

自适应离散差分进化算法策略的选择*

薛羽^{1,2}, 庄毅³, 顾晶晶³, 常相茂³, 王洲⁴

¹(南京信息工程大学 江苏省网络监控中心, 江苏 南京 210044)

²(南京信息工程大学 计算机与软件学院, 江苏 南京 210044)

³(南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016)

⁴(中国航空工业集团公司 洛阳光电设备研究所, 河南 洛阳 471009)

通讯作者: 薛羽, E-mail: xueyu_123@nuaa.edu.cn

摘要: 根据自适应离散差分进化(SaDDE)算法的提出过程,对算法策略选择问题进行了重点研究.策略池在SaDDE中起着重要作用,策略池的设计面临着3个问题,即:(1)怎样鉴别某个候选解产生策略(CSGS)是有效的还是无效的;(2)应该选择哪些CSGS组成策略池;(3)策略池的大小应该是多少.为了解决这些问题,提出了基于相对排列顺序的标度法(RPOSM)和基于RPOSM的层次分析法(RPOSM-AHP).主要采用某电子对抗(electronic countermeasure,简称ECM)仿真实验平台上的6个测试实例(T_INS)进行测试实验.首先,设计了144个不同的CSGS,为了获得这些CSGS在求解问题上的性能排序序列,做了144×6个独立的实验;然后,采用RPOSM和RPOSM-AHP计算这144个CSGS的最终优先级向量;接着,设计了16个具有不同策略池大小的算法,然后在同样的6个测试实例上测试这些算法的性能;最后,再一次采用RPOSM和RPOSM-AHP为SaDDE寻找到了合适的策略池大小.与其他类似算法的对比实验结果表明:在有限的评估次数(NFE)内,SaDDE比同类算法性能优越.

关键词: 智能计算;离散优化;自适应;离散差分进化;层次分析法;决策;协同干扰;武器目标分配

中图法分类号: TP181

中文引用格式: 薛羽,庄毅,顾晶晶,常相茂,王洲.自适应离散差分进化算法策略的选择.软件学报,2014,25(5):984-996. <http://www.jos.org.cn/1000-9825/4448.htm>

英文引用格式: Xue Y, Zhuang Y, Gu JJ, Chang XM, Wang Z. Strategy selecting problem of self-adaptive discrete differential evolution algorithm. Ruan Jian Xue Bao/Journal of Software, 2014, 25(5): 984-996 (in Chinese). <http://www.jos.org.cn/1000-9825/4448.htm>

Strategy Selecting Problem of Self-Adaptive Discrete Differential Evolution Algorithm

XUE Yu^{1,2}, ZHUANG Yi³, GU Jing-Jing³, CHANG Xiang-Mao³, WANG Zhou⁴

¹(Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China)

²(School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China)

³(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

⁴(Luoyang Institute of Optic-Electronic Equipment, Aviation Industry Corporation of China, Luoyang 471009, China)

Corresponding author: XUE Yu, E-mail: xueyu_123@nuaa.edu.cn

Abstract: In line with the proposing process of the self-adaptive discrete differential evolution (SaDDE) algorithm, this research focuses on the strategy selection problem. The strategy pool plays a significant role in the SaDDE algorithm, and there are three issues need to be

* 基金项目: 国家自然科学基金(61202351, 61202350); 国防基础研究基金(Q072006C002-1); 航空科学基金(2010ZC13012); 江苏省普通高校研究生科研创新计划(CXLX11_0203); 江苏高校优势学科建设工程资助项目; 南京信息工程大学科研启动费(2013x034)

收稿时间: 2012-10-28; 修改时间: 2013-01-07; 定稿时间: 2013-06-09

addressed in designing the strategy pool: (1) how to determine if a candidate solution generating strategy (CSGS) is effective; (2) which CSGSes to choose to constitute the strategy pool; and (3) how to find a suitable size for the strategy pool. In order to solve these problems, a relative permutation order based scale method (RPOSM) and a RPOSM based analytic hierarchy process (RPOSM-AHP) are proposed in this paper. The experiments are mainly conducted on six test instances (T_INSes) which come from an electronic countermeasure (ECM) simulation experimental platform. 144 different CSGSes are designed, and 144×6 independent experiments are performed to obtain the sort sequences of the CSGSes. The RPOSM and the RPOSM-AHP are adopted to obtain the priority vector of the 144 CSGSes. Sequentially, 16 algorithms with different sizes of strategy pools are constructed and their performance is tested on the six T_INSes. Further, the RPOSM and RPOSM-AHP are employed again to find the suitable pool size for the SaDDE algorithm. Computational comparisons demonstrate that, within fixed number of fitness evaluations (NFE), the SaDDE algorithm can generate better results than its competitors.

