

# 控制与决策

Control and Decision

## 基于复杂昂贵仿真的体系效能多目标优化

林圣琳, 李伟, 钱晓超, 马萍, 杨明

引用本文:

林圣琳, 李伟, 钱晓超, 等. 基于复杂昂贵仿真的体系效能多目标优化[J]. *控制与决策*, 2021, 36(3): 589–598.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0844>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于分解的多目标多因子进化算法

A multiobjective multifactorial evolutionary algorithm based on decomposition

*控制与决策*. 2021, 36(3): 637–644 <https://doi.org/10.13195/j.kzyjc.2019.0525>

### 基于知识粒度特征的多目标粗糙集属性约简算法

Multi objective rough set attribute reduction algorithm based on characteristics of knowledge granularity

*控制与决策*. 2021, 36(1): 196–205 <https://doi.org/10.13195/j.kzyjc.2019.0490>

### 基于改进多目标优化算法的分布式数据中心负载调度

Multi-objective optimization of energy and performance management in distributed data centers

*控制与决策*. 2021, 36(1): 159–165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

### 基于稀疏度阶数优化的杂波密度估计算法

A clutter density estimation algorithm by optimized sparsity order

*控制与决策*. 2020, 35(12): 2923–2930 <https://doi.org/10.13195/j.kzyjc.2019.0429>

### 基于树形结构无界存档的多目标粒子群算法

Multi-objective particle swarm optimization algorithm based on tree-structured unbounded archive

*控制与决策*. 2020, 35(11): 2675–2686 <https://doi.org/10.13195/j.kzyjc.2019.0276>

# 基于复杂昂贵仿真的体系效能多目标优化

林圣琳<sup>1</sup>, 李伟<sup>1</sup>, 钱晓超<sup>2</sup>, 马萍<sup>1†</sup>, 杨明<sup>1</sup>

(1. 哈尔滨工业大学 控制与仿真中心, 哈尔滨 150080; 2. 上海机电工程研究所, 上海 201109)

**摘要:** 效能优化是实现体系结构设计、多方案配置等工作的重要途径. 体系仿真系统通常具有组成结构复杂、连续离散混合、输入输出变量多、运行开销大等特点, 导致效能优化面临多目标、混合变量、多峰值、低效率等问题, 提出一种基于复杂昂贵仿真的体系效能优化算法. 针对昂贵仿真问题, 提出基于聚类与空间填充准则相结合的开发-探索序贯元模型策略; 引入混合编码策略和相应遗传算子改进传统的 NSGA-II (nondominated sorting genetic algorithm II) 算法, 提高多目标优化效率. 与目前较流行的 3 种昂贵优化算法的对比分析表明, 所提算法得到的优化结果具有较好的收敛性、多样性及分布性. 最后, 以某防空体系的效能仿真优化为例, 验证所提方法的可行性和有效性.

**关键词:** 复杂昂贵仿真; 体系效能优化; 序贯元模型; 混合编码; 多目标优化; 开发-探索策略

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0844

开放科学(资源服务)标识码(OSID):



**引用格式:** 林圣琳, 李伟, 钱晓超, 等. 基于复杂昂贵仿真的体系效能多目标优化[J]. 控制与决策, 2021, 36(3): 589-598.

## Complex and expensive simulation based multi-objective optimization to system-of-system effectiveness

LIN Sheng-lin<sup>1</sup>, LI Wei<sup>1</sup>, QIAN Xiao-chao<sup>2</sup>, MA Ping<sup>1†</sup>, YANG Ming<sup>1</sup>

(1. Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China; 2. Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China)

**Abstract:** Effectiveness optimization is the important approach to design system-of-system structure and configure scenario schemes. The system-of-system simulation system is usually characterized by complex structure, continuous and discrete mixing, multiple input and output variables, and high operation cost, which leads to some problems of multi-objective, mixed variables, multi-peak and low efficiency, etc. existed in effectiveness optimization. A complex and expensive simulation-based approach to system-of-system effectiveness optimization is proposed. The exploitation-exploration based sequential metamodel strategy is presented coupling clustering with space filling criterion for expensive simulation. The hybrid coding strategy and the corresponding genetic operators are introduced to improve the multiobjective optimization efficiency of the traditional NSGA-II algorithm. The comparative analysis with the state-of-art compensive optimization algorithms shows that optimization results of the proposed algorithm have better convergence, diversity and distribution. Futhermore, the effectiveness simulation optimization for some air defense system-of-system is taken as an example to verify the feasibility and effectiveness of the proposed method.

**Keywords:** complex and expensive simulation; effectiveness optimization of system-of-system; sequential meta-model; mixed coding; multiobjective optimization; exploitation and exploration strategy

## 0 引言

体系效能是指在特定的条件下, 利用装备体系执行作战任务所能达到预定目标的程度. 复杂装备体系的作战效能通常受到装备性能、装备数量、作战部署、作战方案以及环境等因素的影响, 效能优化对体

系结构设计、想定编辑和装备贡献度评估具有重要意义. 基于仿真的体系效能优化方法因具有安全性、经济性、高效性等特点而得到了广泛应用. 体系仿真系统具有结构复杂、连续离散混合、非线性程度高、输入输出变量多、运行开销大等特点, 给体系效能优

收稿日期: 2019-06-12; 修回日期: 2019-09-09.

基金项目: 国家自然科学基金项目(61627810).

责任编辑: 林崇.

<sup>†</sup>通讯作者. E-mail: pingma@hit.edu.cn.

化带来困难.具有多目标、多峰值、连续离散变量混合特点的体系效能仿真快速优化问题亟待解决.

目前,国内外针对装备体系结构与效能优化问题的研究取得了一定进展,Davendralingam等<sup>[1]</sup>提出了一种基于金融工程与运筹学方法相结合的体系结构鲁棒优化方法;Gu等<sup>[2]</sup>构建了武器系统的费用-效能模型,并引入连续分段线性元模型实现武器系统的效能优化;Liao等<sup>[3]</sup>将复杂系统性能优化问题转化为混合变量参数优化问题,并提出了一种基于仿真的蚁群优化算法实现系统参数配置;Romich等<sup>[4]</sup>针对传感器覆盖和通信问题,建立了解析模型并选取通信效能度量指标,采用逼近方法和割平面算法进行了优化;程贲等<sup>[5]</sup>分析了当前7种主要的体系仿真优化方法,并提出了亟待解决的问题;钱晓超等<sup>[6]</sup>提出了基于效能评估的装备体系优化设计方法.

面向昂贵计算的多目标优化方法为体系效能仿真优化问题提供了解决途径.引入代理模型对昂贵计算目标进行近似,利用多目标优化算法在元模型上进行迭代优化<sup>[7]</sup>,得到Pareto前沿.根据待优化问题特点,可将多目标昂贵计算优化问题<sup>[8-9]</sup>分为:无约束连续变量优化、无约束离散变量优化、无约束混合变量优化、带约束的连续变量优化、带约束的离散变量优化、带约束的混合变量优化.其中,带约束的优化问题<sup>[10-11]</sup>可分为:等式约束优化、不等式约束优化、黑箱约束优化.涉及的元模型方法<sup>[12-14]</sup>包括:高斯过程元模型、径向基函数(radial basis functions, RBFs)元模型、神经网络、支持向量机等.

