-02	4.40e-03
ZDT4	1.21e-02 ⁺	6.56e-03 ⁺	3.86e-03	4.00e-03
ZDT6	7.08e-03 ⁺	3.18e-03	3.17e-03	3.16e-03
DTLZ1	2.16e-02	4.48e-02	1.49e-01	9.54e-02
DTLZ2	5.50e-02	6.94e-02 ⁺	6.75e-02	6.66e-02
DTLZ3	1.26e-01	6.51e-01 ⁺	1.78e+00 ⁺	2.74e-01
DTLZ4	1.21e-01 ⁺	6.19e-02	6.81e-02	6.72e-02
DTLZ5	1.56e-02 ⁺	4.09e-03	4.47e-03 ⁺	4.30e-03
DTLZ6	9.01e-02 ⁺	8.98e-02 ⁺	4.12e-03 ⁺	4.05e-03
DTLZ7	1.11e-01	3.80e-01 ⁺	4.74e-01 ⁺	7.84e-02
UF1	1.26e-01 ⁺	1.65e-01 ⁺	5.43e-02	5.32e-02
UF2	5.27e-02	6.89e-02 ⁺	5.36e-02	5.08e-02
UF3	2.48e-01	2.82e-01 ⁺	2.34e-01	2.42e-01
UF4	5.92e-02 ⁺	6.97e-02 ⁺	5.37e-02 ⁺	5.12e-02
UF5	3.94e-01	6.66e-01 ⁺	4.65e-01	4.22e-01
UF6	3.14e-01	4.07e-01 ⁺	1.63e-01	2.48e-01
UF7	2.44e-01 ⁺	3.10e-01 ⁺	3.69e-02	3.89e-02
UF8	4.05e-01 ⁺	2.59e-01	2.40e-01	2.63e-01
UF9	2.57e-01 ⁺	3.84e-01 ⁺	2.30e-01 ⁺	1.72e-01
UF10	4.28e-01	6.50e-01 ⁺	4.43e-01	3.94e-01

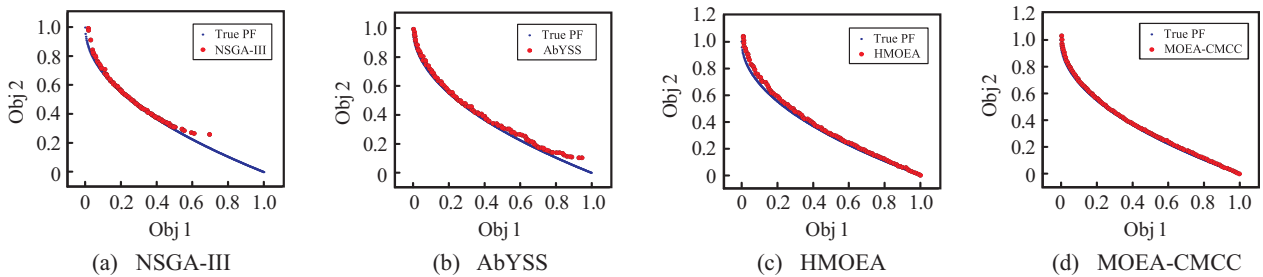


图1 各算法获得的LZ09_F1的Pareto前沿

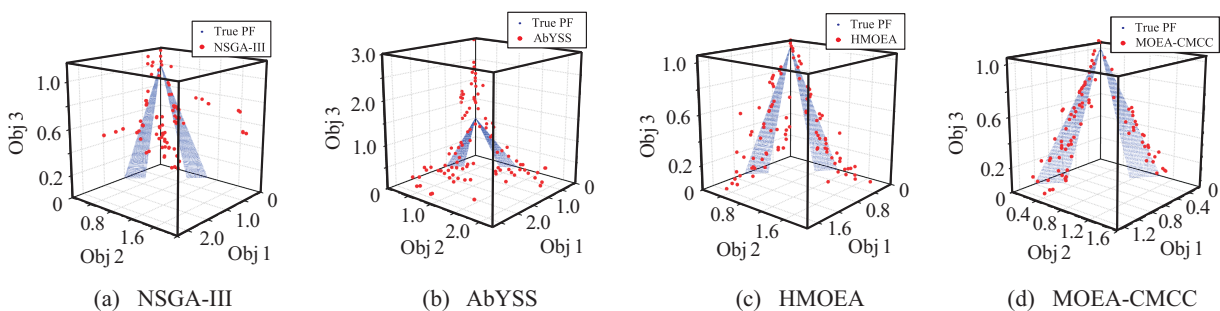


图2 各算法获得的UF9的Pareto前沿

以上的实验结果表明,相对于3种有代表性的多目标进化算法,本文提出的算法能够表现出更好的性能.其原因主要包括:首先,含有不同进化策略的多种群在进化过程中能够保证算法搜索的分散性;然后,竞争协调机制能够将更多的进化资源分配给最适应当前问题的种群,从而在保证算法鲁棒性的同时加快收敛速度;最后,外部档案集的分类与类内搜索不但增强了外部档案集的多样性,也能够帮助加快算法的收敛速度.