Key words: computational intelligence; discrete optimization; self-adaptive; discrete differential evolution; analytic hierarchy process; decision making; cooperative jamming; weapon-target assignment

进化算法(evolutionary algorithm,简称 EA)求解某一问题的成功主要依靠候选解产生策略(candidate solution generating strategy,简称 CSGS)及其相关参数的选择.因此,许多研究工作集中在算法策略选择和参数的改进上.尽管这种方法被实验证实可以提升 EA 性能,但仅仅依靠策略选择和参数改进使算法性能提升的空间越来越小.即使某种策略的效能被全面地发挥出来,仅仅依靠一种策略也不大可能有效解决所有的问题.另一方面,采用不断尝试和调节的方法寻找合适的策略及其相关参数值是十分耗时的,在进化过程的不同阶段,采用不同的策略以及不同的参数值也许会更有效.因此,大量研究开始采用自适应机制增强各种算法的性能^[1-6].最近,文献[1]提出了一种启发式自适应离散差分进化算法,但文献[1]没有对 CSGS 选择问题进行深入的研究.显然,策略池在算法中起着重要作用,好的策略池可以决定算法的主要性能,好的策略池中应能包含不同特性的各种有效策略.文献[7]指出:策略池大小以及策略的选择值得进一步的研究,但该文未对 CSGS 选择问题做深入的研究.因此,本文结合简化的协同干扰武器目标分配(cooperative jamming weapon-target assignment,简称 CJWTA)^[1]应用问题模型对自适应算法策略选择问题展开深入的研究.

本文提出自适应离散差分进化(self-adaptive discrete differential evolution,简称 SaDDE)算法求解 CJWTA 问题,与文献[1]相同的是,本文称变异向量产生操作和审查向量产生操作为 CSGS.根据没有免费的午餐(no free lunch,简称 NFL)理论^[8],在求解不同问题时,不同的 CSGS 性能不同.因此,SaDDE 没有采用不断尝试的方案寻找合适的 CSGS 及其参数值,而是维持了一个包含多种有效 CSGS 的策略池,并且在算法中引入了自适应机制,在进化过程中,针对每一个目标向量,根据 CSGS 学习到的概率在候选策略池中选择 CSGS,并且自适应地改变其相关参数值,采用最终的 CSGS 为目标向量产生审查向量^[1].

为了设计策略池,有 3 个问题需要解决,即:(1) 怎样鉴别某个 CSGS 是有效的还是无效的;(2) 应该选择哪些 CSGS 组成策略池;(3) 策略池的大小应该是多少.

这些问题在现有文献中没有涉及.据我们所知,目前国内外还没有文献对上述问题做相关报道,本文将在这方面做初步的研究工作.本文将前述 3 个问题称作 CSGS-SP(CSGS selecting problem).本文在现有文献工作的基础上深入地研究了 CSGS,并设计了 144 种不同的 CSGS.根据 NFL 理论,几乎不可能找到一种在所有问题上都优于其他策略的 CSGS.因此,怎样评估 CSGS 的性能就成为亟需解决的问题,本文结合层次分析法(analytic hierarchy process,简称 AHP)^[9-11]提出了 CSGS 的性能评估方法.

AHP 是 Saaty 教授提出的一种定性和定量相结合的、层次化、系统化的多准则决策分析方法^[9,10].AHP 的特点是在对复杂问题的本质、影响因素及其内在关系深入分析的基础上,利用较少的定量信息使决策的思维过程数学化、系统化,从而为复杂决策问题提供了一种简便的评价决策方法.本文结合自适应离散差分进化算法的提出过程对算法策略选择问题进行了重点研究.首先提出了基于相对排列顺序的标度法(relative permutation order based scale method,简称 RPOSM),然后提出了基于 RPOSM 的层次分析法(RPOSM based analytic hierarchy process,简称 RPOSM-AHP).实验在某电子对抗仿真系统实验平台上进行,设计了 6 个测试实例(test instances,简称 T_INS).为了获得 144 个 CSGS 在 6 个测试实例上的排序序列做了 864 组实验,通过实验结果数据的对比,获

得了这 144 个 CSGS 在每个测试实例上的排序序列;然后,采用 RPOSM 和 RPOSM-AHP 获得了 144 个 CSGS 的层次总排序序列;在此基础上,设计了具有不同大小策略池的 16 种算法,并且在前述 6 个 T_INS 上做算法性能测试对比实验,在实验数据的基础上再一次采用 RPOSM 和 RPOSM-AHP 为 SaDDE 找到了合适的策略池大小;最终,为了评估 SaDDE 算法的性能,将其和相关文献最近提出的两种离散差分进化算法和求解 CJWTA 问题的一种相关算法在更多的测试实例上做了算法性能对比实验.实验结果表明,SaDDE 在求解质量和求解速度上具有较为明显的优势.

1 CJWTA 问题数学模型及自适应离散差分进化算法简述

假设作战场景中有 w 部干扰设备和 t 部目标雷达.定义 $FITN^{J_i-R_j}$ ($i=1,2,\dots,w; j=1,2,\dots,t$)表示将干扰设备 J_i 分配给目标雷达 R_j 的设备干扰适应度^[1]. $FITN^{J_i-R_j}$ 是一个将 J_i 分配给 R_j 带来的期望收益和期望损失的综合变量.可根据 J_i 干扰 R_j 的合适程度将 $FITN^{J_i-R_j}$ 的值归一化到 $[0,1]$ 内.在给出 $FITN^{J_i-R_j}$ 的计算公式之前,首先引入有关的符号定义如下:

定义 1. 向量 $J=(J_1, J_2, \dots, J_w)$ 表示所有的干扰设备,向量 $R=(R_1, R_2, \dots, R_t)$ 表示所有的目标雷达.定义武器目标对“ J_i-R_j ”表示把干扰设备 J_i 分配给目标雷达 R_j .