基于元模型的优化算法性能优劣由2个因素决定:1)元模型的拟合精度,包括元模型类型和序贯实验点的选取;2)多目标进化算法的性能.文献<sup>[15]</sup>证明了径向基元模型较其他类元模型具有较好的拟合效果,故本文选取RBF元模型作为效能优化的代理模型.需要说明的是,针对体系效能优化问题的研究侧重于多目标、多峰值、连续离散变量混合优化方面.体系仿真系统通常由多个异类子系统构成且含有多个决策变量,其约束条件多数以范围约束为主,故本文针对体系效能优化问题特点研究相应的优化算法.

为提高体系效能仿真优化的精确度,考虑多目标、多峰值、混合变量、运行开销大等特点,提出一种基于复杂昂贵仿真的体系效能优化算法(complex and expensive simulation based system-of-system effectiveness optimization, CES-SEO).引入RBF元模型对体系仿真系统进行近似,并采取基于聚类分析与空间填充准则相结合的开发-探索序贯策略提高

元模型拟合精度;同时,为解决体系效能优化中混合变量、多峰值、非线性等问题,在传统多目标进化算法框架的基础上,引入混合编码策略和相应遗传算子,提高多目标优化效率.将本文算法与ParEGO(pareto efficient global optimization)算法、文献<sup>[13]</sup>的GOMORS(gap optimized multi-objective optimization using response surfaces)算法和文献<sup>[11]</sup>的GLOSADE(global and local surrogate-assisted differential evolution)算法对比分析,以验证本文算法具有良好的性能,并以某防空体系效能仿真优化为例验证本文方法的可行性和有效性.

## 1 体系效能仿真优化问题

体系效能仿真优化可抽象为带有连续离散混合变量的多目标优化问题.为明确效能优化问题及其特点,给出如下定义.

**定义1**(体系效能多目标优化问题) 令  $F: \mathbf{R}^{k_1} \times \mathbf{Z}^{k_2} \mapsto \mathbf{R}^m$  代表具有非线性、多峰值特点的效能优化目标函数.其中:  $k_1$ 、 $k_2$  分别为连续、离散决策变量个数,  $m$  为待优化目标数.定义决策变量  $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{u}^T)$ , 决策变量个数  $k = k_1 + k_2$ ,  $\mathbf{x} \in \mathbf{R}^{k_1}$  代表连续变量,  $\mathbf{u} \in \mathbf{Z}^{k_2}$  代表离散变量.给出体系效能优化问题的数学描述如下:

$$\min F(\mathbf{z}) = (f_1(\mathbf{z}), f_2(\mathbf{z}), \dots, f_m(\mathbf{z})). \quad (1)$$

$$\text{s.t. } -\infty < x_{i_1}^l \leq x_{i_1} \leq x_{i_1}^u < \infty, \forall i_1 = 1, 2, \dots, k_1;$$

$$-\infty < u_{i_2}^l \leq u_{i_2} \leq u_{i_2}^u < \infty,$$

$$\forall i_2 = k_1 + 1, k_1 + 2, \dots, k;$$

$$\mathbf{x} \in \mathbf{R}^{k_1}, \mathbf{u} \in \mathbf{Z}^{k_2}, \mathbf{z}^T = (\mathbf{x}^T, \mathbf{u}^T).$$

其中:  $F(\mathbf{z})$  为目标函数集,包含多个效能指标和费用;  $\mathbf{z}$  为装备体系及其各装备系统的技战术性能参数;  $x_{i_1}^l$  和  $x_{i_1}^u$  为连续变量  $x_{i_1}$  ( $i_1 = 1, 2, \dots, k_1$ ) 的取值范围;  $u_{i_2}^l$  和  $u_{i_2}^u$  为离散变量  $u_{i_2}$  ( $i_2 = k_1 + 1, k_1 + 2, \dots, k$ ) 的取值范围.

基于体系效能多目标优化问题定义,结合文献<sup>[16]</sup>能够得到Pareto支配、Pareto最优解集以及Pareto前沿(pareto front, PF)的相关概念.在此基础上,总结体系效能多目标优化问题是获取一组在目标空间尽量靠近且覆盖真实Pareto前沿的解集,进而支撑体系结构设计及多方案配置等工作.

## 2 CES-SEO算法原理

基于昂贵体系仿真的CES-SEO算法采用序贯策略构建/更新元模型,用于近似装备体系性能参数与

多个效能或费用指标间的函数关系. 在每次算法迭代过程中利用全局进化算法和开发-探索策略选取序贯实验点,直到满足算法终止条件,得到最终 Pareto 最优解,如图1所示.

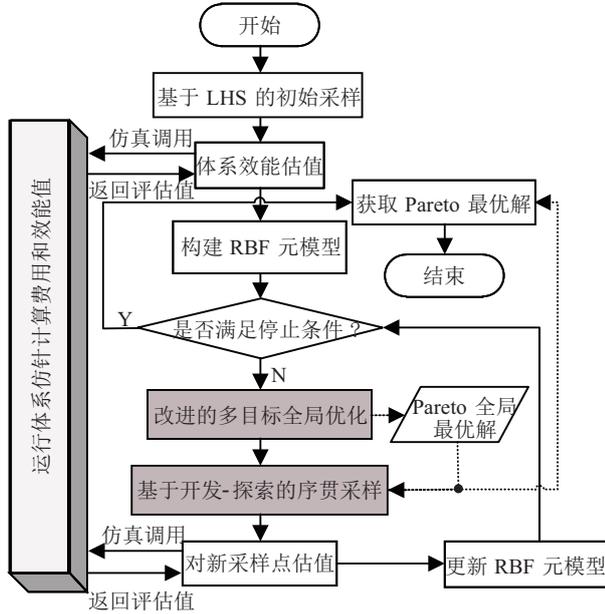


图1 CES-SEO算法框架

图1中,算法开始阶段需要初始化相关参数,包括初始样本数、昂贵仿真最大运行次数、多目标进化种群个体数以及进化代数等. 引入拉丁超立方 (latin hypercube sampling, LHS) 抽样方法<sup>[17]</sup>: 获取初始样本点,运行昂贵体系仿真并计算多个效能指标及费用值. 利用上述样本集构建元模型,若未达到最大昂贵仿真次数,则利用提出的多目标全局优化算法获取当前元模型的最优解. 接下来,采用基于空间填充性与聚类相结合的开发-探索策略从全局最优中选取序贯采样点,并运行昂贵体系仿真获取新的样本值. 将新样本点加入到原有样本集中进行元模型更新. 若满足停止条件,则提取当前 Pareto 最优解集获取最优费用和效能指标值;否则继续进行序贯优化.

