

基于策略自适应的多目标差分进化算法及其应用 *

毕超超¹, 范勤勤^{1,2†}, 王维莉¹

(1. 上海海事大学 物流研究中心, 上海 201306; 2. 上海交通大学 电子信息与电气工程学院, 上海 200240)

摘要: 变异和交叉策略对多目标差分进化算法(differential evolution, DE)的性能有着显著影响。为提高多目标差分进化算法求解多目标优化问题的能力, 提出一种基于策略自适应的多目标差分进化算法(multi-objective differential evolution algorithm based on self-adaptive strategy, MODE-SS)。该算法采用超体积(hyper-volume, HV)对变异策略进行性能评价, 并实现变异策略的自动选择; 使用动态调整的二项式交叉策略和模拟二进制交叉(simulated binary crossover, SBX)策略实现全局搜索与局部搜索的平衡。通过与其他 6 种多目标进化算法在 10 个测试函数上的性能比较, 结果表明 MODE-SS 的整体性能要好于其他所比较算法。最后, 将 MODE-SS 用于求解海铁联运能耗优化问题, 所得结果能够为决策者提供多种可行方案。

关键词: 差分进化; 多目标优化; 自适应; 海铁联运; 能耗优化

中图分类号: TP301.6 **doi:** 10.19734/j.issn.1001-3695.2018.12.0931

Multi-objective differential evolution algorithm based on self-adaptive strategy and ITS application

Bi Chaochao¹, Fan Qinquin^{1,2†}, Wang Weili¹

(1. Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China; 2. School of Electronic Information & Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: The selection of mutation and crossover strategies significantly impact the performance of multi-objective differential evolution (DE) algorithm. To improve the capability of multi-objective differential algorithm, this paper proposes a multi-objective differential evolution algorithm based on self-adaptive strategy (MODE-SS). This algorithm not only used the hyper-volume to evaluate the performance of mutation strategies and selected a suitable mutation strategy automatically, but also employed binomial and simulated binary crossover strategies to balance global and local search capabilities. The results on 10 test functions show that MODE-SS outperforms than the other six state-of-the-art multi-objective optimization algorithms. Finally, MODE-SS was used to solve the problem of energy consumption optimization of sea-rail intermodal transportation; the results can provide a set of available solutions to decision-makers.

Key words: differential evolution; multi-objective optimization; self-adaptation; sea-rail intermodal transportation; energy optimization

0 引言

在现实世界中,大多数优化问题是属于多目标优化问题(Multi-objective optimization problems, MOP),即各个目标之间是彼此相互冲突的。相比于传统求解方法,多目标进化算法(multi-objective evolution algorithms, MOEs)具有诸多优点,如一次运行能得到多个解、鲁棒性好等。其中,多目标差分进化算法(multi-objective differential evolution, MODE)因为其优化能力强、鲁棒性好等优点成为主流多目标进化算法之一,如经典的多目标差分进化算法 GDE3^[1];另外,将差分进化算法与其他进化算法进行混合求解多目标优化问题也取得了较好结果,如 NSGAI-DE^[2]。文献[3]将差分变异算子融入到多目标免疫算法中,改善了多目标免疫算法易陷局部最优问题。文献[4]通过对差分进化算法的参数进行自适应调整来提高算法求解多目标优化问题的能力。虽然以上研究都大大提升了多目标差分进化算法的性能,但是变异策略和交叉策略对差分进化算法的性能有着显著影响。因此,为进一步提高多目标差分进化算法的性能,本文提出一种基于策略自适应的多目标差分进化算法(multi-objective differential

evolution algorithm based on self-adaptive strategy, MODE-SS)。在该算法中,采用超体积(hypervolume, HV)性能指标来对各个变异策略的性能进行评价,并实现变异策略的自动选择。另外,使用二项式交叉策略和模拟二进制交叉策略的动态调整来实现多目标差分进化算法局部搜索与全局搜索的平衡。

1 相关工作

为提升多目标差分进化算法的性能,文献[5]采用多变异策略方法来实现算法在进化过程中解集收敛性与多样性的平衡。文献[6]为防止算法陷入局部最优,同样使用多个差分变异算子来保持种群的多样性。文献[7]提出一种新的混合交叉算子,该多目标进化算法通过差分变异策略 DE/rand/1 和 simulated binary crossover (SBX) 构造出一种新的自适应混合交叉算子,目的是均衡多目标进化算法的全局与局部搜索能力。文献[8]提出一种基于性能指标的多目标差分进化算法(multi-objective differential evolution with performance-metric-based self-adaptive mutation operator, MODE-PMSMO),该算法通过一种多目标性能评价指标来指导变异策略的动态调整。文献[9]提出一种多种群合作的多目标差分进化算法

收稿日期: 2018-12-19; 修回日期: 2019-02-26 基金项目: 国家重点研发计划资助项目(2016YFC0800200); 国家自然科学基金资助项目(61603244); 中国博士后科学基金资助项目(2018M642017)

作者简介: 毕超超(1992-), 男, 江苏宿迁人, 硕士研究生, 主要研究方向为多目标优化算法; 范勤勤(1986-), 男(通信作者), 浙江宁波人, 副教授, 博士(后), 主要研究方向为多目标优化、机器学习、进化计算(forever123fan@163.com); 王维莉(1986-), 女, 湖北武汉人, 讲师, 硕士, 主要研究方向为人群疏散、智能交通。

(cooperative differential evolution with multiple populations for multi-objective optimization, CMODE), 该算法采用不同的差分变异算子和交叉策略对单目标种群以及精英种群进行迭代, 以此来加快算法的收敛速度。文献[10]在动态种群进化模型中加入多个差分变异算子来提高算法所得 Pareto 解集的质量。由以上研究可知, 在多目标差分进化算法中使用多种操作算子有利于提高 Pareto 解集的质量。因此, 本文提出一种基于策略自适应的多目标差分进化算法 (multi-objective differential evolution algorithm based on self-adaptive strategy, MODE-SS)。在 MODE-SS 中, 采用超体积性能评价指标来指导变异策略地动态调整; 同时, 采用二进制交叉策略和模拟二进制交叉策略实现算法的均衡搜索。

2 预备知识

2.1 差分进化算法

差分进化算法是由 Storn 等人^[11]于 1995 年提出的一种高效启发式并行搜索算法, 具有构造简单、收敛速度快、设置参数少等优点, 引起了诸多国内外研究者的关注与重视。差分进化算法的具体操作如下。