定义 2. $B^{J_i-R_j} = (B_1^{J_i-R_j}, B_2^{J_i-R_j}, \dots, B_p^{J_i-R_j})$ 表示 J_i-R_j 的干扰期望收益向量, $B^{J_i-R_j}$ 包含 p 个子项,其中, $B_d^{J_i-R_j}$ ($d=1,2,\dots,p$)表示由子项 d 带来的期望收益.

定义 3. $C^{J_i-R_j} = (C_1^{J_i-R_j}, C_2^{J_i-R_j}, \dots, C_q^{J_i-R_j})$ 表示 J_i-R_j 的干扰期望损失向量, $C^{J_i-R_j}$ 包含 q 个子项,其中, $C_k^{J_i-R_j}$ ($k=1,2,\dots,q$)表示由子项 k 带来的期望损失.

定义 4. $W_1 = (w'_1, w'_2, \dots, w'_p)$ 表示期望收益权重向量, $W_2 = (w'_1, w'_2, \dots, w'_q)$ 表示期望损失权重向量. W_1 和 W_2 分别为 $B^{J_i-R_j}$ 和 $C^{J_i-R_j}$ 中各子项对应的权重向量.

武器目标对 J_i-R_j 的期望收益根据公式(1)计算:

$$E(B^{J_i-R_j}) = \begin{cases} 0, & \text{if } \exists d \in \{1, 2, \dots, p\}, B_d^{J_i-R_j} = 0 \\ (W_1, B^{J_i-R_j}) = \sum_{d=1}^p w'_d B_d^{J_i-R_j}, & \text{otherwise} \end{cases} \tag{1}$$

其中, $E(B^{J_i-R_j})$ 表示武器目标对 J_i-R_j 的干扰期望收益, $(W_1, B^{J_i-R_j})$ 是 W_1 和 $B^{J_i-R_j}$ 的内积.

武器目标对 J_i-R_j 的干扰期望损失根据公式(2)计算:

$$E(C^{J_i-R_j}) = (W_2, C^{J_i-R_j}) = \sum_{k=1}^q w'_k C_k^{J_i-R_j} \tag{2}$$

其中, $E(C^{J_i-R_j})$ 表示武器目标对 J_i-R_j 的干扰期望损失, $(W_2, C^{J_i-R_j})$ 是 W_2 和 $C^{J_i-R_j}$ 的内积.

然后, $FITN^{J_i-R_j}$ 则可以根据公式(3)计算:

$$FITN^{J_i-R_j} = (1-\nu)p_{sav}^{J_i-R_j} E(B^{J_i-R_j}) - \nu p_{des}^{J_i-R_j} E(C^{J_i-R_j}) \tag{3}$$

其中, $p_{sav}^{J_i-R_j}$ 是目标雷达 R_j 被成功干扰的概率, $p_{des}^{J_i-R_j}$ 是干扰设备 J_i 被摧毁的概率.参数 $\nu \in [0,1]$ 是平衡因子,可以平衡对期望损失的抑制和对期望收益的接受程度.

最后,CJWTA 问题可被建模成具有多约束条件的非线性整数规划模型,模型如下所示:

$$F(\pi) = \text{Max} \sum_{j=1}^t \left\{ \left[\sum_{i=1}^w FITN^{J_i-R_j} x_{ij} \right] \cdot \left[1 - \prod_{i=1}^w (1 - p_{sav}^{J_i-R_j}) x_{ij} \right] \right\} \tag{4}$$

$$\text{subject to: } f_{R_j} \subseteq \bigcup_{i=1}^w f_{J_i} x_{ij}, j = 1, 2, \dots, t \tag{5}$$

$$J_{iHstyle} \cap R_{jEstyle} \neq \emptyset, \text{ if } x_{ij} = 1 \tag{6}$$

$$\bigcap_{i=1}^w (J_{iHstyle} x_{ij}) \in R_{jEstyle}, j=1,2,\dots,t \quad (7)$$

$$\sum_{i=1}^w x_{ij} \geq 1, j=1,2,\dots,t \quad (8)$$

$$\sum_{j=1}^t x_{ij} = 1, i=1,2,\dots,w \quad (9)$$

$$x_{ij} \in \{0,1\}, i=1,2,\dots,w, j=1,2,\dots,t \quad (10)$$

公式(4)中: π 表示分配方案,是目标雷达编号的排列,即 $\pi=\{\pi(1),\pi(2),\dots,\pi(t)\},\pi(j)(j=1,2,\dots,t)$ 是某个目标雷达的编号, $\pi(j)$ 的位置顺序表示相应编号的干扰设备; $F(\cdot)$ 是目标函数; x_{ij} 是决策变量.公式(5)表示干扰设备的工作频段需覆盖目标雷达的工作频段, f_{R_j} 是目标雷达 R_j 的当前工作频段, f_{J_i} 是干扰设备 J_i 的整个工作频段.公式(6)表示分配给目标雷达的干扰设备需具有对目标雷达的有效干扰样式, $J_{iHstyle}$ 是干扰设备 J_i 具有的干扰样式集合, $R_{jEstyle}$ 是对 R_j 来说有效干扰样式集合.公式(7)表示干扰设备需在干扰样式上一致.公式(8)表示一部目标雷达至少应被分配一部干扰设备.公式(9)表示一部干扰设备在同一部时刻只能干扰一部雷达目标.

