Advances in Social Science, Education and Humanities Research, volume 425
International Conference on Arts, Humanity and Economics, Management (ICAHEM 2019)

Research on Capacity Planning of Cold Rolled Steel Production Line Based on the Modified Differential Evolution Algorithm

Zhengqi Sui[1,a] and Ziyang Yu[1,b,*]

[1]School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, China

[a]1347603495@qq.com, [b]1113161395@qq.com

*corresponding author

Keywords: *differential evolution algorithm, capacity planning of cold rolling production line, multi-objective optimization*

Abstract: The problem of steel cold rolling production capacity is an important issue in steel production. In this paper, mathematical modeling is carried out to meet the actual constraints of production process, inventory, flow balance, etc. And to determine the type and corresponding output of a certain product in each machine in every process in the production network, then obtain the inventory of each product between processes. The objectives of the model are to maximize unit capacity and minimize production switching costs. Differential evolution is a population-based evolutionary algorithm. It has the characteristics of recording the best solution during searching history and exchanging and sharing information within the population. That is, the optimization problem can be solved through the cooperation and competition among individuals within the population. In this paper, the differential evolution algorithm is modified for steel cold rolling production line capacity planning problem, the modified differential evolution algorithm comparing with other multi-objectives genetic algorithm. The results show that the proposed algorithm is superior to other standard genetic algorithm and can fast convergence, thus verified the feasibility and effectiveness of the algorithm.

基于改进差分进化算法的冷轧产线产能计划问题研究

隋政麒 [1,a]，于子洋 [1,b,*]

[1] 信息科学与工程学院，东北大学，沈阳，辽宁，中国

[a]1347603495@qq.com, [b]1113161395@qq.com

*通讯作者

关键词：差分进化算法，冷轧产能计划，多目标优化

摘要：钢铁企业冷轧产线产能计划是钢铁生产过程中的一个重要问题。本文在满足生产工艺、库存、流平衡等实际约束条件的情况下，构建多目标优化模型，决策生产网络结构中每道工序中每台机器每天生产某种产品类型的产量以及工序间的每种类型产品的库存量，以实现最大化机组产能和最小化生产切换费用的目标。差分进化算法是一种基于种群的进化算法，它具有记忆迭代过程中的最优解以及种群内部信息交换共享的特点，利用构造差分向量进行变异和交叉操作来搜索解空间，即通过种群内个体间的合作与竞争来实现对优化问题的求解。本文将差分进化算法进行改进用于求解钢铁企业冷轧产线的产能计划问题，将改进后的差分进化算法与代表性的多目标优化算法进行比较，结果显示改进后的差分进化算法在解决本问题上优化效果明显优于参比算法且能快速收敛，从而验证了本文所提出算法的可行性与有效性。

1. 引言

差分进化算法（Differential Evolution，简称为 DE）是由两位美国学者 Rainer Storn 和 Kenneth Price 为求解切比雪夫多项式，于 1996 年共同提出的一种功能强大的基于种群的随机实参数的智能优化算法[1]。DE 的原理简单，受控参数较少，实施随机、同步、直接的全局搜索，易于理解和实现[2]。

1996 年[3],IEEE 计算智能协会（Computational Intelligence Society）在日本举行了第一届国际进化计算大会（IEEE Congress on Evolutionary Computation，简称为 CEC）。在该次会议上，DE 算法获得了第三名，而第一名和第二名是两个确定性算法（最优化算法），这也意味着 DE 是所有参赛的智能化算法中最快的算法[4]。之后的历届国际进化计算大会，DE 算法屡创佳绩。经过二十年来的发展，DE 已经作为新兴的智能化算法被智能优化领域认可，并得到全世界范围的学者们青睐[5]。随着科技和经济的快速发展，新出现的标准化函数优化问题的复杂度和实际优化问题的求解难度也在不断增加[6],经典的 DE 算法往往不能做到快速收敛和结果最优化，这就需要对算法进行深入研究并加以改进以获得更好的性能。

冷轧是在炼钢、连铸、热轧工序之后，钢铁工业生产的一个重要的生产阶段，产品经退火、精整、涂镀后成为制造业所必需的原材料[7]。一个合理的冷轧产能计划是保证企业生产连续性和产品质量的根本。为保证整条生产线的生产平衡，全流程生产、机组作业调度以及企业订单产生的动态调度都要进行考虑[8-9]。冷轧机组是整个冷轧厂的核心机组，所有的产品生产都需要经过这个机组，其生产能力和生产计划都将直接影响到后续的生产[10],为保证生产线的连续生产与优化，并使钢厂能获得的利润最大化，所以合理安排冷轧产线产能计划就显得尤为重要[11]。为此本文从实际工程角度出发，建立多目标优化模型，通过改进差分进化算法对问题进行求解，最后将结果与代表性的多目标优化算法的结果进行比较。

2. 冷轧产线产能模型

2.1. 问题描述

产能计划是指通过已知的产线结构以及生产产品的类型，计划当月要生产的产品数量[12]。本文是以某钢厂的产线结构为例，共包含六条生产线。产能计划概括来讲就是安排每天每道工序要生产的产品种类及相应的生产量，并同时推算出工序间各类型产品的库存量。