2.1.1 初始化操作

在本操作中, 设置算法参数例如缩放因子 F , 交叉概率 CR , 种群规模 PS 和最大迭代次数 G_{max} 等; 在定义域 (Ω) 内, 生成初始化种群 $x_i^0 (i = 1, 2, \dots, PS)$ 。

2.1.2 变异操作

初始化操作之后, 算法进入变异操作阶段。这一步的作用是利用种群中的个体通过变异方程生成变异个体 M_i^G 。差分进化算法常见的变异方程如式 (1) ~ (5) 所示。

“DE/rand/1”:

$$M_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) \quad (1)$$

“DE/rand/2”:

$$M_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) + F \cdot (x_{r_4}^G - x_{r_5}^G) \quad (2)$$

“DE/current-to-best/1”:

$$M_i^G = x_i^G + F \cdot (x_{best}^G - x_i^G) + F \cdot (x_{r_1}^G - x_{r_2}^G) \quad (3)$$

“DE/best/1”:

$$M_i^G = x_{best}^G + F \cdot (x_{r_1}^G - x_{r_2}^G), \quad (4)$$

“DE/rand-to-best/1”:

$$M_i^G = x_{r_1}^G + F \cdot (x_{best}^G - x_i^G) + F \cdot (x_{r_2}^G - x_{r_3}^G) \quad (5)$$

其中: $x_{r_1}^G, x_{r_2}^G, x_{r_3}^G, x_{r_4}^G, x_{r_5}^G$ 是第 G 代种群中随机选取的互不相同的个体; x_i^G 表示第 G 代种群的第 i 个个体; r_1, r_2, r_3, r_4, r_5 是 $[1, PS]$ 范围内随机选取的互不相同的整数且都与 i 不同; x_{best}^G 表示第 G 代种群中最好的个体。

2.1.3 交叉操作

对于每个个体 x_i^G , 通过交叉策略生成实验个体 u_i^G 。生成方式如下:

$$u_{ij}^G = \begin{cases} M_{ij}^G, & R_j \leq CR \text{ or } j = j_{rand} \\ x_{ij}^G, & \text{否则} \end{cases}, j = 1, 2, \dots, D \quad (6)$$

其中: D 表示优化问题的维度; R_j 表示 $[0, 1]$ 间均匀分布的随机数; j_{rand} 表示 $[1, D]$ 范围中的一个随机整数。

2.1.4 选择操作

在选择操作中, 实验个体 u_i^G 与种群个体 x_i^G 中较优秀的个体将会进入下一代, 选择方式如下:

$$x_i^{G+1} = \begin{cases} u_i^G, & f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{否则} \end{cases} \quad (7)$$

2.2 多目标优化问题

通常情况下, 多目标优化中的各个子目标之间是相互冲突的。一个子目标的改善可能会导致其他一个或多个子目标性能的降低, 即多个子目标不可能同时达到最优而只能得到到它们的折中解。多目标优化的目的是尽可能地使各个子目

标达到最优解, 多目标的解是一组解而非单个解。不失一般性, 多目标优化问题的数学形式表示如下:

$$\min_{x \in \Omega} F(x) = \{f_1(x), f_2(x), \dots, f_m(x)\}^T, \quad (8)$$

其中: $\Omega \in R^D$ 为 D 维决策空间; $F(x): \Omega \rightarrow R^m$ 包含了 m 个待优化的目标函数; R^m 为目标空间。同时, 多目标一些基本概念如下:

定理 1 支配关系。决策向量 x_1 支配另一个决策向量 x_2 (记做 $x_1 \succ x_2$) 的条件是: 如果对 $\forall n \in \{1, 2, \dots, m\}$, $f_n(x_1) \leq f_n(x_2)$, 且 $\exists k \in \{1, 2, \dots, m\}$, $f_k(x_1) < f_k(x_2)$ 。

定理 2 Pareto 最优解集。一个向量 x^* , 如果不存在其他向量 x , 使得 $x \succ x^*$, 那么就称 x^* 为 Pareto 解, x^* 的集合 (记做 X^*) 被称为 Pareto 最优解集。

定理 3 Pareto 前沿。多目标优化问题中, Pareto 最优解集中的解对应的目标向量被称为 Pareto 前沿 (pareto front, PF), 表示为 $PF = \{F(x^*) | x^* \in X^*\}$ 。

2.3 多目标性能评价指标

多目标性能评价通常采用世代距离 (generational distance, GD) 和反转世代距离 (inverted generational distance, IGD)^[12-14]。GD 表示得到的 PF 与真实 PF 的逼近程度, 而 IGD 则是对得到 PF 的收敛性和分布性进行综合评价。超体积 (hyper-volume, HV) 是一种综合性的多目标性能评价指标, 也被称为 S 测度或 Lebesgue 测度, 能够同时评价所得 PF 的收敛性与多样性, HV 的结果值越大, 表明所得 PF 的收敛性和分布性越好^[13]。各个评价方式具体的数学表达如下所示:

a) 世代距离 (GD) 计算方式。

$$GD(PF, PF^*) = \frac{\sqrt{\sum_{q \in PF} d(q, PF^*)^2}}{|PF|}, \quad (9)$$

其中: PF 表示进化计算得到的 Pareto 前沿, PF^* 表示真实的 Pareto 前沿 (true pareto front); $d(q, PF^*)$ 表示进化计算得到的向量 q 与 PF^* 中的点最小欧氏距离; $|PF|$ 表示 PF 中点的数量。

b) 反转世代距离 (IGD) 计算方式。

$$IGD(PF, PF^*) = \frac{\sqrt{\sum_{q \in PF^*} d(q, PF)^2}}{|PF^*|}, \quad (10)$$

其中: PF 表示进化计算得到的 Pareto 前沿, PF^* 表示真实的 Pareto 前沿 (true pareto front); $d(q, PF)$ 表示真实 Pareto 中的向量 q 与 PF 的最小欧氏距离; $|PF^*|$ 表示 PF^* 中点的数量。

c) 超体积 (HV) 的计算方式^[7] 如式 (11) 所示。

$$HV(PF) = Vol(\cup_{x \in PF} [f_1(x), z_1^r] \times \dots \times [f_m(x), z_m^r]) \quad (11)$$