DDE(discrete differential evolution)算法是一种基于群体的随机搜索优化技术^[12],其中,每个个体 $\pi_i=(\pi_i(1),\pi_i(2),\dots,\pi_i(D))(i=1,2,\dots,NP, NP$ 是种群大小)是 D 个不同整数的排列, D 为实际问题的规模(本文中的 D 等于目标数目,即 $D=t$).从包含 NP 个随机初始化个体的群体 $POP=\{\pi_1,\pi_2,\dots,\pi_{NP}\}$ 开始,DDE算法相继采用变异个体产生操作和审查个体产生操作产生新个体;然后,采用一对一的选择方案在父个体和子个体中选择较优的个体进入下一代,这一过程不断循环直到满足停机准则.

本文研究重点在于算法策略选择的研究,因此对问题和算法根据文献[1]做了适当的简化.与文献[1]有所不同的是:本文采用直接编码方案,问题的解被编码成目标雷达序号的排列向量.本文的假设是干扰设备数目等于目标雷达数目,当目标雷达数目大于干扰设备数目时可以构造虚假武器,虚假武器对目标没有任何作用,将被分配给威胁较低的目标.同样地,当干扰设备数目大于雷达目标数目时,可以构造虚假目标.因此, $w \neq t$ 的问题可以转化为 $w=t$ 的问题求解.此外,SaDDE算法中候选解产生策略自适应和交叉概率自适应过程见文献[1]的相关内容.

2 算法策略池设计

CSGS由变异向量产生操作和审查向量产生操作组成,同时,可有3种不同的方式产生变异向量,可有两种方式产生审查向量.

2.1 候选解产生策略

2.1.1 变异向量产生操作

文献[12]采用公式(11)所示的操作产生当前代 g 的第 $i(i=1,2,\dots, NP)$ 个变异个体:

$$V_i^g = \begin{cases} F_k(\pi_{gbest}^{g-1}), & \text{if } r < P_m \\ \pi_{gbest}^{g-1}, & \text{otherwise} \end{cases} \quad (11)$$

其中, π_{gbest}^{g-1} 是上一代目标向量群体中适应度最好的个体, $P_m \in [0,1]$ 是扰动概率, F_k 是具有扰动强度 k 的扰动操作, r 是 $[0,1]$ 上的随机数.为了使策略适用于自适应学习机制,本文舍弃了扰动强度 k 和扰动概率 P_m .并且产生第 i 个变异个体 V_i^g 的目标个体不是固定的.

为了实现变异个体产生过程,可直接采用文献中存在的3个操作,即RandExMut^[13],ReverseMut^[12]和InsertMut^[12].此外,Tasgetiren提出了Binary swap(Bswap)^[14],但由于Bswap的二进制特性而不能被本文算法直接采用.因此,本文提出了Modified Bswap(MBswap),其步骤如下:

Step 1. 在范围 $[1,D]$ 内产生两个随机整数 u ;

Step 2. 寻找满足 $\pi_i^{g-1}(v) = \pi_i^{g-1}(u) \bmod D+1$ 的随机整数 v ,然后令 $\pi_i^{g-1}(v) := \pi_i^{g-1}(u)$;

Step 3. 令 $\pi_i^{g-1}(u) := \pi_i^{g-1}(u) \bmod D+1$.

2.1.2 审查向量产生操作

继变异操作之后,审查向量可通过公式(12)所示方式获得^[12]:

$$U_i^g = \begin{cases} CX(V_i^g, \pi_i^{g-1}), & \text{if } r < CR \\ V_i^g, & \text{otherwise} \end{cases} \quad (12)$$

其中,CX是交叉操作,CR是交叉概率.

OX^[15]和 PartMapXover(PMX)^[16]是常见的两种审查向量产生操作.除此之外,为了使算法具有多样性,本文提出和采用的 MTwoCutPTL 操作、RandTwoCutPTL 操作、OrderBasedOX 操作和 PosBasedOX 操作如下:

(1) MTwoCutPTL 操作

文献[14]提出了 two-cut PTL 操作,然而它是二进制特性的,因此,本文提出了修改的 two-cut PTL 操作 (MTwoCutPTL).在 MTwoCutPTL 中,首先采用 2 个随机整数在第 1 个父个体中定义一个整数块,然后将块复制到子个体的左端或右端,接着采用第 2 个父个体的不同整数块填充子个体,最后以相等的概率在 2 个不同的子个体中随机选择一个子个体作为后代.

MTwoCutPTL 操作的一个说明例子见表 1(整数块的位置为 2 和 4 之间).

Table 1 An instance of MTwoCutPTL

表 1 MTwoCutPTL 操作举例

Improved-Two-Cut PTL crossover										Improved-Two-Cut PTL crossover											
P_1	2	6	8	10	7	3	9	1	5	4	P_1	5	4	1	9	8	6	7	10	2	3
P_2	5	4	1	9	8	6	7	10	2	3	P_2	5	4	1	9	8	6	7	10	2	3
O_1	5	4	1	9	7	2	3	6	8	10	O_1	5	8	6	7	10	2	3	4	1	9
O_2	6	8	10	5	4	1	9	7	2	3	O_2	4	1	9	5	8	6	7	10	2	3

(2) RandTwoCutPTL 操作

为了使新产生的个体获得更多的多样性,提出了 MTwoCutPTL 的改进操作,称为 RandTwoCutPTL,具体实现如下:

- Step 1. 通过随机选择的两个位置 u 和 v 在第 1 个父个体中产生一个数字块.
- Step 2. 从第 2 个个体中去掉与第 1 个父个体中数字块相同的数字.
- Step 3. 子个体继承第 2 个父个体中的剩余数字.
- Step 4. 在子个体中随机选择一个位置,然后插入第 1 个父个体中产生的数字块.