本模型针对多目标函数问题进行研究，主要分为两个目标。第一个目标最小化机组切换次数，第二个目标就是最大化产能。一个机组在生产两个不同类型的产品时需要一次切换。不同产品之间会因为物理形态或者化学成分不同要求同一机组的生产方式不同，比如机组加入调整材或者升高机组温度等操作。这样会降低生产效率还会浪费资源，影响产能。直接的方法是生产完一种产品再生产下一种产品，但是这样会造成后工序机组闲置影响产能。最大化产能就是让所有非检修的机组连续生产。

2.2. 数学模型

2.2.1. 符号说明

表 1 符号说明表

符号	符号含义	符号	符号含义
i	工序编号；	o_{ijt}	第 *i* 个工序的第 *j* 个机组在第 *t* 天的检修时长；
j	机组编号；	e_{ik}	工序 *i* 生产第 *k* 类产品每小时的产量（吨/每小时）；
k	产品类型；	p_{ik}	工序 *i* 生产第 *k* 类产品的成材料（小数）；
In_{it}	初始库存量；	$A_{ii'}$	工序间供料关系，第 *i* 道工序到第 *i'* 工序；
U_i	库存上限；	B_{ik}	工序与产品类型之间关系，第 *k* 类产品在工序 *i* 上生产；
L_i	库存下限；	D	一天里满产时间设定成 24 小时；
t	第 *t* 天；	M	一个大数 100000，用来做约束用；
T	计划期的天数；	$C_{ii'}$	生产的产品工序 *i* 运输到工序 *i'*的周转时间；
R_k	品种产能任务量；	J_i	第 *i* 道工序包含机组个数；

2.2.2. 决策变量

表 2 决策变量表

决策变量	决策变量含义	决策变量	决策变量含义
x_{ijkt}	决策第 *i* 个工序第 *j* 个机组在第 *t* 天生产第 *k* 种类型的产品；	z_{ijt}	工序 *i* 的第 *j* 号机组相邻两天生产不同类型的产品为 1，否则为 0；
l_{ikt}	第 *t* 天第 *k* 种类型在第 *i* 个库存吨数；	m_{ijt}	工序 *i* 的第 *j* 号机组第 *t* 天除检修外是否满产，满产为 0 否则为 1；
y_{ijkt}	第 *i* 个工序的第 *j* 机组在第 *t* 天是否生产产品类型 *k*；	v_{ijt}	第 *i* 个工序的第 *j* 机组在第 *t* 天是否满产，满产为 1 否则为 0；

2.2.3. 目标函数

产能模型中包含以下两个目标函数：

最大化产能：

$$\sum_{i=0}^{I}\sum_{j=0}^{J_i}\sum_{t=0}^{T}\left(D-o_{ijt}-\sum_{k=0}^{K}x_{ijkt}\right) \tag{1}$$

最小化工艺切换次数：

$$\sum_{i=0}^{I}\sum_{j=0}^{J_i}\sum_{t=1}^{T}z_{ijt} \tag{2}$$

2.2.4. 约束条件

产能模型中包含以下十个约束条件：

前后工序库约束：

$$\sum_{t=0}^{T}l_{ii't}\le A_{ii'}\cdot M,\quad i,i'=0,...,I \tag{3}$$

生产供料关系约束：

$$\sum_{j=0}^{J_i}\sum_{t=0}^{T}x_{ijkt}\le B_{ik}\cdot M,\quad i=0,...,I;k=0,...,K \tag{4}$$

产能约束：

$$\sum_{k=0}^{K}x_{ijkt}\le D-o_{ijt},\quad i=0,...,I;j=0,...,J_i;t=0,...,T \tag{5}$$

工艺路径限制约束：

$$\frac{x_{ijkt}}{M}\le y_{ijkt},\quad i=0,...,I;j=0,...,J_i;k=0,...,K;t=0,...,T \tag{6}$$

$$x_{ijkt}\ge y_{ijkt},\quad i=0,...,I;j=0,...,J_i;k=0,...,K;t=0,...,T \tag{7}$$

$$\sum_{k=0}^{K}y_{ijkt}\le 1,\quad i=0,...,I;j=0,...,J_i;t=0,...,T \tag{8}$$

工艺路径切换约束(实际为大类切换)：

$$\left|\sum_{k=0}^{K}k\cdot y_{ijkt}-\sum_{k=0}^{K}k\cdot y_{ijk,t-1}\right|\ge z_{ijt},\quad i=0,...,I;j=0,...,J_i;t=1,...,T \tag{9}$$

$$\frac{\left|\sum_{k=0}^{K}k\cdot y_{ijkt}-\sum_{k=0}^{K}k\cdot y_{ijk,t-1}\right|}{M}\le z_{ijt},\quad i=0,...,I;j=0,...,J_i;t=1,...,T \tag{10}$$