其中: PF 表示得到的解集; $Vol(\cdot)$ 代表 Lebesgue 测度; $z^r = (z_1^r, \dots, z_m^r)^T$ 表示参考点; $f_1(x), \dots, f_m(x)$ 表示目标空间中目标向量不同的优化目标的函数值。需要注意的是, 如果目标函数值比参考点差, 那这个目标函数将不被计入到 HV 值中。

3 基于策略自适应的多目标差分进化算法

为提高差分算法求解多目标问题的能力, 提出了基于策略自适应的多目标差分进化 (MODE-SS) 算法。在该算法中采用 HV 对变异策略进行评价, 实现变异策略的自适应选择, 选择 HV 作为评价指标的主要原因是其可评价解集的收敛性和分布性。本文中的变异策略分别选择 DE/rand/1 和 DE/current-to-best/1, 其主要原因是前者有很好的全局搜索能力, 后者有很好的局部开发能力; 而且为保证算法前期全局搜索与后期局部搜索的平衡, 本文采取二项式交叉和模拟二进制交叉的混合交叉策略, 具体操作步骤见 3.1 与 3.2 节。

3.1 自适应变异策略

a)利用 HV 对变异策略进行评价, 计算每个变异策略的数量。

$$N_{str_name}^G = round(PS \times \frac{HV_{str_name}^G}{S^G}), \quad (12)$$

其中: str_name 是代表两个变异策略名即为 $rand$ 或者 $current\text{-}to\text{-}best$; $N_{str_name}^G$ 表示在下一代选择名为 str_name 的变异策略的数量; $round(\cdot)$ 表示四舍五入函数;

$S^G = HV_{rand/1}^G + HV_{current\text{-}to\text{-}best}^G$ 表示两个变异策略 HV 值之和。

b)每一个变异策略数量^[8]更新方式如式 (13) 所示。

$$N_{str_name}^G = \begin{cases} N_{str_name}^{G-1} + 1, & \text{if } N_{str_name}^G > N_{str_name}^{G-1} \\ N_{str_name}^{G-1} - 1, & \text{if } N_{str_name}^G < N_{str_name}^{G-1} \\ N_{str_name}^{G-1}, & \text{否则} \end{cases} \quad (13)$$

由式 (13) 可以看出, 变异策略的数量跟其性能成正比。

3.2 混和交叉策略

为保证算法全局搜索与局部搜索的平衡, 本文将多项式交叉与模拟二进制交叉进行混合。SBX 是以父代作为搜索中心的一种搜索方法, 具有很强的局部搜索能力^[7]; 而多项式交叉策略则有较好的全局搜索能力, 因此本文将设置一个随迭代次数不断变小的值 μ 来平衡算法全局搜索能力与局部搜索能力。在算法前期, 因为 μ 值较大, 所以大概率选择式 (6) 的交叉策略, 算法具有较强的全局搜索能力; 算法后期, 因为 μ 值减小, 所以算法会大概率选择 SBX 的交叉策略, 算法具有较强的局部搜索能力。

3.3 算法的基本步骤

a)初始化种群数量 PS , 缩放因子 $F=0.5$, 交叉概率 $CR=N(0.25, 0.02)$ 是正态分布随机数, 每一个变异策略的数量初始化为 $PS/2$ 。

b)对种群进行变异操作。当 $G < G_s$ ($G_s = 0.2 * G_{max}$) 时^[8], 只采用 $DE/rand/1$ 变异策略; 当 $G \geq G_s$ 时, 同时采用 $DE/rand/1$ 和 $DE/current\text{-}to\text{-}best/1$ 两种变异策略。其中: G_{max} 、 G 分别表示最大迭代次数与当前代数; x_{best}^G 选取上一代中非支配排序的第一层所有个体。变异策略自适应方式按照 3.1 节进行。

c)对种群中的每一个个体进行交叉操作, 交叉操作见 3.2 节, 其中 $\mu = 1 - 0.7 * (G / G_{max})$ 。

d)对实验个体进行边界值^[8]判定。如果一个随机数 $rand > Bset$, 那么 $u_{i,j}^G = \begin{cases} x_{i,j}^G & u_{i,j}^G < L_j \\ x_{i,j}^G & u_{i,j}^G > U_j \end{cases}$ (L_j 和 U_j 是个体 x_i 的第 j 维的下界与上界), 否则在定义域 (Ω) 范围内重新生成。

e)依据支配关系进行选择操作, 选择操作采用文献^[8]的方法, 即:

(a)如果 $x_i^G > u_i^G$, x_i^G 依然保留在种群 x^G 中。

(b)如果 $u_i^G > x_i^G$, u_i^G 替换 x_i^G 并被保留在种群 x^G 中。

(c)如果 u_i^G 与 x_i^G 互相不支配, u_i^G 将被加到种群 x^G 中。

f)利用非支配排序和拥挤距离选择 PS 个个体进入下一代;

g) $G = G + 1$;

h)循环步骤 b)到 g), 直到满足最大迭代次数。

4 实验结果与分析

为验证 MODE-SS 算法性能, 将 MODE-SS 与其他 6 种多目标算法在 5 个双目标(ZDT1~ZDT4, ZDT6)和 5 个三目标(DTLZ1~DTLZ5)上进行性能对比。其他 6 种多目标算法分别为 GDE3^[1]、NSGAII-DE^[2]、MODE-PMSMO^[8]、MODE-RMO^[15]、SA-MTLBO^[16] 和 MOMDE-AM^[17]。为保证实验结果分析的可靠性, 采用 Wilcoxon 方法^[18]和 Friedman 非参数检验方法^[19]对实验结果进行统计分析, 显著水平设定为 5%。其中, “+”、“-”表示所提算法优于、劣于其他算法; “≈”表示所提算法与其他算法性能相似。

4.1 实验参数设置

本实验所有实验参数均与文献^[8]相同, 具体为所有测试函数种群数量均设置为 100, 双目标的最大迭代次数均设置为 250, DTLZ1~DTLZ2 的最大迭代次数设置为 300, DTLZ3 的最大迭代次数设置为 500, DTLZ4~DTLZ5 的最大迭代次数设置为 200。每一个算法求解每一个测试函数, 均独立运行 20 次。比较算法的参数设定均取自原始值。

4.2 实验比较

MODE-SS 与其他 6 种算法所得 GD 平均值与方差、Wilcoxon 方法统计结果见表 1。

表 1 不同多目标优化算法得到的 GD 平均值与方差

Table 1 Mean and standard deviation values of GD achieved by different moeas.