(3) OrderBasedOX 操作

- Step 1. 随机选择一个位置的集合,并记录第 1 个父个体中这些位置上的数字.
- Step 2. 在第 2 个父个体中,把位置集合中记录的数字按升序排列.
- Step 3. 将按升序排列的数字串复制到子个体的相应的位置,其余数字直接从第 2 个父个体继承.

(4) PosBasedOX 操作

- Step 1. 随机产生一个位置的集合.
- Step 2. 在子个体中继承第 1 个父个体中位置集合中的数字串.
- Step 3. 在第 2 个父个体中删除子个体中已有数字串.
- Step 4. 将第 2 个父个体中剩余的字符串从左到右依次插入到子个体中.

2.1.3 选择操作

选择操作主要选择 U_i^g 和 π_i^{g-1} 中适应度较好的个体进入到下一代,描述如公式(13):

$$\pi_i^g = \begin{cases} U_i^g, & \text{if } f(U_i^g) \leq f(\pi_i^{g-1}) \\ \pi_i^{g-1}, & \text{otherwise} \end{cases} \quad (13)$$

其中, $f(U_i^g)$ 和 $f(\pi_i^{g-1})$ 分别是 U_i^g 和 π_i^{g-1} 的适应度.

2.1.4 启发式规则数据库

在 SaDDE 算法中,在选择操作之前,为了处理约束条件,在每个 CSGS 中嵌入了约束条件满足检查过程.某协同电子对抗仿真系统为算法设计了启发式规则数据库,其中包含我方干扰设备对目标雷达的约束规则.嵌入式过程实现如下:

Step 1. 设置循环次数 $z=0$.

Step 2. while $z \neq t$ do

 采用选择的同一个 CSGS 产生审查向量;

$z:=z+1$.

Step 3. 检查审查向量是否满足约束条件,如果满足则停止;否则,转向 Step 2.

2.1.5 144 个候选解产生策略

Tasgetiren 在变异个体产生操作的基础上提出了 3 种产生变异群体的方式^[14],这 3 种方式分别为

$$V_i^g = m_1 \oplus F_k(X_i^{g-1}) \quad (14)$$

$$V_i^g = m_1 \oplus F_k(X_a^{g-1}) \quad (15)$$

$$V_i^g = m_1 \oplus F_k(X^{g-1}) \quad (16)$$

其中, X_i^{g-1} 是第 i 个个体, X_a^{g-1} 是在目标群体中随机选择的个体, X^{g-1} 是上一代的最优个体, m_1 是变异概率.

此外,审查向量群体还可以通过以下两种不同的方式获得^[14]:

$$U_i^g = c_1 \oplus CR(X_i^{g-1}, V_i^g) \quad (17)$$

$$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1}) \quad (18)$$

因此,在前述各种变异向量产生操作、审查向量产生操作及其不同的产生方式的研究基础上,可设计 4 (mutated individual generating operators)×6(trial individual generating operators)×3(manners for obtaining mutant individual)×2(manners for obtaining trial individual)=144 种不同的 CSGS.

2.2 候选解策略池中 CSGS 的选择方法

2.2.1 基于相对排列顺序的标度法

RPOSM 的输入为一个排序序列向量,排序序列向量可以根据元素对决策结果的相对重要程度进行构造,RPOSM 的输出为每个元素的等级.准则层排序序列根据 CJWTA 问题的主观求解偏好进行构造,而属性层排序序列根据实验数据获得.

对 RPOSM 的简要描述如下:设 seq 表示根据各元素相对重要程度构造的排序序列向量,其中,第 K 个因素的等级计算如下:

$$o_K = OS - \lceil fac_K / GS \rceil \quad (19)$$

其中, o_K 表示第 K 个元素的等级; fac_K 表示第 K 个元素在序列 seq 中的序列号; GS 表示分组大小; OS 表示总的等级数,其值可以根据实际问题设定.

2.2.2 基于 RPOSM 的层次分析法

在 RPOSM 的基础上,提出了 RPOSM-AHP. RPOSM-AHP 分为 4 步,即:① 构造层次结构;② 构造判断矩阵并进行一致性检查;③ 层次单排序;④ 层次总排序.以 CSGS-SP 为例:

- ① 层次结构如图 1 所示. CSGS-SP 是最终目标,因此作为目标层;测试实例是影响 CSGS 选择的重要因素,因此作为准则层;而每种 CSGS 可被看作可选方案,因此作为属性层.
- ② 由公式(19)可计算出准则层和属性层各因素的等级,在此基础上,采用公式(21)、公式(22)得到准则层和属性层判断矩阵,然后根据公式(23)~公式(25)对判断矩阵进行一致性检查.
- ③ 在准则层和属性层判断矩阵的基础上进行层次单排序,即采用公式(26)、公式(27)得到各层相应因素对于上一层中某个因素的相对排序.
- ④ 综合准则层和属性层排序,由公式(28)计算准则层和属性层对目标层的层次总排序,从而为选择 CSGS

提供决策依据.