库存约束：

$$l_{ii'0}=In_{ii'},\quad i,i'=0,...,I \tag{11}$$

$$l_{ii't}=l_{ii',t-1}+\left(\sum_{k=0}^{K}A_{ii'}B_{ik}B_{i'k}\left(\sum_{j=0}^{J_i}e_{ik}p_{ik}x_{ijkt}-\sum_{j=0}^{J_{i'}}e_{i'k}p_{i'k}x_{i'jk,t-C_{ii'}}\right)\right), \tag{12}$$

$$i,i'=0,...,I;t=1,...,T$$

$$L_{ii'}\le l_{ii't}\le U_{ii'},\quad i,i'=0,...,I;t=0,...,T \tag{13}$$

产品类型供料平衡：

$$L_{ii'}\le l_{ii't}\le U_{ii'},\quad i,i'=0,...,I;t=0,...,T \tag{14}$$

满产约束：

$$\sum_{k=0}^{K}x_{7jkt}-\left(D-o_{7jt}\right)=0,\quad j=0,...,J_7;t=0,...,T \tag{15}$$

$$\sum_{k=0}^{K}x_{ijkt}-\left(D-o_{ijt}\right)=0,\quad i=3\|5;j=0,...,J_i;t=0,...,T \tag{16}$$

产能任务约束：

$$\sum_{i=3,5,7}\sum_{j=0}^{J_i}\sum_{t=0}^{T} e_{ik}p_{ik}x_{ijkt} \geq r_k, \quad k=1,...,K \tag{17}$$

$$\sum_{i=5,7}\sum_{j=0}^{J_i}\sum_{t=0}^{T} e_{ik}p_{ik}x_{ijkt} = 0, \quad k=3,8,16\sim 21 \tag{18}$$

变量定义：

$$0\leq x_{ijkt}\leq 24,\ \ i=0,...,I;j=0,...,J_i;k=0,...,K;t=0,...,T\ l_{ii't}\geq 0,\ \ i,i'=0,...,I;t=0,...,T$$

$$y_{ijkt},z_{ijt},v_{ijt}\in\{0,1\},\ \ i=0,...,I;j=0,...,J_i;k=0,...,K;t=0,...,T$$

3. 差分进化算法的编码设计与改进

差分进化算法（Differential Evolution，简称 DE）的基本思想如下：种群初始化后，利用从种群中随机选择的两个个体向量的差分量作为第三个随机基准向量的扰动量，从而得到变异向量（Mutant Vector），之后将变异向量与基准向量，又称为目标向量（Target Vector）进行杂交操作，生成试验向量（Trial Vector），最后通过评价函数将基准向量与试验向量进行比较，较优者保存在下一代群体中，形成子代[14]。通过上述操作，DE 逐代改善个体质量，引导种群汇聚到最优解位置。

产能模型的目标函数有两个，故属于多目标问题。多目标问题不会存在一个最优解，也就是说不可能存在唯一一个解在各个目标上都是最优的，而是存在一个解的集合，在集合中存在一个在各个目标上都表现较优的解，这类解集被称为非劣解集[15]。下面提出改进 DE 算法来求解该产能计划问题。

3.1. 编码解码设计

在使用差分进化算法时需要进行编码操作。产能计划需要对各个工序的每个机组进行每一天的计划安排，本问题包含 6 种产品和 6 个工序、每个月 30 天。规定每天每个工序只能生产一种产品类型的物品，所以编码方式采用离散编码向量，即 $x_i=\{x_i^j \mid j=1$,每个工序其中 x_i^j 为正整数。x_i^j 的取值范围是 1 到 6。通过设计先把问题设计成二维向量形式，列向量是同一天各个工序生产的产品类型，每一维是一个产品类型编号，行向量是一个工序这三十天的生产安排。每一维上数值范围是 1 到 6，一共 6 行，每行代表一个工序。

3.2. 种群初始化

设种群规模为 *NP*，可行解空间的维数为 *D*，种群中第 *i* 个个体向量的第 *j* 个参数可以按下式产生：

$$x_{i,j} = rand_j[0,1](U_j - L_j) + L_j \tag{19}$$

其中 $i\leq NP$，$j\leq D$，L_j 和 U_j 分别表示下界和上界，$rand_j$ 返回一个在[0,1]范围内均匀随机数。

3.3. 变异操作与交叉操作改进

经过初始化后，DE 通过对父代种群进行变异和重组操作来产生一个由 *NP* 个实验向量构成的子代种群，对于父代种群中任意一个目标向量 x_i 而言，DE 按如下公式生成变异向量 v_i:

$$v_i = x_{r1} + F(x_{r2} - x_{r3}),\ \ i=1,2,\ldots,NP \tag{20}$$

式中: $\{x_{r1}, x_{r2}, x_{r3}\}$ 是在父代种群中随机选择的三个不同个体，且不同于当前个体 x_i；*F* 为缩放因子。算法通过如下公式生成新的交叉向量 $u_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,j}\}$:

$$u_i = \begin{cases} v_{i,j}, & rand_j \leq CR \text{或} j = Rand_j \\ x_{i,j}, & rand_j > CR \text{或} j \neq Rand_j \end{cases} \tag{21}$$