	GDE3	NSGAII-DE	MODE-RMO	SA-MTLBO	MOMDE-AM	MODE-PMSMO	MODE-SS
ZDT1	2.40E-03+ (5.62E-04)	5.83E-03+ (1.22E-03)	3.85E-03+ (3.22E-04)	2.32E-04- (4.65E-05)	1.06E-03+ (3.89E-04)	7.83E-04+ (4.62E-05)	3.99E-04 (2.48E-05)
ZDT2	8.20E-03+ (2.33E-03)	7.75E-03+ (5.99E-04)	6.97E-03+ (7.25E-04)	1.01E-04- (1.03E-05)	1.45E-03+ (1.36E-04)	1.10E-03+ (9.88E-05)	4.76E-04 (3.23E-05)
ZDT3	2.76E-03+ (9.22E-04)	5.31E-03+ (9.23E-04)	4.76E-03+ (5.45E-04)	1.68E-04- (1.43E-05)	8.37E-04+ (7.32E-05)	6.87E-04+ (4.55E-05)	3.20E-04 (2.52E-05)
ZDT4	3.87E-02+ (3.77E-02)	1.16E-02+ (2.15E-02)	7.29E-02+ (1.63E-01)	3.44E-03+ (6.60E-03)	8.21E-04≈ (2.81E-03)	8.01E-04≈ (2.69E-03)	2.41E-04 (4.66E-05)
ZDT6	1.71E-01+ (5.23E-02)	5.90E-02+ (4.59E-03)	2.58E-02+ (1.12E-02)	6.98E-05- (4.94E-06)	1.29E-02+ (9.61E-04)	8.47E-03+ (4.85E-04)	6.69E-03 (5.22E-04)
DTLZ1	7.08E-02+ (7.21E-02)	4.80E-03+ (1.71E-02)	2.58E-04≈ (8.75E-06)	4.18E-04+ (1.24E-04)	2.58E-04≈ (8.22E-06)	2.57E-04≈ (7.14E-06)	2.58E-04 (8.72E-06)
DTLZ2	7.25E-04≈ (2.69E-05)	1.96E-03+ (3.02E-04)	7.28E-04≈ (1.79E-05)	2.32E-03+ (2.29E-03)	7.32E-04≈ (2.27E-05)	7.30E-04≈ (2.05E-05)	7.25E-04 (2.25E-05)
DTLZ3	3.27E+00+ (4.87E+00)	1.82E-01+ (4.29E-01)	7.88E-03+ (2.32E-02)	1.51E+00+ (3.49E+00)	1.56E-02≈ (3.64E-02)	2.06E-02≈ (5.21E-02)	1.06E-02 (3.05E-02)
DTLZ4	7.12E-04 (2.35E-05)	1.46E-03+ (1.94E-04)	7.26E-04≈ (2.12E-05)	1.13E-03+ (1.12E-04)	7.29E-04≈ (2.27E-05)	7.22E-04≈ (2.27E-05)	7.21E-04 (2.39E-05)
DTLZ5	1.09E-05≈ (5.04E-06)	1.87E-04+ (5.02E-05)	9.12E-06≈ (4.41E-07)	1.26E-04+ (6.60E-05)	9.26E-06≈ (4.72E-07)	8.93E-06≈ (4.41E-07)	9.09E-06 (4.75E-07)
+	7	10	6	6	4	4	
-	0	0	0	4	0	0	
≈	3	0	4	0	6	6	

从表 1 可以看出, MODE-SS 分别优于 GDE3、MODE-RMO、SA-MTLBO、MOMDE-AM、MODE-PMSMO 算法 7、6、6、4、4 个测试函数, 其中 MODE-SS 在全部 10 个测试函数上优于 NSGAI-DE; 而 GDE3、NSGAI-DE、MODE-RMO、MOMDE-AM、MODE-PMSMO 在任何测试函数上都不能优于本文所提算法, 但 SA-MTLBO 在 4 个测试函数上优于本文所提算法; 同时, 7 种算法所得 GD 的 Friedman 统计分析结果见表 2, 实验结果显示 MODE-SS 排名第一, 说明本文所提算法得到 PF 的收敛性是所有算法中最好的, 其主要原因是 MODE-SS 使用自适应变异策略和混合交叉策略, 实现了算法搜索能力的自动匹配。

另外, MODE-SS 与其他 6 种算法所得 IGD 的平均值与方差、Wilcoxon 方法统计结果见表 3。从表 3 可以看出, MODE-SS 分别优于 GDE3、NSGAI-DE、MODE-RMO、SA-MTLBO、MOMDE-AM、MODE-PMSMO 算法 7、6、5、3、5、4 个测试函数; 而 GDE3、NSGAI-DE、MOMDE-AM 不能在任何一个测试函数上优于本文所提算法; MODE-RMO、SA-MTLBO、MODE-PMSMO 分别在 1、3、1

个测试函数上优于 MODE-SS; 7 种算法 IGD 的 Friedman 方法统计分析结果见表 4。结果显示 MODE-SS 算法排名第四, 优于算法 GDE3、NSGAI-DE、MODE-RMO, 但劣于算法 MODE-PMSMO、MOMDE-AM、SA-MTLBO, 其主要原因是在算法中加入了 SBX 增强了种群的收敛性, 导致种群分布性较差。

表 2 不同多目标优化算法得到的 GD 平均值

Friedman 测试结果排序

Table 2 Ranking of GD obtained by Friedman's test on experimental results with different moeas

Algorithm	排序
NSGAI-DE	6.2
GDE3	5.2
MODE-RMO	4.25
MOMDE-AM	3.85
SA-MTLBO	3.8
MODE-PMSMO	2.7
MODE-SS	2

表 3 不同多目标优化算法的 IGD 平均值与方差

Table 3 Mean and standard deviation values of IGD achieved by different moeas.