RPOSM-AHP 在准则层引入了优先权赋值,可使 RPOSM-AHP 满足主观求解偏好,使 RPOSM-AHP 更加灵活适用;而属性层输入来自于实验数据,因此最终计算结果更客观.可以使 RPOSM-AHP 在决策中充分体现实验结果的重要性,而引入 RPOSM 能使 RPOSM-AHP 适用于多标度决策问题.结合策略选择问题,RPOSM-AHP 的详细步骤描述如下:

Step 1. 构造层次结构.

采用“选择较好的 CS GS”作为目标层.由于 CS GS 在测试实例上的表现对决策非常重要,因此,本文选择 N 个不同规模的实例作为准则层,并且按照其规模从小到大依次对其编号.在属性层,根据实验数据结果为每个测试实例求得 MS 个 CS GS 的排列序列.

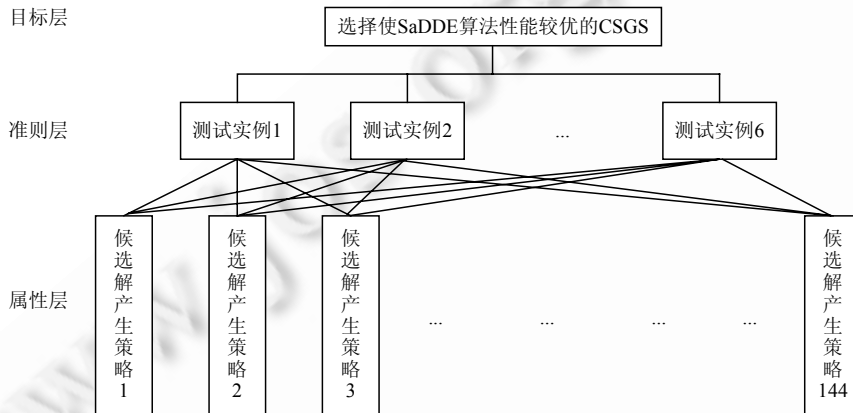


Fig.1 Hierarchical structure for the better CS GS selecting problem

图 1 较优 CS GS 选择问题的层次结构

Step 2. 构造判断矩阵和进行一致性检查.

Step 2.1. 构造判断矩阵.

首先,根据 CJWTA 问题的求解偏好将准则层的测试实例依次排列.其中,规模越大的实例在序列中的位置越靠前.对于每个测试实例,通过实验数据结果的比较得到 CS GS 在每个测试实例上的排序序列.采用 RPOSM 计算属性层每个排序序列中各元素的等级号,然后为准则层和属性层建立判断矩阵,计算方法如下:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \dots & \dots & \dots & \dots \\ a_{s1} & a_{s2} & \dots & a_{ss} \end{bmatrix} \tag{20}$$

设 o_i 和 o_j 分别表示排序序列 seq 中元素 i 和元素 j 的等级号 ($i=1,2,\dots,s, i \leq j \leq s$),对于矩阵 A 中 $\forall a_{ij}$,有以下计算公式:

$$a_{ij} = \begin{cases} o_i - o_j + 1, & \text{if } o_i \geq o_j \\ 1/(o_j - o_i + 1), & \text{otherwise} \end{cases} \tag{21}$$

$$a_{ji} = 1/a_{ij} \tag{22}$$

其中, a_{ij} 为矩阵 A 的第 i 行第 j 列元素, $a_{ii}=1$,且 $a_{ij}>0$;而 s 表示序列 seq 中元素的总数目.根据公式(21)和公式(22)计算,可得到准则层和属性层的判断矩阵.

Step 2.2. 一致性检查.

Step 2.2.1. 采用公式(23)^[9]计算每个判断矩阵的最大特征值:

$$\lambda_{\max} = (1/s) \cdot \left(\sum_{i=1}^s ((AW)_i / W_i) \right) \quad (23)$$

其中, λ_{\max} 表示判断矩阵 A 的最大特征值, W 是相应的特征向量, $(AW)_i$ 是向量 AW 的第 i 个元素.

Step 2.2.2. 一致性指标 CI 的计算公式见公式(24)^[9]:

$$CI = (\lambda_{\max} - s) / (s - 1) \quad (24)$$

Step 2.2.3. 一致性比率 CR 的计算公式见公式(25)^[9]:

$$CR = CI / RI \quad (25)$$

其中, RI 表示平均一致性指标. 当 $CR < 0.1$ 时, 认为判断矩阵的一致性是可以接受的^[9].

Step 3. 层次单排序.

对于 Step 2.1 中的判断矩阵, 层次单排序计算公式为

$$\bar{a}_{ij} = a_{ij} / \sum_{k=1}^s a_{kj}, i, j = 1, 2, \dots, s \quad (26)$$

$$W_i = \sum_{j=1}^s \bar{a}_{ij} / \sum_{k=1}^s \sum_{j=1}^s \bar{a}_{kj}, i = 1, 2, \dots, s \quad (27)$$

由公式(26)和公式(27)可以得到准则层排序序列向量 $C = (C_1, C_2, \dots, C_{NI})$ 和属性层排序序列向量:

$$W_i = (w_{i1}, w_{i2}, \dots, w_{iNS}), i = 1, 2, \dots, NI.$$

Step 4. 层次总排序.

在 Step 3 计算结果的基础上, 层次总排序序列向量计算方法见公式(28):

$$W' = \left(\sum_{i=1}^{NI} c_i w_{i1}, \sum_{i=1}^{NI} c_i w_{i2}, \dots, \sum_{i=1}^{NI} c_i w_{iNS} \right) \quad (28)$$

3 仿真实验

3.1 算法参数设置

本文主要采用 6 个 CJWTA 问题实例(T_INS)来测试算法的性能.