式中：$rand_j$ 为一个在[0,1]范围内均匀分布的随机数，*CR* 是范围在[0,1]之间的常数，称为

交叉率，*CR* 值越大，子代个体中就有更多的元素是从变异个体中继承的，$Rand_j$ 为[1,D]内的随机整数，它保证 u_i 至少会从 v_i 中继承一个元素，确保了新个体的产生，避免了这一代种群的进化停滞。差分进化算法的算法参数对结果和收敛速度有很大的影响。*F* 较大时，种群在迭代前期可保证较好的多样性，搜索步长，避免陷入局部最优；*F* 值较小时，种群在迭代后期会在优秀个体周围进行局部搜索。*CR* 参数在种群迭代时主要希望把较好个体尽可能多的保留到下一代[15]。

针对产能问题的特性，对 *F* 参数和 *CR* 参数进行改进。改进策略思想为参数自适应选择策略，控制参数在（0,1）范围内进行选择，每经过 G 代的进化操作会对控制参数进行更新。算法流程图如图 3 所示。

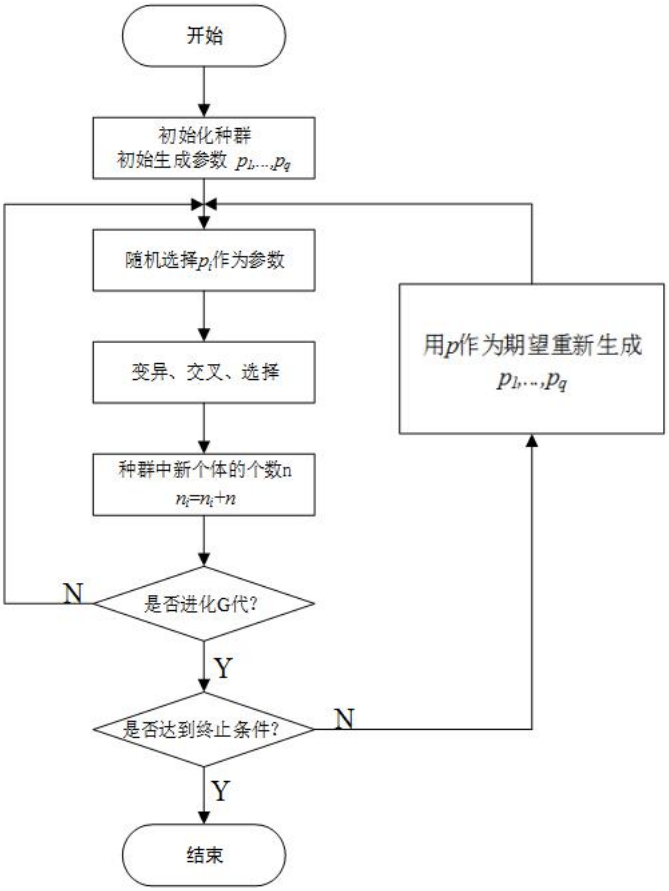

图 3 控制参数更新流程图

在每一代选择操作结束后，统计新的种群中有多少新产生的个体 n，而参数 n_i 是在控制参数 Pi 下求得的多代新个体总和。P 是由下列通式计算得到：

$$P=\sum_{i=1}\frac{n_i}{\sum_{j=1}n_j}P_i,\quad j=1,\ldots,9,i=1,\ldots,9 \tag{22}$$

优秀的参数会在每次进化后使整个新的种群添加更加多的子代，而劣质的参数不会对种群进化起到促进作用。参数 P 在更新时是对 9 个参数进行加权求和，也是对参数整体进行综合评价。用参数 P 替换标准正态分布的均值，用替换了均值的正态分布随机在生成 9 个参数完成对参数的更新，进入下一轮进化。

产能计划的控制参数为变异参数 *F* 和交叉参数 *CR*，按照下式进行更新：

$$F = \sum_{i=1} \frac{n_i}{\sum_{j=1} n_j} F_i, \quad j = 1,...,9, i = 1,...,9 \tag{23}$$

$$CR = \sum_{i=1} \frac{n_i}{\sum_{j=1} n_j} CR_i, \quad j = 1,...,9, i = 1,...,9 \tag{24}$$

每个参数前都有一个权重，代表每个参数在生成新的期望时的重要性，优秀的参数权重值大。而下一代参数的取值则也会偏向于上一代的优秀值，保证了新一代参数值在优秀参数值附近选取。

3.4. 选择操作改进

DE 采用贪婪的搜索方式，产生的新个体 u_i 的评价函数值小于 x_i，u_i 才会被种群接受，对于求最小值的问题：

$$x_i(t+1) = \begin{cases} u_i(t)，f(u_i(t)) \le f(x_i(t)) \\ x_i(t)，其他 \end{cases} \tag{25}$$

式中：f（x）为评价函数，DE 的选择操作是由父代个体与新产生的候选个体之间一对一地进行竞争，优胜劣汰，使子代个体总是优于或等于父代个体，从而使种群向最优解方向进化。

对 DE 的选择策略进行了如下改进，加入了精英再选择策略。精英再选择策略分为三步。第一步计算出需要的适应值较优的精英个体（Pareto）个数，第二步挑选出精英个体，第三步在精英个体周围区间随机生成一些个体，这些新生成的个体适应值相对较高，精英再选择策略可以加强算法的局部寻优能力。