	GDE3	NSGAI-DE	MODE-RMO	SA-MTLBO	MOMDE-AM	MODE-PMSMO	MODE-SS
ZDT1	2.22E-03+ (6.61E-04)	2.29E-03+ (1.16E-04)	1.62E-03+ (1.11E-04)	2.69E-04+ (1.15E-05)	4.82E-04+ (2.75E-05)	4.19E-04+ (1.90E-05)	2.87E-04 (1.15E-05)
ZDT2	3.87E-03+ (1.12E-03)	3.46E-03+ (2.80E-04)	2.95E-03+ (2.78E-04)	2.71E-04- (2.32E-05)	7.01E-04+ (5.44E-05)	5.61E-04+ (4.56E-05)	3.32E-04 (1.62E-05)
ZDT3	1.74E-03+ (5.27E-04)	3.03E-03+ (2.68E-04)	2.70E-03+ (1.65E-04)	3.09E-04- (1.76E-05)	6.48E-04+ (3.07E-05)	5.34E-04+ (3.31E-05)	3.19E-04 (9.84E-06)
ZDT4	1.54E-02+ (1.42E-02)	2.14E-03+ (2.35E-03)	1.93E-03+ (3.47E-03)	3.32E-03+ (6.01E-03)	5.17E-04+ (1.20E-03)	5.20E-04≈ (1.20E-03)	2.51E-04 (1.02E-05)
ZDT6	8.74E-02+ (2.48E-02)	2.68E-02+ (1.40E-03)	9.52E-03+ (7.25E-04)	2.45E-04- (1.78E-05)	5.71E-03+ (4.38E-04)	3.76E-03+ (2.30E-04)	3.01E-03 (2.49E-04)
DTLZ1	8.06E-03+ (7.74E-03)	4.14E-04≈ (2.13E-05)	4.00E-04≈ (4.24E-04)	4.14E-04≈ (2.06E-05)	3.98E-04≈ (1.48E-05)	4.05E-04≈ (1.63E-05)	4.06E-04 (1.73E-05)
DTLZ2	1.06E-03≈ (4.79E-05)	1.06E-03≈ (5.78E-05)	1.01E-03- (3.25E-05)	1.03E-03≈ (3.22E-05)	1.01E-03≈ (3.27E-05)	1.01E-03≈ (4.52E-05)	1.05E-03 (7.09E-05)
DTLZ3	1.59E-01+ (1.05E-01)	5.93E-03+ (6.95E-03)	1.78E-03≈ (3.04E-03)	9.28E-03+ (1.13E-02)	2.98E-03≈ (4.81E-03)	3.69E-03≈ (7.04E-03)	2.35E-03 (4.03E-03)
DTLZ4	1.07E-03≈ (4.04E-05)	1.04E-03≈ (3.95E-05)	1.03E-03≈ (4.09E-05)	1.02E-03≈ (2.50E-05)	1.02E-03≈ (2.94E-05)	1.02E-03- (3.48E-05)	1.05E-03 (3.81E-05)
DTLZ5	1.02E-04≈ (6.46E-06)	9.73E-05≈ (4.38E-06)	9.72E-05≈ (7.58E-06)	9.65E-05≈ (5.76E-05)	9.91E-05≈ (6.89E-06)	9.88E-05≈ (5.91E-06)	9.98E-05 (6.19E-06)
+	7	6	5	3	5	4	
-	0	0	1	3	0	1	
≈	3	4	4	4	5	5	

表 4 不同多目标优化算法得到的 IGD 平均值

Friedman 测试结果排序

Table 4 Ranking of IGD obtained by Friedman's test on experimental results with different moeas

Algorithm	Ranking
GDE3	6.65
NSGAI-DE	5.6
MODE-RMO	3.6
MODE-SS	3.2
MOMDE-AM	3.1
MODE-PMSMO	3
SA-MTLBO	2.85

由以上分析可见, MODE-SS 在 10 个多目标测试函数上的整体表现优于其他 6 种多目标优化算法。交叉策略中引入 SBX 增强了算法的收敛性; 对变异策略的自适应选择, 提高了

算法求解复杂多目标优化问题的能力。

4.3 参数分析

在本部分, 对混合交叉策略中的参数 μ 进行分析, μ 变化范围较大有利于局部搜索, μ 变化范围较小有利于全局搜索。因此, μ 的选择会对算法整体性能产生影响。利用 4.2 节的 10 个测试函数对不同 μ 值进行测试, 使用 Friedman 方法对所得 GD 与 IGD 进行性能排序。图 1 表示由 Friedman 方法得到的不同 μ 值下 GD 与 IGD 性能排序平均值。由图 1 可知, 当 $\mu = 1 - 0.7 * G / G_{max}$ 时, 算法整体性能最好, 因此在所提算法中, $\mu = 1 - 0.7 * G / G_{max}$ 被使用。

5 在海铁联运能耗优化问题中的应用

经济的快速发展离不开全球之间的贸易联系, 船舶因为其运量大、运输距离长的特点在国际贸易运输中扮演着主要角色, 但船舶运输会消耗大量燃油和排放大量有毒有害气体,

如硫化物、碳化物 (SO_x, CO_x) 等。为保护环境, 国际海事组织 (The International Maritime Organization, IMO) 在港口岸附近设立了硫排放控制区 (Sulphur Emissions Control Areas, SECA), 船舶在硫排放控制区需要使用价格更加高昂的低硫燃料代替价格低廉的高硫燃料, 因此运输成本进一步增加。经研究发现^[20,21], 船舶速度在成本控制与污染物排放方面有着重要影响, 合理的船舶行驶速度可以有效控制成本与污染物排放。但国际贸易往来除了控制成本、污染物排放之外还需要控制时间; 航速不仅影响成本, 而且对运输时间也有重要影响。铁路作为陆地主要运输方式之一, 常被用来与船舶运输相结合来达到控制成本与运输时间的目的^[22]。

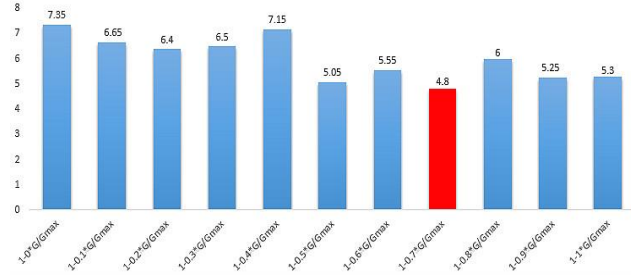


图 1 不同参数 μ 的性能排序平均值

Fig. 1 Performance average ranking of different μ

5.1 海铁联运速度与能耗关系

根据文献^[22~24], 船舶航速与能耗关系表示如下:

$$E_{ship} = \frac{0.0043 \times v_{ship}^{3.358} \times s_{ship} \times 1000}{24 \times v_{ship}} \quad (14)$$