这 6 个 T_INS 来自于某协同电子对抗(electronic countermeasure, 简称 ECM)系统仿真实验平台. 这 6 个测试实例依次为 T_INS_1($w=10, t=10$), T_INS_2($w=30, t=30$), T_INS_3($w=80, t=80$), T_INS_4($w=120, t=120$), T_INS_5($w=180, t=180$) 和 T_INS_6($w=200, t=200$). W_1, W_2 和 ν 的值均根据领域专家经过大量的实测实验获得的经验设置.

硬件实验环境为 Intel(R) Core(TM) i3 PC with 3G; 所有的算法代码采用 Matlab 7.5 编写; 每种算法均在所有测试实例上运行 30 次, 取数据统计结果比较分析; 所有算法均采用最大评估次数(number of fitness evaluations, 简称 NFE)作为算法运行终止条件, 为了比较的公平, 设置 NFE 为 100 000; 为了编码方便, 采用 $1/F(\pi)$ 作为适应度函数. 算法 SaDDE 的参数设置如下: NP 等于每个测试实例的目标数; 初始时: $p_q = 0.25, CR_{mq} = 0.5 (q=1, 2, 3, 4)$; 根据初步实验经验, 设置 LP 为 10; 准则层 $OS=7, GS=1$; 属性层 $OS=10, GS=16, NI=6, NS=144$. 采用 $A_t (t=1, 2, \dots, 7)$ 表示准则层判断矩阵和属性层判断矩阵. 本文将设计的 144 个 CSGS 编号为 CS1~CS144, 具体编号方法不再赘述. 对于其他对比算法, 即 HDDE(hybrid discrete differential evolution)^[17], DABC(discrete artificial bee colony)^[18] 和 GA-GE(genetic algorithm with general eugenics)^[19], 文献[17, 19]分别做了算法参数选择实验, 在实验结果的基础上, 最终确定了各自的算法参数; 而文献[18]则直接给出了较优的参数值. 因此, 为了使其他对比算法具有较好的性能, 除 NFE 外, 其余参数均按原文献进行设置.

3.2 CSGS选择实验

为了获得 CSGS 在每个 T_INS 上的排序序列, 本文首先做了 CSGS 在每个 T_INS 上的初步数据获取实验, 即做了 144×6 组独立的实验, 然后通过对比实验结果获得了 144 个 CSGS 在每个 T_INS 上的排序序列. 为简短起见, 本文仅列出每个 T_INS 上的前 17 个 CSGS, 见表 2.

由于求解侧重于大规模 CJWTA 问题, 因此设置准则层的 6 个 T_INS 的排序序列为 [6 5 4 3 2 1]. 结合上述所得计算结果, 采用 RPOSM 法和 RPOSM-AHP 法可计算出 144 个 CSGS 的总层次优先选择排序序列. 为简短起见, 仅列出总排序序列中的前 17 个 CSGS, 见表 3. 其中, 计算所得各个指标数据为: A_1 的 λ_{\max} 为 6.123 2; CI 为 0.024

6;CR 为 $0.019 < 0.1$; $A_1(l=2,3,\dots,7)$ 的 λ_{\max} 均为 150.716 3,CI 为 0.047 0.

由于 15 阶矩阵的 RI 为 1.59,易知 144 阶矩阵的 RI 大于 1.59.因此有 $CR < 0.029 = 0.0470 / 1.59 < 0.1$,最终获得的计算结果是可以接受的.

Table 2 Sort sequences of the 144 CSGSs on each test instance
(Only the first 17 CSGSs are list for short)

表 2 144 个 CSGS 在每个测试实例上的排序序列(为简短起见,仅列出了前 17 个 CSGS)

Instances	CSGS																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T_INS_1	CS 8	CS 9	CS 13	CS 14	CS 26	CS 37	CS 41	CS 42	CS 55	CS 65	CS 69	CS 70	CS 97	CS 98	CS 121	CS 125	CS 126
T_INS_2	CS 120	CS 121	CS 124	CS 125	CS 92	CS 14	CS 13	CS 123	CS 12	CS 37	CS 65	CS 126	CS 11	CS 93	CS 10	CS 70	CS 68
T_INS_3	CS 39	CS 122	CS 95	CS 123	CS 38	CS 94	CS 98	CS 96	CS 97	CS 124	CS 125	CS 126	CS 92	CS 93	CS 120	CS 121	CS 136
T_INS_4	CS 38	CS 122	CS 123	CS 39	CS 94	CS 95	CS 98	CS 97	CS 96	CS 124	CS 125	CS 93	CS 126	CS 92	CS 121	CS 120	CS 52
T_INS_5	CS 123	CS 122	CS 38	CS 94	CS 95	CS 39	CS 97	CS 96	CS 98	CS 32	CS 31	CS 124	CS 125	CS 126	CS 120	CS 93	CS 92
T_INS_6	CS 123	CS 122	CS 38	CS 94	CS 95	CS 97	CS 96	CS 98	CS 39	CS 31	CS 32	CS 124	CS 125	CS 126	CS 93	CS 92	CS 121

Table 3 Final priority vector of the CSGS (Only the first 17 CSGSs are list for short)