### 2.1 径向基函数元模型

径向基函数方法是通过对多个径向基函数的线性叠加得到的,其中径向基函数是以径向距离为自变量的函数,数学模型如下:

$$s(\mathbf{z}) = \sum_{i=1}^n \omega_i \varphi_i(r_i). \quad (2)$$

其中: $n$ 为训练样本点个数; $\omega_i$ 为每个基函数的权系数; $\varphi_i(r_i)$ 为径向基函数; $r_i$ 为某样本点 $\mathbf{z}$ 距离中心点 $\mathbf{z}_i$ 的距离,即

$$r_i = \|\mathbf{z} - \mathbf{z}_i\|. \quad (3)$$

将上述 $n$ 个样本点的输入输出关系 $[\mathbf{z}_i, f_i]$ 代入

式(2)中,得到关于权系数向量 $\omega$ 的线性方程组

$$\begin{bmatrix} \varphi_1(r_{11}) & \varphi_2(r_{12}) & \cdots & \varphi_n(r_{1n}) \\ \varphi_1(r_{21}) & \varphi_2(r_{22}) & \cdots & \varphi_n(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(r_{n1}) & \varphi_2(r_{n2}) & \cdots & \varphi_n(r_{nn}) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}. \quad (4)$$

其中: $r_{ij}$ 为样本点 $i$ 与 $j$ 之间的欧氏距离,即 $r_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$ ;  $\varphi_j(r_{ij})$ 为基函数. 常用基函数形式为 Gauss 基函数、Multiquadric 基函数以及 Cubic 基函数,为简单起见,选取 Cubic 函数作为效能优化中 RBF 元模型的基函数. 求解式(4)方程组得到权系数向量 $\omega$ 后,将其代入式(2)即可得到预测点 $\mathbf{z}$ 的代理模型 $s(\mathbf{z})$ . 与高斯过程和支持向量回归机相比较, RBF 元模型更加简单、易实现,且计算量小,是一类较为方便且非线性泛化能力较强的元模型.

### 2.2 改进的多目标全局优化方法

体系效能多目标优化问题具有连续离散变量混合、非线性以及多峰值等特点. 传统的进化算法在处理离散变量时,多数将其视为连续变量并在其值域内进行遗传迭代,得到最优解,再将连续最优解进行离散化处理,包括两类: 1) 直接圆整法,即选取最近的离散值作为最优解,例如,文献[18]将圆整法应用于结构可置信性离散优化; 2) 在连续最优解附近按照某种方式搜索离散最优解,例如,文献[19]将连续优化问题的最优解集离散化,提出了基于就近取值策略的离散多目标优化算法. 上述两类方法均是在连续最优解附近寻找离散最优解,然而离散最优解不一定在连续最优解附近,且获取最优解效率大大降低,故应直接在连续离散混合空间进行搜索,得到最优解.

本文在传统多目标全局优化算法 NSGA-II 的基础上,引入染色体混合编码方式,对连续离散变量进行分别编码,并提出混合变量染色体的交叉、变异算子. 通过测试用例验证所提方法的性能优于传统 NSGA-II 方法,具体内容如下:

#### 1) 混合编码方式.

根据体系效能优化中多个连续、离散输入变量集 $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{u}^T)$ 可知,每个染色体包含两个基因片段. 与连续变量个数相对应,第1片段上包含 $k_1$ 个基因点位,采用浮点数的编码方式;与离散变量个数对应,第2片段上包含 $k_2$ 个基因点位,采用整数编码的方式.

#### 2) 交叉算子.

为提高遗传优化种群的多样性、避免早熟收敛,

针对混合编码染色体的离散、连续基因片段分别设计不同的交叉算子. 交叉概率  $P_c$  一般由用户指定. 浮点数基因片段采用算数交叉, 假设选择的两个父代染色体分别为  $\text{Fa}_1 = (x_1^1, x_2^1, \dots, x_i^1, \dots, x_{k_1}^1)$ 、 $\text{Fa}_2 = (x_1^2, x_2^2, \dots, x_i^2, \dots, x_{k_1}^2)$ , 经过浮点数交叉产生的子代基因片段为  $\text{Ch}_1 = (y_1^1, y_2^1, \dots, y_i^1, \dots, y_{k_1}^1)$  和  $\text{Ch}_2 = (y_1^2, y_2^2, \dots, y_i^2, \dots, y_{k_1}^2)$ , 其中第  $i$  个基因可由下式得到:

$$\begin{cases} y_i^1 = P_{a,i} x_i^1 + (1 - P_{a,i}) x_i^2, \\ y_i^2 = P_{a,i} x_i^2 + (1 - P_{a,i}) x_i^1. \end{cases} \quad (5)$$

其中:  $P_{a,i} \in [0, 1]$  为每个基因位的交叉因子, 一般在交叉操作前随机生成.

整数基因片段采用点式交叉, 给定父代染色体分别为  $\text{Fa}_1 = (u_1^1, u_2^1, \dots, u_i^1, \dots, u_{k_2}^1)$ 、 $\text{Fa}_2 = (u_1^2, u_2^2, \dots, u_i^2, \dots, u_{k_2}^2)$ , 经过点式交叉后得到的子代基因片段为

$$\begin{cases} \text{Ch}_1 = (u_1^1, u_2^1, \dots, u_i^1, u_{i+1}^2, \dots, u_{k_2}^2), \\ \text{Ch}_2 = (u_1^2, u_2^2, \dots, u_i^2, u_{i+1}^1, \dots, u_{k_2}^1). \end{cases} \quad (6)$$

其中:  $i \in [1, 2, \dots, k_2 - 1]$  代表随机产生的基因交叉点位标号.

### 3) 变异算子.

采用混合编码方式的染色体中, 浮点数和整数基因片段同样需要设计不同的交叉方式, 进而获得较高的全局优化性能, 其中染色体变异概率  $P_m$  一般由用户指定. 给定父代染色体  $\text{Fa}_1$  上的基因  $x_i^1$ , 变异后产生的子代染色体  $\text{Ch}_1$  上的基因  $y_i^1$  由下式确定:

$$y_i^1 = \begin{cases} x_i^1 + \tau(b_i - x_i^1)\varepsilon, \text{rnd}(2) = 0; \\ x_i^1 - \tau(x_i^1 - a_i)\varepsilon, \text{rnd}(2) = 1. \end{cases} \quad (7)$$

其中:  $\text{rnd}(2)$  为随机均匀地产生的正整数与 2 取模所得的结果,  $\tau \in (0, 1]$  为变异系数,  $\varepsilon$  为  $(0, 1)$  区间上的随机数,  $[a_i, b_i]$  为该基因位取值的上下界.

针对整数基因片段采用随机变异法, 变异后产生的子代个体  $\text{Ch}_1$  上的基因  $y_i^1$  为  $[a_i, b_i]$  内的随机整数, 若与父代基因相同, 则继续执行变异操作, 直至新的基因产生.

### 4) 改进 NSGA-II 的算法验证.

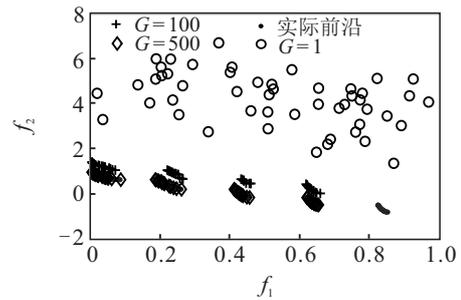
为验证所提方法的有效性, 即避免早熟收敛、提升算法多样性和优化效率, 分别选取连续离散变量混合的多目标测试函数 ZDT3 和 ZDT1 对所提算法和传统 NSGA-II 算法进行测试. ZDT3 为多峰、非连续 Pareto 前沿的函数, 用于验证所提算法具有避免早熟

收敛特性, 其函数形式为

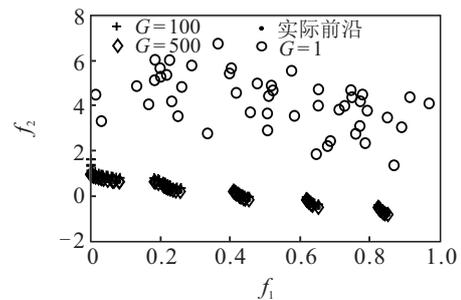
$$\begin{cases} f_1(x) = y_1, \\ f_2(x) = g(x) \left[ 1 - \sqrt{f_1(x)/g(x)} - \frac{f_1(x)}{g(x)} \right. \\ \left. \sin(10\pi x_1) \right]. \end{cases} \quad (8)$$

其中:  $g(x) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$ ;  $x = [x_1, x_2, \dots, x_7]^T \in [0, 1]^7$ ;  $x_7$  为整数型离散输入变量, 取值为 0 或 1;  $x_1 \sim x_6$  为连续型输入变量.