其中: E_{ship} (kg) 表示船舶在航程 s_{ship} (knts) 上以速度 v_{ship} (knot/hour) 行驶时消耗燃油的质量。

根据文献^[22~24], 铁路运输能耗与速度关系表示如下:

$$E_{train} = \frac{\rho \times P_{train} \times s_{train}}{1000 \times v_{train}} \quad (15)$$

$$\rho = 0.004 \times v_{train}^2 - 0.8245 \times v_{train} + 271.4$$

$$P_{train} = -0.004285 \times v_{train}^3 + 0.917 \times v_{train}^2 - 35.78 \times v_{train} + 817.1$$

其中: E_{train} (kg) 表示火车在路程 s_{train} (km) 以速度 v_{train} (km/hour) 行驶时消耗燃油的质量; ρ 表示能耗率 ($g / (kW \cdot hour)$); P_{train} 表示功率 (Kw)。

5.2 海铁联运模拟运输路线

本文中海铁联运模拟运输路线分为两条铁路运输 (Rail-1, Rail-2) 和一条海洋运输, 海洋运输分为硫排放控制区 (SECA1, SECA2) 和非硫排放控制区, 非硫排放控制区分为 S1、S2、S3 共 3 段, 详见图 2。

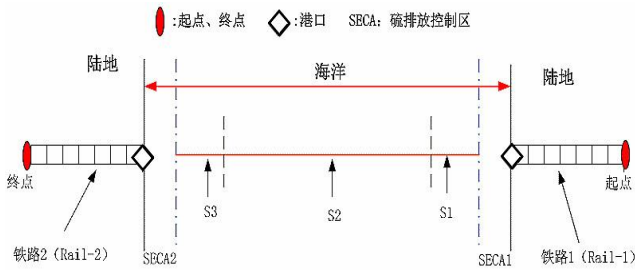


图 2 海铁联运运输路线示意图

Fig. 2 Sea-rail combined transport route diagram

5.3 目标函数建立

在本文中, 假设一艘船舶的吨位是火车的 20 倍, 船舶低硫燃料的价格是高硫燃料价格的 1.5 倍。根据 5.1 和 5.2 节的介绍, 能量损耗 T (h) 与运输时间 E (kg) 表达式如下:

$$T = \frac{S_{ship}^{SECA1}}{v_{ship}^{SECA1}} + \frac{S_{ship}^{SECA2}}{v_{ship}^{SECA2}} + \frac{S_{ship}^{S1}}{v_{ship}^{S1}} + \frac{S_{ship}^{S2}}{v_{ship}^{S2}} + \frac{S_{ship}^{S3}}{v_{ship}^{S3}} + \frac{S_{train}^{rail1}}{v_{train}^{rail1}} + \frac{S_{train}^{rail2}}{v_{train}^{rail2}} + 2 \times t_{transit} \quad (16)$$

$$E = \frac{0.0043 \times (v_{ship}^{SECA1})^{3.358} \times S_{ship}^{SECA1} \times 1000 \times 1.5}{24 \times v_{ship}^{SECA1}} + \frac{0.0043 \times (v_{ship}^{SECA2})^{3.358} \times S_{ship}^{SECA2} \times 1000 \times 1.5}{24 \times v_{ship}^{SECA2}} + \frac{0.0043 \times (v_{ship}^{S1})^{3.358} \times S_{ship}^{S1} \times 1000}{24 \times v_{ship}^{S1}} + \frac{0.0043 \times (v_{ship}^{S2})^{3.358} \times S_{ship}^{S2} \times 1000}{24 \times v_{ship}^{S2}} + \frac{0.0043 \times (v_{ship}^{S3})^{3.358} \times S_{ship}^{S3} \times 1000}{24 \times v_{ship}^{S3}} + \frac{\rho \times P_{train}^{rail1} \times S_{train}^{rail1} \times 20}{1000 \times v_{train}^{rail1}} + \frac{\rho \times P_{train}^{rail2} \times S_{train}^{rail2} \times 20}{1000 \times v_{train}^{rail2}} \quad (17)$$

$$s. t. \quad 4knot / h \leq v_{ship}^{SECA1}, v_{ship}^{SECA2} \leq 8knot / h$$

$$8knot / h \leq v_{ship}^{S1}, v_{ship}^{S3} \leq 15knot / h$$

$$15knot / h \leq v_{ship}^{S2} \leq 20knot / h$$

$$30km / h \leq v_{train}^{rail1}, v_{train}^{rail2} \leq 100km / h$$

其中: $t_{transit}=14hr$ 是海铁运输转换时间; $v_{train}^{rail1}, v_{train}^{rail2}$ 表示铁路 1 和 2 的运输速度; $v_{ship}^{SECA1}, v_{ship}^{SECA2}$ 表示船舶在硫排放控制区 1 和 2 的航速; $v_{ship}^{S1}, v_{ship}^{S2}, v_{ship}^{S3}$ 表示船舶在非硫排放控制区 S1、S2、S3 的航速; $S_{ship}^{rail1} = S_{ship}^{rail2} = 1200km$ 表示铁路运输路程; $S_{ship}^{SECA1} = S_{ship}^{SECA2} = 150knts$ 表示硫排放控制区海运路程; $S_{ship}^{S1} = S_{ship}^{S3} = 300knts$ 表示非硫排放控制区 S1、S3 的海运路程; $S_{ship}^{S2} = 700knts$ 表示非硫排放控制区 S2 的海运路程。