精英个体个数设计考虑到每一次进化迭代精英个体数都是需要改变的，需要在算法进行的不同时期准确的设计精英个体数。在算法初期为了避免因精英再选择策略给算法带来过早收敛到局部最小点的缺点，需要多个精英个体进行局部搜索，但随着算法进行，需要减少精英个体的数目以减少对不必要个体的计算，加快算法计算速度。精英个体个数 EN 由下式确定：

$$EN = EN_{\max} - \frac{E}{G_{\max}} G \tag{26}$$

$G_{\max}$ 为终止条件设定的最大迭代次数，G 为当前迭代次数，E 是最大精英个体个数和最小精英个体个数的差值，$EN_{\max}$ 为设定的最大精英个体数。$G_{\max}$ 需要根据实际要求进行设定，在短时间内对问题求解但是不要求结果是最优解可以把 $G_{\max}$ 设定较小；求解精确度较高，可以把 $G_{\max}$ 设定为较大的值。$EN_{\max}$ 和 E 是根据种群规模决定的，针对本问题 $EN_{\max}$ 设定值为 15，E 的设定值为 10。

针对该多目标问题，挑选出 Pareto 非支配解，在支配解中进行精英个体的选择。如果精英个体个数少于 Pareto 非支配解个数则根据 Pareto 非支配解分布均匀的取出精英个体，增加种群多样性。精英个体数目多时，先选出全部的 Pareto 非支配解，之后再在种群中去除 Pareto 非支配解代表的个体，最后在求取 Pareto 非支配解，直到选完所需的精英个体。在精英个体周围进行再选择精英时,针对上述需求拟采用正态分布函数对精英个体再选择：

$$n_i^j = x_i^j + Gaussion(0,1), \quad i = 1,...,NP; j = 1,...,D \tag{27}$$

式中 x_j^i 为原精英个体向量，n_j^i 为新生成的精英个体。

3.5. 基于 k 均值的种群分类

针对这种大规模优化问题，在外部档案集加入 k 均值分类操作，找到分类中心点。采用高斯分布以中心点为中心生成新的个体。用新的个体和外部档案集做支配关系比较，如果是非支配关系则把新生成的个体加入外部档案集。聚类操作的目标是把具有相似特征的采样数据聚到一起，同时把不相似的个体分开。k 均值算法的原理就是通过计算每个样本之间的距离，把同一类样本分在一起，把不同类的样本分到不同类别当中。

k 均值算法计算时主要分为两个步骤，第一步数据分配，第二步质心更新。数据分配是定义 k 个质心，每个样本到 k 个质心的距离都计算一遍找出距离最近的质心，把该样本分配给该质心，最后每个质心都代表一类。本文距离计算公式采用欧几里得距离，计算公式如下：

$$d(a,b)=\sqrt{\sum_{i=1}^{n}\left(a_i^2-b_i^2\right)} \tag{28}$$

$$c_i=\frac{1}{\left|S_j\right|}\sum_{x_i\in S_j}x_i \tag{29}$$

上式为样本 a 和样本 b 的欧几里得距离，两个样本都是 n 维向量；式(29)中 C_i 为新计算出的质心，S_i 为第 j 类的集合，x_i 为 j 类中的样本。通过 k 均值算法对外部档案集进行分类，在合理的位置生成新的精英个体增加外部档案集规模。

4. 数值实验

为验证改进差分进化算法的有效性与高效性，本节根据验证目标的不同设计了两类对比实验。首先将改进差分进化算法与经典差分进化算法进行对比，用于验证改进算法基于参数、精英再选择这两个策略的性能。然后将改进的多目标差分进化算法与经典的多目标算法进行对比，目的为验证改进差分进化算法的工程应用特性。

4.1. 实验测试指标

本节将采用三种指标来评价多目标优化算法的改进性能，它们分别是收敛性指标γ、多样性指标Δ和支配性指标Ω。

收敛性指标γ由下式计算得出，实际意义是求取每个非劣解到帕累托最优解的最近距离的均值。均值越小越好，证明算法收敛性越好。其中 d_i^* 表示每个非劣解 i 到帕累托最优解集的最近距离，最近距离用欧氏距离计算，N 为算法求得非劣解的个数。

$$\gamma = \frac{1}{N}\sum_{i=1}^{N} d_i^* \tag{30}$$

多样性指标的表达式为

$$\Delta=\frac{d_f+d_l+\sum_{i=1}^{N-1}\left|d_i-\overline{d}\right|}{d_f+d_l+(N-1)\cdot\overline{d}} \tag{31}$$

其中 d_f 为帕累托最优解集的上、下边界解，d_l 为非劣解上、下边界之间的欧式距离，d_i 为两个非劣解 i 与 i+1 之间的欧式距离，$\overline{d}$ 表示所有 d_i 求和的均值。Δ越大，表示非劣解集多样性越好。通过计算非劣解和非劣解均值的差的绝对值来近似代表非劣解的多样性。

支配性指标Ω由下式计算，其中 $P(X)$ 函数表示的是选择集合 X 中的非劣解，进而得到非劣解集。而集合 O 和集合 H 分别表示问题已知帕累托最优解集和算法求解得到的非劣解集。

$$\Omega=\frac{\left|P(O\cap H)\right|}{\left|P(O\cup H)\right|}\times 100\% \tag{32}$$