针对连续离散变量混合的 ZDT3 函数, 给定两种优化算法的初始种群个体数为 50, 迭代次数为 500, 固定两种算法的初始种群样本取值. 设定所提算法的交叉、变异概率为  $P_c = 0.9$  和  $P_m = 0.3$ ; NSGA-II 算法中模拟二项式交叉 (simulated binary crossover, SBX) 算子的交叉、变异分布指数分别设置为  $\eta_\mu = 20$  和  $\eta_m = 20$ . 两种算法的 Pareto 前沿随进化代数变化的散点如图 2 所示.



(a) 传统 NSGA-II 算法各代前沿



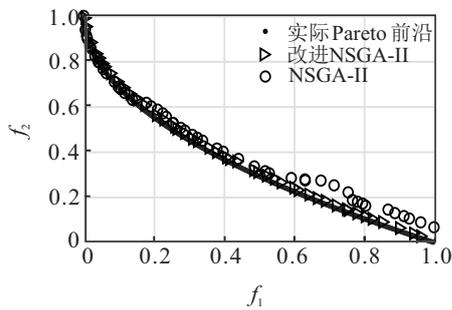
(b) 改进 NSGA-II 算法各代前沿

图 2 两种算法各代前沿对比

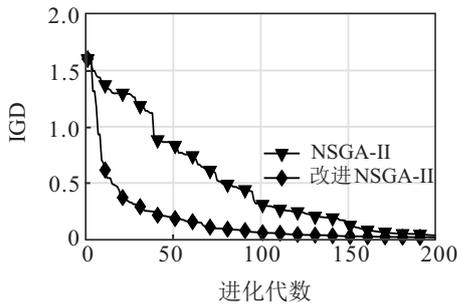
由图 2 可知, 针对连续离散混合变量的 ZDT3 函数, 传统 NSGA-II 算法得到的各代 Pareto 前沿于第 151 代提前收敛到局部, 未能覆盖所有实际前沿, 存在早熟收敛问题; 改进的 NSGA-II 算法于 151 代以前收敛到全局最优, 且各代前沿能够覆盖全局最优解, 避免了早熟收敛问题, 且较传统 NSGA-II 算法更逼近实际前沿.

本文以连续离散混合变量的多目标测试函数 ZDT1 为例, 与传统 NSGA-II 算法进行对比, 以验证改进 NSGA-II 得到 Pareto 前沿的性能, 即收敛性、分布

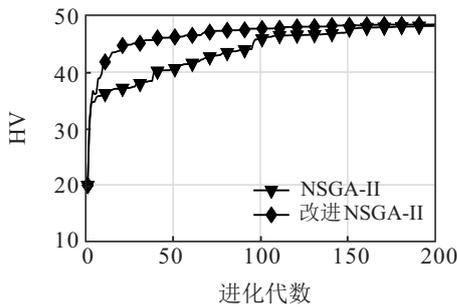
性和多样性. 给定两种优化算法的初始种群个体数为50, 迭代次数为200, 固定两种算法的初始种群样本取值, 其他参数配置与ZDT3函数相同, 得到的Pareto最优前沿如图3(a)所示. 为获取算法性能指标的量化结果, 选取较常用的性能评价指标: 反向世代距离(inverted generational distance, IGD)<sup>[12]</sup>和超体积(hypervolume, HV)<sup>[10]</sup>度量Pareto前沿的收敛性、分布性及多样性. 其中: IGD数值越小, Pareto前沿的收敛性和分布性越好; HV指标数值越大表明算法收敛性和多样性越好. 给定HV性能指标的参考点为(7, 7). 得到算法性能指标IGD和HV变化趋势如图3(b)和图3(c)所示.



(a) 两种优化算法 Pareto 前沿对比



(b) 两种优化算法 Pareto 前沿的IGD指标对比



(c) 两种优化算法 Pareto 前沿的HV指标对比

图3 两种算法Pareto前沿及性能指标对比

由图3可知,改进NSGA-II较传统的NSGA-II算法更靠近实际的Pareto前沿,且具有较好的多样性、分布性及收敛性.同时,就性能指标IGD和HV的变化趋势而言,改进的NSGA-II较传统NSGA-II算法收敛更快,有效提高了混合变量多目标优化的效率.综上所述,改进的NSGA-II算法性能优于传统NSGA-II,能够更好地适应体系效能多目标优化任务.

此外,所提算法需要由用户指定交叉概率 $P_c$ 和变异概率 $P_m$ .其中: $P_c$ 的取值范围一般为[0.7, 0.9], $P_m$ 的取值范围一般为[0.2, 0.4].为辅助用户选取 $P_c$ 、 $P_m$ 参数值,本文以函数ZDT1为例,对交叉概率分别取0.7, 0.8和0.9,变异概率分别取0.2, 0.3和0.4时的算法性能进行分析.给定算法的初始种群个体数为50,迭代次数为200,固定参数分析实验的初始样本值.选取IGD和HV指标度量算法性能,设定HV性能指标的参考点为(7, 7).为消除随机性带来的影响,对每个交叉变异概率取值下的实验重复运行30次,获取平均性能指标值,结果如表1所示.

表1 改进NSGA-II算法交叉变异参数分析实验IGD和HV指标值

		$P_c = 0.7$	$P_c = 0.8$	$P_c = 0.9$
IGD	$P_m = 0.2$	0.0659	0.0851	0.1070
	$P_m = 0.3$	0.0336	0.0392	0.0513
	$P_m = 0.4$	0.0196	0.0218	0.0299
HV	$P_m = 0.2$	47.79	47.56	47.35
	$P_m = 0.3$	48.19	48.12	48.02
	$P_m = 0.4$	48.39	48.39	48.27

根据表1得到IGD指标值的标准差为0.03, HV指标值的标准差为0.37.由此可知,选取不同的交叉和变异算子对所提算法的性能未产生较大影响.因此,本文所提的改进NSGA-II算法对正常范围内不同的交叉和变异算子取值具有较好的鲁棒性.