根据文献^[22~24], 能量损耗 E 与运输时间 T 的双目标优化建立如下:

$$\min_{v \in \Omega} F(v) = \{f_1(v), f_2(v)\}, \quad (18)$$

其中: $f_1(v) = E; f_2(v) = T; v$ 表示火车速度和船的航速。

5.4 实验结果

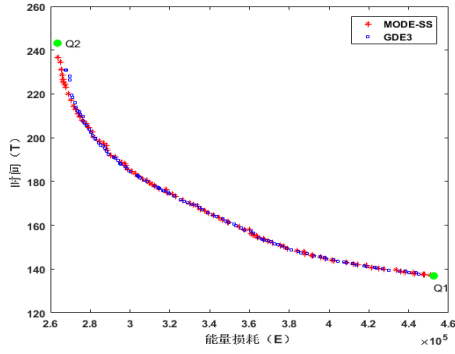
为解决海铁联运能耗优化模型, 选取本文算法 MODE-SS 与其他 6 种多目标优化算法对其进行求解, 种群数量 100, 迭代次数 250, 求解结果见表 5。其中, 真实 Pareto 前沿由 7 个算法共同确定。从实验结果表 5 可知, 本文所提算法在 GD 与 IGD 结果值上均优于算法 GDE3、NAGSII-DE、MODE-RMO、MOMDE-AM 和 MODE-PMSMO, 具有较好的收敛性与分布性; 其中 MODE-SS 在 GD 结果值上与算法 SA-MTLBO 相差无几, 但在 IGD 结果值上劣于算法 SA-MTLBO。从 4.2 节测试结果可以看出, SA-MTLBO 具有较强的局部搜索性能, 因此 SA-MTLBO 在求解双目标问题上性能更强。另外, 7 种算法求解海铁联运能耗优化问题所得 PF 见图 3。从图 3 中也可以看出, MODE-SS 优于算法 GDE3、NAGSII-DE、MODE-RMO、MOMDE-AM 和 MODE-PMSMO, 但劣于 SA-MTLBO。从 MODE-SS 的求解结果中, 分别选取燃料消耗最大的点 Q1 与运输时间最长的点 Q2, Q1 与 Q2 点见图 3(a)。Q1 与 Q2 点火车与船舶在各路段的速度与航速见表 6。从表 6 可以看出当能耗最大时, 各路段的行驶速度相对较高; 当运输时间最长时, 各路段行驶速度远小于于能量消耗最大时的行驶速度。另外, 对比 Q1 与 Q2 点各路段速度发现, 铁路运输速度的变化高于船舶运输速度的变化, 且铁路速度的变化对运输时间有显著影响。决策者可以根据所得结果和具体情况, 合理控制能耗成本与运输时间的矛盾关系。

表 5 各种算法求解模型得到的 GD 与 IGD 值
Table 5 GD and IGD obtained by different moeas solving model

	GDE3	NSGAI-DE	MODE-RMO	SA-MTLBO	MOMDE-AM	MODE-PMSMO	MODE-SS
GD	158.2674	151.7631	148.789	132.0126	158.0727	143.8455	136.7878
IGD	277.1261	134.5324	178.2917	79.1868	130.2743	161.1309	107.5755

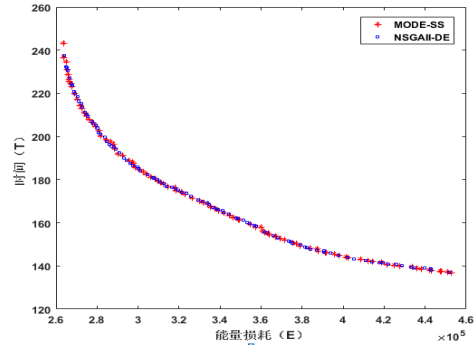
表 6 Q1 与 Q2 在各路段的速度
Table 6 Speed of Q1 or Q2 in different routes

	耗能 (kg)	时间 (h)	铁路 1(km/h)	SECA1(knot/h)	S1(knot/h)	S2(knot/h)	S3(knot/h)	SECA2(knot/h)	铁路 2 (km/h)
Q1	452508.14	136.92	99.7	7.98	15	19.91	14.84	8	99.98
Q2	263544.16	243.19	39.36	5	8.06	15	8.6	4.37	40.47



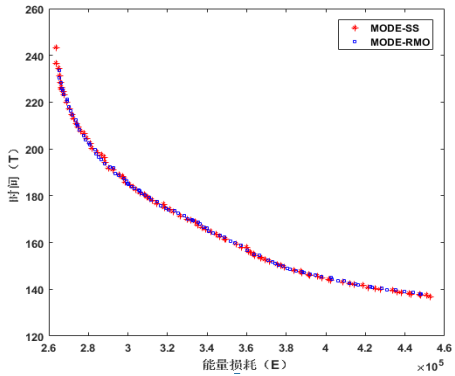
(a) MODE-SS 与 GDE3 Pareto 解集对比

(a) Comparison of PF obtained by MODE-SS and GDE3



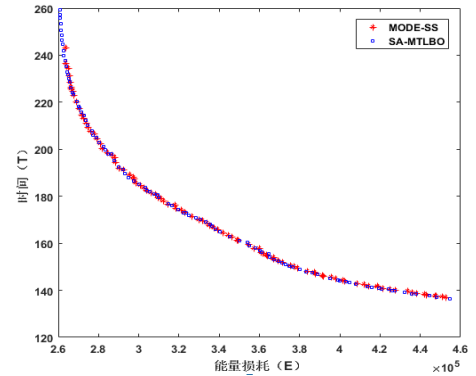
(b) MODE-SS 与 NSGAI-DE Pareto 解集对比

(b) Comparison of PF obtained by MODE-SS and NSGAI-DE



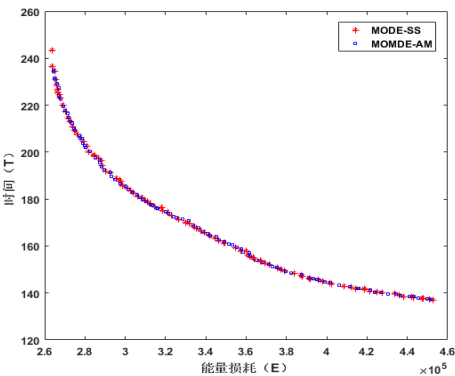
(c) MODE-SS 与 MODE-RMO Pareto 解集对比

(c) Comparison of PF obtained by MODE-SS and MODE-RMO



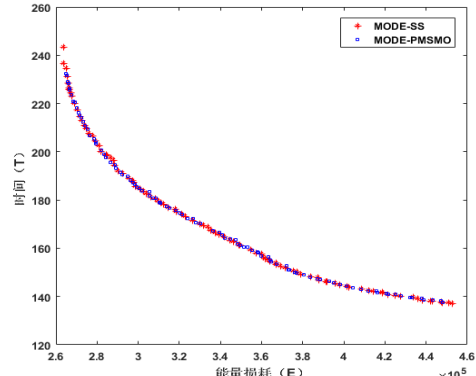
(d) MODE-SS 与 SA-MTLBO Pareto 解集对比

(d) Comparison of PF obtained by MODE-SS and SA-MTLBO



(e) MODE-SS 与 MOMDE-AM Pareto 解集对比

(e) Comparison of PF obtained by MODE-SS and MOMDE-AM



(f) MODE-SS 与 MODE-PMSMO Pareto 解集对比

(f) Comparison of PF obtained by MODE-SS and MODE-PMSMO

图 3 MODE-SS 与其他 6 中算法求解模型所得 PF 对比

Fig. 3 Comparison of PF obtained by MODE-SS and other 6 algorithms solving models