由于帕累托最优解集无法直接得到，这里对参加对比的算法所获得的非劣解集进行支配性比较来构造帕累托最优解集。

4.2. 改进策略性能测试

本节将对各个改进策略的有效性进行测试，将改进算法与不同的未改进算法进行对比：

(1) MKP，不带控制参数改进策略

(2) MKS，不带精英再选择策略

(3) MKK，不带 k 均值算法

采用上述三种算法和 MK 算法进行求解，得到非支配解后对收敛性指标、多样性指标和支配性指标值进行统计。表 4 对每个月进行 15 次计算求取每个指标的均值 Mean 和标准差 Std。

表 4 MK 与三种变体的多目标性能指标比较

月份	指标	MK		MKP		MKS		MKK	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	γ	43.50	14.03	59.62	23.92	48.65	19.57	46.20	20.30
	Δ	0.83	0.24	0.71	0.38	0.69	0.26	0.77	0.23
	Ω	15.95	7.69	9.37	6.49	3.35	8.10	9.56	7.31
2	γ	37.38	15.68	53.86	14.27	46.15	16.93	45.62	15.73
	Δ	0.75	0.35	0.65	0.30	0.70	0.27	0.69	0.23
	Ω	17.80	7.06	16.07	8.42	12.83	8.40	13.84	8.36
3	γ	47.89	19.28	55.82	19.44	49.53	22.90	53.89	20.48
	Δ	0.84	0.24	0.80	0.29	0.75	0.22	0.78	0.25
	Ω	23.21	5.76	15.40	5.90	12.61	6.69	18.64	5.76
4	γ	34.01	24.03	44.62	22.74	41.41	21.87	50.63	23.03
	Δ	0.85	0.31	0.78	0.27	0.69	0.18	0.77	0.26
	Ω	24.13	7.31	18.93	6.85	14.35	8.40	17.15	8.53
5	γ	37.53	17.68	53.41	16.57	46.63	19.42	45.91	17.52
	Δ	0.72	0.23	0.64	0.18	0.68	0.21	0.70	0.25
	Ω	16.80	13.26	12.97	10.18	8.13	12.07	14.70	8.26
6	γ	46.79	19.57	57.75	16.65	49.58	21.64	51.46	19.12
	Δ	0.88	0.28	0.74	0.36	0.75	0.29	0.82	0.28
	Ω	28.56	7.58	23.35	6.52	18.54	6.25	24.33	5.96
7	γ	47.52	21.53	55.75	27.35	50.26	23.87	53.54	25.28
	Δ	0.67	0.23	0.59	0.24	0.55	0.23	0.61	0.28
	Ω	22.45	8.35	19.87	8.67	13.13	8.56	21.78	8.64
8	γ	52.85	16.35	63.27	19.12	56.57	16.58	59.38	15.84
	Δ	0.77	0.23	0.70	0.24	0.68	0.18	0.70	0.25
	Ω	14.58	8.07	10.97	7.62	8.83	8.02	12.80	8.05
9	γ	43.55	19.78	64.32	23.64	57.19	19.43	60.58	20.46
	Δ	0.83	0.25	0.69	0.29	0.74	0.35	0.74	0.28
	Ω	23.54	5.86	14.78	6.46	11.65	6.54	19.87	6.53
10	γ	43.54	25.78	58.35	18.56	71.28	28.83	67.58	22.29
	Δ	0.82	0.23	0.69	0.29	0.72	0.27	0.78	0.26
	Ω	30.86	7.45	15.75	8.34	23.86	9.73	21.58	7.76
11	γ	47.43	25.28	68.64	24.57	57.67	28.57	50.38	30.68
	Δ	0.86	0.35	0.73	0.29	0.76	0.25	0.81	0.29
	Ω	24.80	8.06	12.97	8.10	18.83	8.00	19.30	8.74
12	γ	38.89	19.48	58.22	21.84	49.15	23.90	43.89	22.41
	Δ	0.83	0.28	0.73	0.31	0.76	0.32	0.79	0.28
	Ω	18.21	6.76	10.43	6.90	13.61	6.09	15.21	6.76

利用下式计算单个性能指标的改进量，式中 Index1 和 Index2 分别表示两个不同算法得到的性能指标。Index1 表示的是 MK 算法的各指标的平均值，Index2 是 MKP、MKS、MKK 算法的各性能指标的平均值。

$$provement = \frac{Index_1 - Index_2}{|Index_1|} \times 100\% \tag{33}$$

计算结果如表 5、表 6、表 7 所示。

表 5 MK 相对三种变体的γ性能指标改进量(%)

月份	MKP	MKS	MKK
1	37.06	11.84	6.21
2	44.09	23.46	22.04
3	16.56	3.42	12.53
4	31.20	21.76	48.87
5	42.31	24.25	22.33
6	23.42	5.96	9.98
7	17.32	5.77	12.67
8	19.72	7.04	12.36
9	47.69	31.32	39.10
10	34.01	63.71	55.21
11	44.72	21.59	6.22
12	49.70	26.38	12.86