3.4 连退多目标操作优化问题的数值实验

连退多目标操作优化问题的优化目标有4个,属于Many-Objective^[20]优化问题,因此,在本实验中只对NSGA-III和MOEA-CMCC进行对比分析.实验数据来自于某钢厂8条带钢的实际生产数据,根据该钢厂的实际情况,取 $\sigma = 5$;连退生产过程工艺约束的上下限值见表3.

表3 连退工艺约束的上下限

<i>i</i>	1	2	3	4	5	6	7	8
a_i	0	-1	100	0	0	0	25	0
b_i	15	4	160	15	10	3	35	1

3.4.1 解的可行性修复策略

由于连退操作优化模型涉及的约束和变量众多且相互关联,传统的Pareto约束支配^[10]方法难以获得问题的可行解.本文提出一种基于启发式方法的可行性修复策略.在算法中,除质量安全约束和生产安全约束的处理遵循Pareto约束支配之外,其他约束的处理均按照所设计的启发式修复策略,其过程如下.

对于加热炉5个区温度约束的处理:若产生的新解不满足约束(3)和约束(4),则先对 $x_1 \sim x_5$ 从小到大进行排序使其满足约束(3);如果排完序的解也满足约束(4),则保留,否则,以 x_1 为基准,加上在 (a_1, b_1) 之间产生的随机数作为 x_5 ,再在 (x_1, x_5) 之间产生3个不重复的随机数,按照从小到大的顺序分别赋给 x_2, x_3, x_4 .

对于其他炉内温度约束的处理:若产生的新解不满足该炉内两个分区温度差的约束,则以两区中编号小的区的温度为基准,加上(减去)在两区温度差之间产生的随机数作为两区中编号大的区的温度.以1#冷炉的2区和3区为例,若新解不满足约束(7),则以2区的温度 x_9 为基准,减去在 $[a_4, b_4]$ 之间产生的随机数作为3区的温度 x_{10} .

3.4.2 实验结果

考虑到实际问题的真实Pareto前沿无法获得,本文将种群规模设置为1000,终止条件目标评价次数

F_{max} 设置为1000万,用MOEA-CMCC和NSGA-III分别对每条带钢数据进行求解,然后从所得到的两个Pareto集合中选取非支配的可行解作为参考Pareto前沿,用来评价算法性能.

表4给出了MOEA-CMCC和NSGA-III在操作优化模型上求得的IGD值的平均值和方差以及结果的差异性,其中检验结果差异性所用的方法和表中“+”,“-”的含义都与3.3节中的相同.从表4中可以看出,MOEA-CMCC在8个实际问题均获得了最好的结果,且都显著优于NSGA-III.

表4 实际问题的IGD结果

带钢	NSGA-III		MOEA-CMCC	
	meanIGD	stdIGD	meanIGD	stdIGD
1	5.15e-01 ⁺	2.59e-01	1.34e-01	4.30e-02
2	6.14e-01 ⁺	2.89e-01	1.44e-01	5.18e-02
3	8.99e-01 ⁺	2.90e-01	1.63e-01	8.39e-02
4	6.73e-01 ⁺	2.80e-01	1.29e-01	7.28e-02
5	6.30e-01 ⁺	2.41e-01	9.47e-02	2.59e-02
6	5.50e-01 ⁺	2.37e-01	8.13e-02	1.03e-02
7	5.06e-01 ⁺	2.63e-01	6.35e-02	3.33e-03
8	2.17e-01 ⁺	1.03e-01	6.28e-02	4.73e-03

表5给出了实际生产结果与操作优化结果在目标函数值上的比较.在实际生产中,操作设定值的选择一般遵循如下偏好:在保证质量的前提下选择使得产能高、能耗低的操作设定值,本文中的操作优化结果是在MOEA-CMCC求得的Pareto前沿中根据该偏好进行选择的.从表5中可以看出,除了在带钢5上实际生产结果的能耗比操作优化结果的能耗低以外,

表5 实际生产结果与操作优化结果的比较

带钢	结果	硬度偏差 f_1	加热炉温度 f_2 (°C)	带钢速度 f_3 (m·min ⁻¹)	退火曲线偏差 f_4
1	实际生产	3.00	682.00	529	37.36
	操作优化	0.45	675.40	578	13.43
2	实际生产	2.00	707.60	500	36.89
	操作优化	1.60	690.69	587	8.311
3	实际生产	3.00	714.40	632	37.38
	操作优化	2.68	713.21	634	3.644
4	实际生产	1.00	726.20	580	37.41
	操作优化	0.69	709.54	608	10.11
5	实际生产	2.00	741.60	649	38.62
	操作优化	1.04	747.44	650	4.68
6	实际生产	2.00	712.20	549	37.38
	操作优化	1.48	709.29	574	4.997
7	实际生产	2.00	728.80	519	37.25
	操作优化	1.50	713.56	524	11.64
8	实际生产	4.00	744.40	500	37.85
	操作优化	3.28	744.37	515	5.877

对于其他带钢,操作优化的结果均显著优于实际生产的结果.由以上结果可以看出,相对于根据传统的人工经验对操作变量进行设定,本文提出的操作优化模型与算法求得的参数设定值能够获得更好的结果.