6 结束语

为提高多目标差分进化算法求解多目标优化问题的能力, 本文提出一种基于策略自适应的多目标差分进化 (MODE-SS) 算法。在 MODE-SS 中, 使用 HV 评价每个变异策略的性能, 并实现变异策略的自适应; 同时采用二项交叉策略和模拟二

进制交叉策略来平衡全局搜索和局部搜索。通过与其他多目标优化算法对比实验, 结果显示 MODE-SS 的整体性能优于其他算法。最后, MODE-SS 被用于解决海铁联运能耗优化问题, 其计算结果提供了一组能耗 (E) 与时间 (T) 的 Pareto 解集, 有助于决策者作出不同的运输策略以适应不同要求。

参考文献:

- [1] Kukkonen S, Lampinen J. GDE3: The third evolution step of generalized differential evolution [C]// Proc of IEEE Congress on Evolutionary Computation. 2005: 443-450.
- [2] Li Hui, Zhang Qingfu. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2009, 13 (2): 284-302.
- [3] Lin Qiuzhen, Ma Yueping, Chen Jianyong, *et al.* An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies [J]. Information Sciences, 2018, 430: 46-64.
- [4] 侯莹, 韩红桂, 乔俊飞. 基于参数动态调整的多目标差分进化算法 [J]. 控制与决策, 2017, 32 (11): 1985-1990. (Hou Ying, Han Honggui, Qiao Junfei. Adaptive multi-objective differential evolution algorithm based on the dynamic parameters adjustment [J]. Control and Decision, 2017, 32 (11): 1985-1990.)
- [5] 艾兵, 董明刚, 敬超. 基于多策略排序变异的多目标差分进化算法 [J]. 计算机应用研究, 2018, 35 (7): 36-40. (Ai Bing, Dong Minggang, Jing Chao. Multi-objective differential evolution algorithm with multi-strategy and ranking-based mutation [J]. Application Research of Computers, 2018, 35 (7): 36-40.)
- [6] 童旅杨, 董明刚, 敬超. 基于分解和多策略变异的多目标差分进化算法 [J]. 计算机应用研究, 2019, 36 (7) . (Tong Lyuyang, Dong Minggang, Jing Chao. Multiobjective differential evolution algorithm based on decomposition and multi-strategy mutation [J]. Application Research of Computers, 2019 (7).
- [7] Zhu Qingling, Lin Qiuzhen, Du Zhihua, *et al.* A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm [J]. Information Sciences, 2016, 345: 177-198.
- [8] Fan Qinqin, Wang Weili, Yan Xuefeng. Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems [J]. Applied Soft Computing, 2017, 59: 33-44.
- [9] Wang Jiahai, Zhang Weiwei, Zhang Jun. Cooperative differential evolution with multiple populations for multiobjective optimization [J]. IEEE Trans on Cybernetics, 2015, 46 (12): 2848-2861.
- [10] 王亚辉, 吴金妹, 贾晨辉. 基于动态种群多策略差分进化模型的多目标进化算法 [J]. 电子学报, 2016, 44 (6): 1472-1480. (Wang Yahui, Wu Jinmei, Jian Chenhui. Multi-objective evolutionary algorithm based on dynamic population multi-strategy differential models [J]. Acta Electronica Sinica, 2016, 44 (6): 1472-1480.)
- [11] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.
- [12] 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.
- [13] Zitzler E, Thiele L, Laumanns M, *et al.* Performance assessment of multiobjective optimizers: an analysis and review [J]. IEEE Trans on Evolutionary Computation, 2003, 7 (2): 117-132.
- [14] Van V D A, Lamont G B. Multiobjective evolutionary algorithm research: a history and analysis , Technical Report TR-98-03 [R]. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, 1998.
- [15] Chen Xu, Du Wenli, Qian Feng. Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization [J]. Chemometrics & Intelligent Laboratory Systems, 2014, 136 (16): 85-96.
- [16] Yu Kunjie, Wang Xin, Wang Zhenlei. Self-adaptive multi-objective teaching-learning-based optimization and its application in ethylene cracking furnace operation optimization [J]. Chemometrics & Intelligent Laboratory Systems, 2015, 146: 198-210.
- [17] Fan Qinqin, Yan Xuefeng. Multi-objective modified differential evolution algorithm with archive-base mutation for solving multi-objective p-xylene oxidation process [J]. Journal of Intelligent Manufacturing, 2015: 1-15.
- [18] Wilcoxon F. Individual comparisons by ranking methods [J]. Biometrics Bulletin, 1945, 1 (6): 80-83.
- [19] Friedman M. A Correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance [J]. Publications of the American Statistical Association, 1939, 32 (200): 675-701.
- [20] Doudnikoff M, Lacoste R. Effect of a speed reduction of containerships in response to higher energy costs in sulphur emission control areas [J]. Transportation Research Part D, 2014, 28 (5): 51-61.
- [21] Cullinane K, Bergqvist R. Emission control areas and their impact on maritime transport [J]. Transportation Research Part D, 2014, 28 (5): 1-5.
- [22] 许冬敏, 丁一, 黄有方, 等. 考虑能耗的国际海铁联运速度优化模型 [J]. 上海海事大学学报, 2015, 36 (2): 20-24. (Xu Dongmin, Ding Yi, Huang Youfan, *et al.* A model on speed optimization for sea-rail intermodal transportation considering energy consumption [J]. Journal of Shanghai Maritime University, 2015, 36 (2): 20-24.)
- [23] Wang S, Meng Q. Sailing speed optimization for container ships in a liner shipping network [J]. Transportation Research Part E, 2012, 48 (3): 701-714
- [24] Fan Qinqin, Jin Yaochu, Wang Weili, *et al.* A performance-driven multi-algorithm selection strategy for energy consumption optimization of sea-rail intermodal transportation [J]. Swarm and Evolutionary Computation, 2019, 44: 1-17.