表 6 MK 相对三种变体的Δ性能指标改进量(%)

月份	MKP	MKS	MKK
1	17.44	19.77	10.47
2	13.33	6.67	8.00
3	4.76	10.71	7.14
4	8.24	18.82	9.41
5	11.11	5.56	2.78
6	15.91	14.77	6.82
7	11.94	17.91	8.96
8	9.09	11.69	9.09
9	16.87	10.84	10.84
10	15.85	12.20	4.88
11	15.12	11.63	5.81
12	12.05	8.43	4.82

表 7 MK 相对三种变体的Ω性能指标改进量(%)

月份	MKP	MKS	MKK
1	41.25	78.99	40.06
2	9.71	27.92	22.27
3	33.69	45.67	19.90
4	21.50	40.53	28.27
5	22.78	51.60	12.00
6	18.22	35.08	14.81
7	11.42	41.51	2.98
8	24.60	39.43	12.20
9	37.13	50.51	15.90
10	48.63	22.68	30.71
11	47.02	24.07	22.17
12	42.24	25.26	16.47

从表格里可以看出， MK 算法和三种变体算法相比收敛性指标γ的最大改进量为 63.71%，多样性指标的最大改进量出现在 MK 算法和 MKS 算法对比上改进量为 19.77%，支配性指标的最大改进量为 51.60%。这体现出三种改进策略对算法各个指标都有所改进。

4.3. 算法性能测试

为测试MK算法的整体性能，将会和一个经典的多目标优化算法NSGA-II做对比。NSGA-II算法的参数配置按照原文献中建议值进行设置。

NSGA-II(带有精英策略的非支配排序的遗传算法)算法的基本思想。首先通过父代种群 Pt 种群大小为 NP，产生自带种群 Qt，并将两个种群联合在一起形成大小为 2NP 的种群 Rt。之后通过快速非支配解进行排序并且对排序后的种群进行分层，同时对每个非支配层中的个体进行拥挤度计算。最后根据非支配关系和拥挤度选取出合适的 NP 个个体组成新的种群。其中快速非支配解排序和拥挤度计算是本论文的创新点，快速非支配解计算加速了精英个体的计算而拥挤度计算增强了精英个体的多样性。重复上面三步操作，直到算法达到终止条件。

通过计算 12 个月的产能计划，本论文对 MK 算法和 NSGA-II 进行了综合对比，计算各自的收敛性指标、多样性指标和支配性指标值。如下表 8。

表 8 MK 算法与 NSGA-II 算法多目标性能指标比较

月份	指标	MK		NSGA-II	
		Mean	Std.	Mean	Std.
1	γ	43.50	14.03	46.79	24.51
	Δ	0.83	0.24	0.61	0.17
	Ω	15.95	7.69	9.38	7.14
2	γ	37.38	15.68	51.64	12.74
	Δ	0.75	0.35	0.68	0.28
	Ω	17.80	7.06	12.47	8.36
3	γ	37.89	19.28	50.69	20.48
	Δ	0.84	0.24	0.77	0.19
	Ω	23.21	5.76	15.26	6.14
4	γ	34.01	24.03	53.13	26.52
	Δ	0.85	0.31	0.79	0.21
	Ω	24.13	7.31	19.65	6.81
5	γ	37.53	17.68	49.86	19.37
	Δ	0.72	0.23	0.68	0.21
	Ω	16.80	13.26	9.37	7.63
6	γ	46.79	19.57	57.83	21.83
	Δ	0.88	0.28	0.81	0.35
	Ω	28.56	7.58	21.63	8.73
7	γ	47.52	21.53	60.81	23.78
	Δ	0.67	0.23	0.63	0.20
	Ω	22.45	8.35	17.28	7.64
8	γ	52.85	16.35	65.85	18.47
	Δ	0.77	0.23	0.65	0.31
	Ω	14.58	8.07	6.17	8.75
9	γ	43.55	19.78	54.61	21.37
	Δ	0.83	0.25	0.76	0.20
	Ω	23.54	5.86	17.63	6.82
10	γ	43.54	25.78	61.52	26.63
	Δ	0.82	0.23	0.68	0.28
	Ω	30.86	7.45	18.61	7.52
11	γ	47.43	25.28	62.25	30.28
	Δ	0.86	0.35	0.81	0.29
	Ω	24.80	8.06	16.29	7.54
12	γ	38.89	19.48	50.81	23.83
	Δ	0.83	0.28	0.69	0.25
	Ω	18.21	6.76	12.72	7.28

通过上表 8 可以看出 MK 算法在整体性能上还是优于 NSGA-II 的，下面通过表 9 对具体改进量进行统计。

表 9 MK 算法相对 NSGA-II 算法性能指标改进量(%)

月份	γ	Δ	Ω
1	7.56	29.07	41.19
2	38.15	9.33	29.94
3	5.85	8.33	34.25
4	56.22	7.06	18.56
5	32.85	5.56	44.22
6	23.59	7.95	24.26
7	27.97	5.97	23.02
8	13.25	15.58	57.68
9	25.40	8.43	25.10
10	41.30	17.07	39.69
11	31.25	5.81	34.31
12	30.65	16.87	30.14