4 结论

针对冷轧连续退火生产过程中经常出现的薄料带钢跑偏等安全性问题,本文进一步考虑了质量安全约束和生产安全约束,建立了问题的多目标操作优化模型,从而保证了生产的安全性与连续性.同时,针对所建立的模型,提出了一种基于分类与多种群竞争协调机制的多目标进化算法.基于30个Benchmark问题的实验结果表明,该算法的性能优于传统的多目标与高维多目标优化算法.针对实际连退多目标操作优化问题的测试结果表明,本文所提出的算法显著优于传统的高维多目标优化算法NSGA-III,并且操作优化的结果能够在保证生产安全性与连续性的前提下,进一步提高产品质量和机组产能,降低能源消耗.

参考文献(References)

- [1] Chen L, Tang L X, Tang L J. Operation optimization problem of hot rolling based on the transcendental geometric programming[J]. *Control and Decision*, 2015, 30(4): 703-708.
- [2] Hu C B, Tong C N, Peng K X. Load distribution optimization of hot strip mills with improved evolutionary algorithm for complex-process optimization[J]. *Control and Decision*, 2012, 27(1): 15-21.
- [3] Jiang A P, Cheng W, Wang J, et al. Operational optimizations of full flowsheet spiral-wound seawater reverse osmosis system[J]. *CIESC Journal*, 2014, 65(4): 1333-1343.
- [4] Tang Z Y, Li H. Research of Baosteel 1220 mm CAPL tension control system[J]. *Baosteel Technology*, 2016 (2): 25-30.
- [5] Guo C, Zhang Y, You X, et al. Optimal control of continuous annealing process using PSO[C]. *Proc of the IEEE Int Conf on Automation and Logistics*. Shenyang: IEEE, 2009: 602-605.
- [6] Zhang Y, Cao W, Wu M, et al. A hybrid modeling method for continuous annealing line heating process[C]. *Proc of the 35th Chinese Control Conf*. Chengdu: IEEE, 2016: 2263-2267.
- [7] Yang D L, Zhou Y, Liu Y, et al. Data-driven modeling and operation optimization for continuous annealing process[J]. *Baosteel Technology*, 2015(6): 70-75.
- [8] Yang J M, Mu X W, Che H J, et al. Improved multi-objective particle swarm optimization algorithm based on multiple strategies[J]. *Control and Decision*, 2017, 32(3): 435-442.
- [9] Gong D W, Liu Y P, Sun X Y, et al. Parallel many-objective evolutionary optimization using objectives decomposition[J]. *Acta Automatica Sinica*, 2015, 41(8): 1438-1451.
- [10] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2): 182-197.
- [11] Zhang J, Zhou A, Zhang G. A classification and pareto domination based multiobjective evolutionary algorithm[C]. *Proc of the Congress on Evolutionary Computation(CEC-2015)*. Sendai: IEEE, 2015: 2883-2890.
- [12] Lin X, Zhang Q, Kwong S. A decomposition based multiobjective evolutionary algorithm with classification[C]. *Proc of the Congress on Evolutionary Computation(CEC-2016)*. Vancouver: IEEE, 2016: 3292-3299.
- [13] Tang L, Wang X. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems[J]. *IEEE Trans on Evolutionary Computation*, 2013, 17(1): 20-45.
- [14] Eshelman Larry J, Schaffer David J. Real-coded genetic algorithms and interval-schemata[J]. *Foundations of Genetic Algorithms*, 1993, 2: 187-202.
- [15] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. *Evolutionary Computation*, 2000, 8(2): 173-195.
- [16] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(2): 284-302.
- [17] Zhang Q, Zhou A, Zhao S, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition[R]. Colchester: University of Essex, 2008.
- [18] Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multi-objective optimization[J]. *Evolutionary Multi-objective Optimization: Theoretical Advances and Applications*, 2005: 105-145.
- [19] Neri F, Cotta C. Memetic algorithms and memetic computing optimization: A literature review[J]. *Swarm and Evolutionary Computation*, 2012, 2: 1-14.
- [20] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints[J]. *IEEE Trans on Evolutionary Computation*, 2014, 18(4): 577-601.
- [21] Nebro A J, Luna F, Alba E, et al. AbYSS: Adapting scatter search to multiobjective optimization[J]. *IEEE Trans on Evolutionary Computation*, 2008, 12(4): 439-457.