### 2.3 基于开发-探索的序贯抽样

CES-SEO算法的目的是基于元模型寻找体系效能指标和费用的Pareto最优解,每次算法迭代过程中,需要从现有元模型的最优解中选取昂贵仿真序贯实验点,用于下次迭代更新元模型,进而逐渐逼近实际的Pareto前沿.本文采用基于开发-探索的序贯抽样策略迭代更新元模型.探索策略方面,分别考虑决策域和目标域的空间填充性,引入基于距离的 $\phi_p$ 准则<sup>[20]</sup>和均匀性的 $CL_2$ -离差准则<sup>[21]</sup>,给出公式如下:

$$\begin{cases} z_{t_1}^{new} = \arg \min_{i=1,2,\dots,N} \left[ \sum_{j=1,2,\dots,n} d(z_i^{new}, z_j) \right]^{-p} \Bigg]^{1/p}, \\ z_{t_2}^{new} = \arg \min_{i=1,2,\dots,N} \left[ \sum_{j=1,2,\dots,n} d(\hat{F}(z_i^{new}), F(z_j)) \right]^{-p} \Bigg]^{1/p}. \end{cases} \quad (9)$$

$$\begin{cases} z_{t_3}^{new} = \arg \min_{i=1,2,\dots,N} CL_2(z_i^{new} \cup \{z_1, z_2, \dots, z_n\}), \\ z_{t_4}^{new} = \arg \min_{i=1,2,\dots,N} CL_2(\hat{F}(z_i^{new}) \cup \{F(z_1), F(z_2), \dots, F(z_n)\}). \end{cases} \quad (10)$$

其中:  $[z_j, \mathbf{F}_j(z_j)] (j = 1, 2, \dots, n)$  为当前昂贵仿真样本集,  $[z_i^{\text{new}}, \hat{\mathbf{F}}_i(z_i^{\text{new}})] (i = 1, 2, \dots, N)$  为当前元模型的最优样本集,  $p$  为正整数,  $d(z_i^{\text{new}}, z_j)$  为样本点  $z_i^{\text{new}}$  与  $z_j$  间的欧氏距离. 根据  $\phi_p$  和  $\text{CL}_2$ -离差准则筛选当前元模型最优解中满足空间填充性准则的样本点  $[z_i^{\text{new}}, \mathbf{F}(z_i^{\text{new}})]$ ,  $i = t_1, t_2, t_3$ , 用于下次迭代更新元模型.

开发策略方面, 在当前元模型最优样本集的决策和目标空间分别进行  $k$ -means 聚类, 选取距每类中心最近的样本点作为序贯采样点. 需要说明的是, 聚类前需先评估数据集最优的类别数. 本文采用 Calinski-Harabasz 准则计算类别数, 公式如下:

$$\text{VRC}_K = \frac{\text{SS}_B}{\text{SS}_W} \times \frac{(N - K)}{K - 1}. \quad (11)$$

其中:  $\text{SS}_B$  为类间差异,  $\text{SS}_W$  为类内差异, 计算公式如下:

$$\text{SS}_B = \sum_{i=1}^K l_i \|C_i - C\|^2, \quad (12)$$

$$\text{SS}_W = \sum_{i=1}^K \sum_{e \in Q_i} \|e - C_i\|^2. \quad (13)$$

其中:  $l_i$  为第  $i$  类的样本点个数;  $C_i$  为第  $i$  类的中心,  $C$  为全体  $N$  个样本点的均值;  $Q_i$  为第  $i$  类样本集;  $e$  为  $Q_i$  中的一个数据点. 由此可见, 令 Calinski-Harabasz 准则值  $\text{VRC}_K$  最大的类别数为聚类最优类别数.

## 2.4 CES-SEO 算法流程

综上所述, CES-SEO 算法流程可归纳如下.

step 1: 算法参数初始化.

$M$ : 体系仿真最大运行次数;

$N$ : 多目标进化算法种群个体数;

$G$ : 多目标进化算法进化代数;

$K_{\max}$ : 聚类算法最大类别数;

$n_{\text{int}}$ : 初始样本点个数.

step 2: 初始样本点选择. 采用 LHS 实验设计方法对输入变量进行初始设计, 获取  $n_{\text{int}}$  个初始样本点  $z_1, z_2, \dots, z_{n_{\text{int}}}$ ; 并运行体系仿真获取费用或效能指标值  $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_{n_{\text{int}}}$ , 构成初始训练样本集  $S_n$ .

step 3: 若未达到最大体系仿真次数, 则 While  $n \leq M$ .

step 3.1: 拟合/更新元模型. 利用昂贵仿真样本集  $S_n$  对每个费用或效能目标函数进行拟合, 得到元模型集  $\hat{\mathbf{F}}_n(\mathbf{z}) = [\hat{f}_{n,1}, \hat{f}_{n,2}, \dots, \hat{f}_{n,m}]$ .

step 3.2: 基于元模型的全局优化. 利用改进的 NSGA-II 算法获取元模型集  $\hat{\mathbf{F}}_n(\mathbf{z})$  的最优解集及最优 Pareto 前沿  $\hat{S}_N = [z_i^{\text{new}}, \hat{\mathbf{F}}(z_i^{\text{new}})]$ ,  $i = 1, 2, \dots, N$ .

step 3.3: 基于开发-探索的序贯采样. 利用 2.3 节提出的基于空间填充性与聚类相结合的序贯方法从元模型最优解集  $\hat{S}_N$  中抽取序贯实验点  $\mathbf{Z}^{\text{new}} = \{z_{t_1}^{\text{new}}, z_{t_2}^{\text{new}}, \dots, z_{t_g}^{\text{new}}\}$ .

step 3.4: 运行昂贵体系仿真. 在昂贵体系仿真系统上运行序贯样本点集  $\mathbf{Z}^{\text{new}}$ , 获取仿真结果并经过计算获取体系效能指标和费用值  $\mathbf{F}(z_i^{\text{new}})$ ,  $i = t_1, t_2, \dots, t_g$ ; 将得到的训练样本点  $[z_i^{\text{new}}, \mathbf{F}(z_i^{\text{new}})]$  纳入样本集  $S_n$  中,  $i = t_1, t_2, \dots, t_g$ .

End Loop

step 4: 返回 Pareto 最优解集. 由 step 3.2 得到的 Pareto 最优解集  $\hat{S}_N$  为体系效能仿真最优解, 相应的 Pareto 前沿为最佳费用和效能指标值, 至此算法结束.

## 3 性能分析与应用验证

为验证 CES-SEO 算法的有效性, 首先给出最优解的性能评价指标; 而后, 选取 4 个具有代表性的测试函数, 与 GLoSADE、GOMORS 和 ParEGO 算法的性能进行比较; 最后, 以某防空体系的效能仿真优化为例, 验证算法的可行性和有效性.

### 3.1 性能评价准则

对于基于元模型的多目标优化算法, 通常以最优前沿 PF 逼近真实 Pareto 前沿  $\text{PF}^*$  的程度和最优解分布的均匀性为准则对其性能进行评价. 选取性能评价指标 IGD 和 HV 对本文算法性能进行评价, IGD 指标计算公式如下:

$$\text{IGD}(\text{PF}^*, \text{PF}) = \frac{\sum_{\nu \in \text{PF}^*} \min(d(\nu, \text{PF}))}{|\text{PF}^*|}. \quad (14)$$

其中:  $\min(d(\nu, \text{PF}))$  为  $\nu$  和  $\text{PF}$  中样本点最小欧式距离;  $|\text{PF}^*|$  为真实 Pareto 样本点个数;  $\text{IGD} \in [0, \infty)$  用于度量最优前沿的分布性和收敛性, 越小表明算法性能越好.