由表 9 可知针对产能计划问题 MK 算法相比于 NSGA-II 算法在收敛性指标、多样性指标和支配性指标都有较大提高。由表 9 收敛性γ绘制折线图 10、多样性指标Δ绘制折线图 11、支配性指标Ω绘制折线图 12。

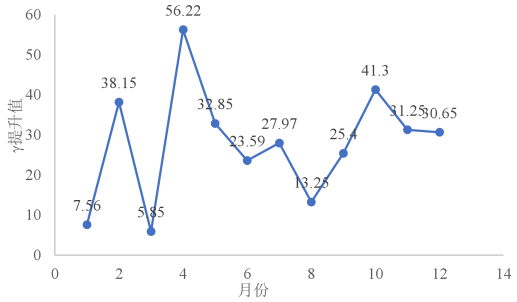

图 10 收敛性γ改进量

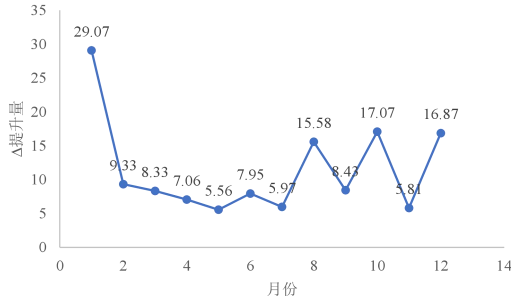

图 11 多样性指标Δ改进量

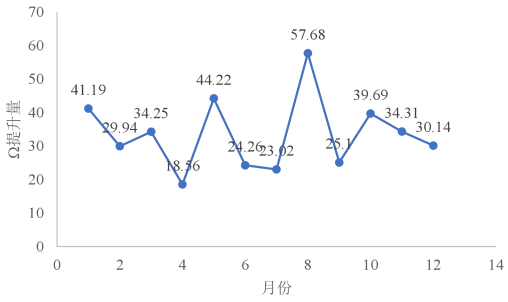

图 12 支配性指标Ω改进量

由图 10、图 11 和图 12 可以清晰地看出收敛性指标提升最大值为 56.22%、最小值为 5.85%，多样性指标最大提升为 29.07%、最小值为 5.56%，而支配性指标最大提升为 57.68%、最小提升 18.56%。综上可以得出 MK 算法在三个多目标判断指标上都优于 NSGA-II 算法。

5. 结论

为解决冷轧产线产能计划问题，本文采用差分进化算法对产能计划问题进行求解，但是为了适应实际问题，在经典的差分进化算法的基础上对控制参数、种群选择策略进行了改进。针对多目标本文加入了外部档案集，通过 k 均值分类算法对外部档案集进行分类，加速了多目标差分进化算法的计算速度，并且减少了计算资源。使用改进的多目标差分进化算法和经典的多目标差分进化算法、多目标优化问题经典算法 NSGA-II 基于收敛性、支配性、多样性三个指标进行对比。结果证明本文所采用的改进策略极大地增强了算法的寻优性能。

参考文献

[1] Storn R, Price K. Differential Evolution : a simple and efficient adaptive scheme for global optimization over continuous spaces[R]. Tech. Rep. TR-95-012,ICSI,USA,1995
[2] 吴志峰，差异演化算法及应用进展[D]. 北京：北京交通大学，2009
[3] Qing A Y. Differential Evolution: Fundamentals and Applications in Electrical Engineering[M], Singapore:Wiley-IEEE Press, 2009
[4] Price K, Storn R, Lampinen J. Differential Evolution: Apritical Approach to Global Optimization[M],Berlin, Germany: Springer, 2005
[5] 董赟．差分进化算法研究及在港口物流调度中的应用[D].东北大学，2015
[6] Das S,Abraham A,Chakraborty U K, Konar A.Differential evolution using a neighborhood based mutation operator[J],IEEE Transactions on Evolutionary Computation,2009,13(3):526- 533
[7] 郑秉霖，胡琨元，常春光.一体化钢铁生产计划系统的研究现状与展望. 控制工程，2003,10（1）：6-10
[8] P.Loos，T.Allweyer. Application of production planning and scheduling in the process industries. Computers in Industry 36(1998): 199-208
[9] F.A.Rodammer, K.P.White. A recent survey of production scheduling .IEEE TRANS.ON SYSTEMS, MAN. AND CYBERNETICS.6(19 -88):841-851
[10] 宋执环，高春华，李平.流程工业 CIMS 若干问题探讨.系统工程理论与实践. 2001，No.5：50-55
[11] 蔡之华，龚文引. 差分演化算法及其应用[M]. 武汉：中国地质大学出版社，2010
[12] 苗月. 基于数据驱动的钢铁冷轧库存问题建模及软件系统[D]. 2014.
[13] 佚名. 一种钢铁企业冷轧区多产线钢卷协调调度方法:, CN 104376424 B[P]. 2017.
[14] 王凌，钱斌. 混合差分进化与调度算法[M]. 北 京：清华大学出版社，2012
[15] Piotrowski A P. Differential Evolution algorithms applied to Neural Network training suffer from stagnation[J]. Applied Soft Computing, 2014,21:382-406.