此外, HV 指标通过度量 PF 的超体积达到度量 PF 收敛性和多样性的目的, 计算公式如下:

$$\text{HV}(\text{PF}) = \delta \left( \bigcup_{i=1}^{|S|} \varsigma_i \right). \quad (15)$$

其中:  $\text{HV} \in (0, \infty]$ ;  $\delta$  为 Lebesgue 测度, 用来测量体积;  $|S|$  为非支配解集的数目;  $\varsigma_i$  为参考点与第  $i$  个前沿构成的超立方体. HV 是一个有效的一元质量度量指标, 在 Pareto 支配方面是严格单调的, HV 的值越大代表算法性能越好.

### 3.2 时间复杂度分析

CES-SEO 算法复杂度由 LHS 算法复杂度、RBF 元模型拟合/更新算法复杂度、改进的 NSGA-II 算法

复杂度、序贯采样算法复杂度构成. 分析算法一次迭代过程的时间复杂度如下: LHS算法用于初始样本的一次设计, 时间复杂度为 $O(1)$ . RBF元模型方法需要计算各样本点间的距离, 并与目标个数有关, 时间复杂度为 $O(mkn^3)$ . 改进的NSGA-II算法与传统NSGA-II算法具有相同的时间复杂度 $O(mN^2)$ <sup>[21]</sup>. 序贯采样算法由基于 $\phi_p$ 与 $CL_2$ (时间复杂度分别为 $O(N)$ 、 $O(Nn)$ )相结合的探索准则与基于聚类的开发准则(时间复杂度为 $O(N)$ )构成. 综上所述, 得到CES-SEO算法的总体时间复杂度为 $O(\rho mkn^3)$ , 其中 $\rho = (M - n_{\text{int}})/(4 + 2K)$ 为CES-SEO算法的迭代次数.

相比而言, ParEGO算法主要由元模型拟合及GA算法构成. 选取Kriging元模型方法对目标模型进行近似, 算法的时间复杂度为 $O(\rho kn^3)$ , GA算法的时间复杂度为 $O(\rho N^2)$ . 因此, ParEGO算法的时间复杂度为 $O(\rho kn^3)$ , 其中 $\rho = (M - n_{\text{int}})$ . GOMORS算法同样采用了RBF元模型方法, 与本文算法不同的是, 在全局搜索的基础上引入了局部搜索策略, 时间复杂度为 $O(mN^2)$ , 故GOMORS算法的整体复杂度仍为 $O(\rho mkn^3)$ , 其中 $\rho = (M - n_{\text{int}})/4$ . GLoSADE算法的时间复杂度取决于神经网络元模型的时间复杂度 $O(kn)$ , 故算法的整体时间复杂度为 $O(\rho kn)$ ,  $\rho = (M - n_{\text{int}})/2$ . 由于GOMORS算法设计了并行优化机制, 4种算法的时间复杂度排序结果为: CES-SEO > GOMORS > ParEGO > GLoSADE.

### 3.3 实验分析

将本文算法与GLoSADE、GOMORS和ParEGO算法进行比较, 选取4个具有代表性的多目标优化函数VLMOP2、ZDT1、LZ08F1和DTLZ2进行性能测试, 并将其部分决策变量进行离散化, 用于模拟多目标效能优化问题. 给出4个算法的初始参数配置: 初始样本个数 $n_{\text{int}} = 10k$ 、 $M_{\text{VL}} = 100$ 、 $M_{\text{ZDT}} = 200$ 、 $M_{\text{LZ}} = 300$ 、 $M_{\text{DTLZ}} = 200$ 、 $N = 100$ 、 $G = 200$ 、 $K_{\text{max}} = 3$ 、 $P_c = 0.9$ 、 $P_m = 0.3$ . GOMORS算法采用传统NSGA-II算法进行优化, 其中SBX算子的交叉、变异分布指数取为 $\eta_{\mu} = 20$ 、 $\eta_m = 20$ . GLoSADE算法参数配置为: 备选个体数 $\lambda = 100$ 、差分进化交叉算子 $CR_1 = 0.4$ 、变异算子 $F_1 = 0.8$ . 由于LHS方法生成的初始样本集以及迭代优化结果存在随机性, 每种算法对每个测试函数进行30次重复实验, 选取性能指标均值和方差对4种算法性能进行比较.

VLMOP2函数具有非凸的Pareto前沿, 针对部分

决策变量进行离散化, 用以模拟体系效能优化问题, 给出函数形式如下:

$$\min \begin{cases} f_1(\mathbf{z}) = 1 - \exp\left(-\sum_{i=1}^k \left(z_i - \frac{1}{\sqrt{k}}\right)^2\right), \\ f_2(\mathbf{z}) = 1 - \exp\left(-\sum_{i=1}^k \left(z_i + \frac{1}{\sqrt{k}}\right)^2\right). \end{cases} \quad (16)$$

其中:  $\mathbf{z} = (x_1, x_2, x_3, x_4)^T \in [-2, 2]^4$ ,  $x_1$ 为整数型离散变量,  $x_2 \sim x_4$ 为连续型变量. ZDT1函数具有凸的Pareto前沿, 对其部分决策变量进行离散化, 给出函数形式如下:

$$\min \begin{cases} f_1(\mathbf{z}) = z_1, \\ f_2(\mathbf{z}) = g(\mathbf{z})[1 - \sqrt{z_1/g(\mathbf{z})}], \\ g(\mathbf{z}) = 1 + 9\left(\sum_{i=2}^k z_i\right)/(k-1). \end{cases} \quad (17)$$

其中:  $\mathbf{z} = (x_1, x_2, \dots, x_{10})^T \in [0, 1]^{10}$ , 其中 $x_1 \sim x_8$ 为连续型变量,  $x_9$ 和 $x_{10}$ 为整数型离散变量.

决策变量经离散化后的LZ08F1函数, 其Pareto前沿具有非凸、多峰、非线性特性, 给出定义如下:

$$\min \begin{cases} f_1(\mathbf{z}) = z_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[ z_j - z_1^{0.5 + \frac{3(j-2)}{2(n-2)}} \right]^2, \\ f_2(\mathbf{z}) = 1 - \sqrt{z_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[ z_j - z_1^{0.5 + \frac{3(j-2)}{2(n-2)}} \right]^2. \end{cases} \quad (18)$$

其中:  $\mathbf{z} = (x_1, x_2, \dots, x_{12})^T \in [0, 1]^{12}$ ,  $J_1 \in [2, k]$ 为奇数,  $J_2 \in [2, k]$ 为偶数. 对部分决策变量进行离散化,  $x_1 \sim x_{11}$ 为连续型变量,  $x_{12}$ 为整数型离散变量.

DTLZ2函数具有3个目标、多峰值、非凸的Pareto前沿, 对其部分决策变量进行离散化, 给出函数形式如下:

$$\min \begin{cases} f_1(\mathbf{z}) = z_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[ z_j - z_1^{0.5 + \frac{3(j-2)}{2(n-2)}} \right]^2, \\ f_2(\mathbf{z}) = 1 - \sqrt{z_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[ z_j - z_1^{0.5 + \frac{3(j-2)}{2(n-2)}} \right]^2. \end{cases} \quad (19)$$

其中:  $\mathbf{z} = (x_1, x_2, \dots, x_4)^T \in [0, 1]^2 \times [-1, 1]^2$ ,  $x_1 \sim x_3$ 为连续型变量,  $x_4$ 为整数型离散变量.

为度量4种算法得到的Pareto前沿性能, 设定VLMOP2函数的HV性能指标参考点为(1, 1)、ZDT1函数为(1.2, 1.2)、LZ08F1函数为(3, 3)、DTLZ2函数

为(7, 7, 7), 得到性能指标统计分析结果如表2和表3所示.

表2 4种算法对测试函数优化的性能指标IGD均值(方差)统计结果  $10^{-3}$

函数名称	CES-SEO	GLoSADE	GOMORS	ParEGO
VLMOP2	4.4(0.2)	5(0.4)	10(7.6)	44.5(31.9)
ZDT1	10.2(6.8)	12.4(5.1)	14.2(9.1)	39(32.3)
LZ08F1	21(4.6)	26.4(4.8)	35.7(4.1)	95.3(11.8)
DTLZ2	122.2(3.5)	125.7(3.9)	125.6(12.8)	151.6(20.1)

通过实验可知: 用4种算法对测试函数进行优化得到的Pareto前沿均能完全覆盖真实前沿. 对于VLMOP2、LZ08F1和DTLZ2函数, CES-SEO算法较GLoSADE、GOMORS和ParEGO算法更加逼近真实

Pareto前沿; 对于ZDT1函数, 4种算法均具有较优的Pareto前沿. 由30次重复实验得到的性能指标均值和方差可以看出, 对于VLMOP2函数而言, CES-SEO算法性能优于其他算法; 对于ZDT1函数, CES-SEO算法性能优于其他算法, 但鲁棒性略低于GLoSADE算法; 对于LZ08F1函数, CES-SEO算法的IGD指标优于其他算法, 但该指标鲁棒性低于GOMORS算法, 其HV指标性能优于其他算法; 对于DTLZ2函数, CES-SEO算法的IGD指标优于其他算法, HV指标劣于GLoSADE算法, 但具有较高的鲁棒性. 总体而言, CES-SEO算法的性能优于GLoSADE、GOMORS和ParEGO算法, 得到的Pareto前沿具有较好的收敛性、分布性和多样性.

表3 4种算法对测试函数优化的性能指标HV均值(方差)统计结果

函数名称	CES-SEO	GLoSADE	GOMORS	ParEGO
VLMOP2	0.2940(0.0003)	0.2905(0.0008)	0.2871(0.0100)	0.2323(0.0535)
ZDT1	1.0925(0.0138)	1.0912(0.0114)	1.0858(0.0130)	1.0494(0.0550)
LZ08F1	6.9162(0.2575)	6.7387(0.2837)	6.5578(0.3993)	5.9886(1.4466)
DTLZ2	241.5293(7.5849)	244.0680(9.3650)	238.3604(9.7653)	56.1294(3.9215)

### 3.4 应用验证

通过对CES-SEO算法的实验分析验证了本文算法的有效性. 此外, 本文以文献[6]中某防空体系效能优化为例, 验证所提算法的可行性. 红方装备体系主要由预警探测、指挥控制、火力拦截、通讯系统以及相关的维护保障系统组成. 作战想定为防御方由两套防空武器系统构成, 蓝方由空袭目标群构成, 以飞行器为主, 并以某一高度逼近目标, 如图4所示. 基本作战流程为: 蓝方对红方保卫目标展开进攻, 红方预警探测装备发现来袭目标, 将目标信息发送至指控中心, 指控中心进行威胁排序及目标分配, 并通过通信系统为各拦截装备下达拦截任务, 拦截装备对来袭目标进行拦截并将拦截结果上报至指控中心, 同时将装备系统实时状态反馈至指控中心, 若存在设备故障, 则进行设备维修与保障, 直至所有来袭目标全部销毁, 仿真结束.

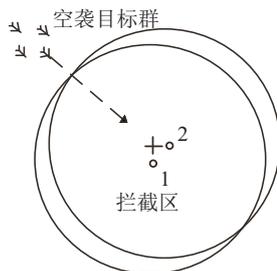


图4 防空体系作战想定

选取来袭目标拦截率  $P_s$  作为防空体系效能指标, 并与作战费用  $E$  共同构成待优化目标. 涉及的装备体系技战术性能指标包括: 最大拦截斜距  $z_1$ 、雷达发现概率  $z_2$ 、最大有效射程  $z_3$ 、单发杀伤概率  $z_4$ 、两发累积杀伤概率  $z_5$ 、火力通道数  $z_6$ 、战斗准备时间  $z_7$ 、系统反应时间  $z_8$ 、武器填装时间  $z_9$ 、携带I型武器数量  $z_{10}$ 、携带II型武器数量  $z_{11}$ 、携带III型武器数量  $z_{12}$ . 给出优化问题如下:

$$\min F(\mathbf{z}) = \min\{1 - P_s(\mathbf{z}), E(\mathbf{z})\}. \quad (20)$$

其中:  $P_s(\mathbf{z}) \in (0, 1]$ ,  $E(\mathbf{z}) \in (0, 1]$ . 作战费用  $E$  由弹药费  $E_m$ 、折损费  $E_d$ 、维护费  $E_s$  构成, 计算公式如下:

$$E(\mathbf{z}) = E_m(z_{10}, z_{11}, z_{12}) + E_d(z_4, z_5, z_{10}, z_{11}, z_{12}) + E_s(z_7, z_8, z_9). \quad (21)$$

$z_1 \sim z_5, z_7 \sim z_9$  为连续型变量,  $z_6, z_{10} \sim z_{12}$  为整数型离散变量. 给出决策变量取值区间为:  $z_1(\text{km}) \in [6, 25]$ ;  $z_2 \in [0.7, 0.9]$ ;  $z_3(\text{km}) \in [18, 20]$ ;  $z_4 \in [0.6, 0.8]$ ;  $z_5 \in [0.5, 0.7]$ ;  $z_6 \in [1, 8]$ ;  $z_7(\text{min}) \in [10, 15]$ ;  $z_8(\text{min}) \in [0.5, 1]$ ;  $z_9(\text{min}) \in [2, 4]$ ;  $z_{10} \in [3, 8]$ ;  $z_{11} \in [3, 8]$ ;  $z_{12} \in [2, 6]$ .

利用改进的NSGA-II算法对简化的数字模型迭代优化50000次, 获取体系效能优化问题的近似最优Pareto前沿, 并依此为参考; 而后采用CES-SEO算法

对体系效能和费用目标进行优化. 给出算法的参数配置为: 初始样本个数  $n_{int} = 10k$ 、 $M = 300$ 、 $N = 100$ 、 $G = 200$ 、 $K_{max} = 3$ 、 $P_c = 0.9$ 、 $P_m = 0.3$ , 得到最优Pareto前沿如图5所示.

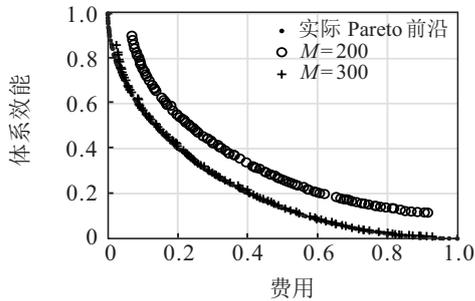


图5 防空体系效能优化结果

由图5可知: 当昂贵仿真次数  $M=200$  时, 获得的 Pareto 前沿较真实前沿存在较大差异; 继续进行序贯采样和优化直到  $M = 300$ , 获得的 Pareto 最优前沿能够较好地逼近和覆盖真实前沿, 此时用户可选取费用值在  $[0.3, 0.5]$  区间内的最优解进行使用. 由此可知, 随着昂贵仿真次数的增加, 算法得到的最优 Pareto 前沿逐渐逼近真实前沿, 表明 CES-SEO 算法是可行且有效的.

## 4 结论

本文针对复杂装备体系效能仿真优化面临的多目标多峰值、连续离散变量混合、运行开销大等问题, 提出了基于复杂昂贵仿真的体系效能多目标优化算法. 在元模型构建/更新方面, 采取基于聚类 and 空间填充准则相结合的开发-探索序贯策略, 有效地逼近了待优化目标函数; 在多目标优化方面, 引入混合编码策略和交叉变异算子, 提高了多目标全局优化性能. 通过与 GLoSADE、GOMORS 和 ParEGO 算法的实验对比, 表明 CES-SEO 算法具有较好的性能. 进一步, 以某防空体系效能仿真优化为例, 验证了算法的可行性.

CES-SEO 算法结合了序贯元模型和改进多目标优化算法, 提高了 Pareto 前沿的收敛性、分布性和多样性, 同时也增加了算法的时间复杂度. 因此, 如何提高 CES-SEO 算法的时间效率是后续需要解决的问题.

## 参考文献(References)

[1] Davendralingam N, DeLaurentis D A. A robust portfolio optimization approach to system of system architectures[J]. Systems Engineering, 2015, 18(3): 269-283.  
 [2] Gu L, Xi X, Liu K, et al. Cost effectiveness model

and optimization of weapon system based on cost as an independent variable[C]. 2016 12th World Congress on Intelligent Control and Automation. Guilin: IEEE, 2016: 2455-2459.

- [3] Liao T, Li R, You G, et al. Simulation-based ant colony optimization for complex system configuration problems[C]. Proceedings of the 9th International Conference on System of Systems Engineering. Adelaide: IEEE, 2014: 254-259.  
 [4] Romich A, Lan G H, Smith J C. A robust sensor covering and communication problem[J]. Naval Research Logistics, 2015, 62(7): 583-594.  
 [5] 程贲, 鲁延京, 周宇. 武器装备体系优化方法研究进展[J]. 系统工程与电子技术, 2012, 34(1): 85-90.  
 (Cheng B, Lu Y, Zhou Y. Research process on weapon system-of-system optimization methods[J]. System Engineering and Electronics, 2012, 34(1): 85-90.)  
 [6] 钱晓超, 董晨, 陆志洋. 基于效能评估的武器装备体系优化设计方法[J]. 系统仿真技术, 2017, 13(4): 286-291.  
 (Qian X C, Dong C, Lu Z F. Weapon system optimization design based on effectiveness evaluation[J]. System Simulation Technology, 2017, 13(4): 286-291.)  
 [7] Massaro A, Benini E. A surrogate-assisted evolutionary algorithm based on the genetic diversity objective[J]. Applied Soft Computing, 2015, 36: 87-100.  
 [8] Bartz-Beielstein T, Zaefferer M. Model-based methods for continuous and discrete global optimization[J]. Applied Soft Computing, 2017, 55: 154-167.  
 [9] Brownlee A E I, Wright J A. Constrained, mixed-integer and multi-objective optimization of building designs by NSGA-II with fitness approximation[J]. Applied Soft Computing, 2015, 33: 114-126.  
 [10] Beykal B, Boukouvala F, Floudas C A, et al. Optimal design of energy systems using constrained grey-box multi-objective optimization[J]. Computers and Chemical Engineering, 2018, 116: 488-502.  
 [11] Wang Y, Yin D Q, Yang S X, et al. Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints[J]. IEEE Transactions on Cybernetics, 2019, 49(5): 1642-1656.  
 [12] Palar P S, Tsuchiya T, Parks G T. A comparative study of local search within a surrogate-assisted multi-objective memetic algorithm framework for expensive problems[J]. Applied Soft Computing, 2016, 43: 1-19.  
 [13] Akhtar T, Shoemaker C A. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection[J]. Journal of

- Global Optimization, 2016, 64(1): 17-32.
- [14] Rosales-Perez A, Coello C A C, Gonzalez J A, et al. A hybrid surrogate-based approach for evolutionary multi-objective optimization[C]. IEEE Congress on Evolutionary Computation. Cancun: IEEE, 2013: 2548-2555.
- [15] Diaz-Manriquez A, Toscano-Pulido G, Gomez-Flores W. On the selection of surrogate models in evolutionary optimization algorithms[C]. IEEE Congress of Evolutionary Computation. Piscataway: IEEE, 2011: 2155-2162.
- [16] 杨宁, 霍炬, 杨明. 基于多层次信息交互的多目标粒子群优化算法[J]. 控制与决策, 2016, 31(5): 907-912.  
(Yang N, Huo J, Yang M. Multi-objective particle swarm optimization algorithm based on the interaction of multi-level information[J]. Control and Decision, 2016, 31(5): 907-912.)
- [17] Li W, Lu L, Xie X, et al. A novel extension algorithm for optimized latin hypercube sampling[J]. Journal of Statistical Computation and Simulation, 2017, 87(13): 2549-2559.
- [18] 杜剑明. 结构可置信性鲁棒优化若干算法研究及其拓展[D]. 大连: 大连理工大学运载工程与力学学部, 2011.  
(Du J M. A study on some algorithms for confidence structural robust optimization and their extensions[D]. DaLian: Faculty of Vehicle Engineering and Mechanics, Dalian University of Technology, 2011.)
- [19] 李二超, 马玉泉. 基于就近取值策略的离散多目标优化[J]. 南京大学学报: 自然科学, 2018, 54(6): 1216-1224.  
(Li E C, Ma Y Q. Discrete multi-objective optimization based on the nearest value strategy[J]. Journal of Nanjing University: Natural Science, 2018, 54(6): 1216-1224.)
- [20] Morris M D, Mitchell T J. Exploratory designs for computational experiments[J]. Journal of Statistical Planning and Inference, 1995, 43(3): 381-402.
- [21] Hickernell F J. A generalized discrepancy and quadrature error bound[J]. Mathematics of Computation of the American Mathematical Society, 1998, 67(221): 299-322.

### 作者简介

林圣琳(1989—), 男, 博士生, 从事复杂仿真评估与VV&A的研究, E-mail: lin\_44627079@yeah.net;

李伟(1980—), 男, 副教授, 博士生导师, 从事仿真分析与评估、分布式仿真、仿真实验设计等研究, E-mail: frank@hit.edu.cn;

钱晓超(1986—), 男, 工程师, 博士, 从事武器装备体系作战效能评估等研究, E-mail: everqxc@hotmail.com;

马萍(1970—), 女, 教授, 博士生导师, 从事复杂系统建模与仿真、分布式仿真与VV&A等研究, E-mail: pingma@hit.edu.cn;

杨明(1963—), 男, 教授, 博士生导师, 从事飞行器制导与控制、复杂系统仿真实理论与方法等研究, E-mail: myang@hit.edu.cn.

(责任编辑: 闫妍)