表 3 CSGS 总排序序列(为简短起见,仅列出了前 17 个 CSGSs)

NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CSGS	CS 123	CS 122	CS 38	CS 94	CS 97	CS 95	CS 39	CS 98	CS 96	CS 124	CS 125	CS 126	CS 93	CS 92	CS 120	CS 121	CS 32

3.3 策略池大小寻找实验

对于引言部分提出的第 2 个问题,本文仅给出一个简单的解决方案,即将较好的 CSGS 组合形成策略池.关于候选解产生策略性质及组合的研究是一个非常值得研究的方向,也是一个难以解决的问题,本文仅提供一个简单的可行的解决方案.在表 3 所示结果的基础上,本文设计了具有不同策略池大小的 16 种算法,并将这 16 种算法从 ALG1 到 ALG16 依次编号.其中,第 1 种算法的策略池由表 3 的前两个策略组成,第 2 种算法的策略池由表 3 的前 3 个策略组成,依此类推,最后一种算法的策略池由表 3 的 17 个策略组成.本文做了这 16 种算法在前述 6 个 T_INS 上的性能测试实验.表 4 给出了这 16 种算法在这 6 个 T_INS 上求得的最优解平均适应度.

Table 4 Mean values of the best fitness obtained by the 16 algorithms on the 6 T_INSes

表 4 16 种算法在 6 个 T_INS 上求得的最优解平均适应度

Algorithms	Instances					
	T INS 1	T INS 2	T INS 3	T INS 4	T INS 5	T INS 6
ALG1	1.13973E-001	3.60941E-002	1.33433E-002	9.13105E-003	6.34837E-003	5.78446E-003
ALG2	1.12552E-001	3.57340E-002	1.32646E-002	9.04638E-003	6.37818E-003	5.79474E-003
ALG3	1.12552E-001	3.61184E-002	1.32766E-002	9.01134E-003	6.30750E-003	5.81897E-003
ALG4	1.12552E-001	3.64158E-002	1.31977E-002	8.95459E-003	6.28346E-003	5.75797E-003
ALG5	1.13010E-001	3.57793E-002	1.32093E-002	8.96635E-003	6.35690E-003	5.77649E-003
ALG6	1.13469E-001	3.58790E-002	1.30979E-002	8.72347E-003	6.10275E-003	5.61024E-003
ALG7	1.12552E-001	3.63262E-002	1.31241E-002	8.66510E-003	6.02456E-003	5.55310E-003
ALG8	1.12552E-001	3.62750E-002	1.31227E-002	8.65285E-003	5.93458E-003	5.45835E-003
ALG9	1.13010E-001	3.65154E-002	1.31065E-002	8.78159E-003	5.94276E-003	5.45978E-003
ALG10	1.16265E-001	3.63015E-002	1.30707E-002	8.66559E-003	5.99522E-003	5.51245E-003
ALG11	1.13973E-001	3.62402E-002	1.31221E-002	8.73546E-003	6.01745E-003	5.53326E-003
ALG12	1.15536E-001	3.64306E-002	1.31533E-002	8.71623E-003	5.96685E-003	5.48987E-003
ALG13	1.14139E-001	3.64084E-002	1.31526E-002	8.73930E-003	5.97169E-003	5.38497E-003
ALG14	1.12552E-001	3.60319E-002	1.31042E-002	8.75757E-003	6.02671E-003	5.40508E-003
ALG15	1.14319E-001	3.63836E-002	1.31221E-002	8.73485E-003	5.95439E-003	5.40994E-003
ALG16	1.13010E-001	3.62743E-002	1.30959E-002	8.73433E-003	6.17430E-003	5.46084E-003

通过观察表 4 可以看出:实际上很难找到一种算法在所有的测试实例上比其他算法都表现得更好,同时也很难确定哪种算法性能较好.因此,本文再一次采用 RPOSM 和 RPOSM-AHP 计算这 16 种算法的排序序列向量.计算所得算法排序向量为 (ALG8,ALG14,ALG13,ALG15,ALG9,ALG10,ALG12,ALG16,ALG7,ALG6,ALG11,ALG2,ALG4,ALG5,ALG3,ALG1), λ_{max} 为 17.8852,CI 为 0.1257.易知,16 阶判断矩阵的 RI 大于 1.59.因此,根据公式(25), $CR < 0.079 = 0.1257 / 1.59 < 0.1$,所以计算结果是可以接受的.最终,本文选择第 8 个候选解策略池作为 SaDDE 算法的最终策略池.第 8 个策略池所包含的 9 个 CSGS 分别为 CS123,CS122,CS38,CS94,CS97,CS95,CS39,CS98 和 CS96.这些策略的详细信息见表 5.

Table 5 Detail information of the 9 CSGSs that finally employed by the SaDDE in the strategy pool
表 5 最终 SaDDE 策略池包含的 9 种 CSGS 的详细信息

CSGS	Components			
	Mutated individual generating operators	Manners for obtaining mutant individual	Trial individual generating operators	Manners for obtaining mutant individual
CS123	MBswap	$V_i^g = m_i \oplus F_k \cdot (X_a^{g-1})$	PartMapXover	$U_i^g = c_1 \oplus CR(X_i^{g-1}, V_i^g)$
CS 122	MBswap	$V_i^g = m_i \oplus F_k \cdot (X_a^{g-1})$	OX	$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1})$
CS 38	ReverseMut	$V_i^g = m_i \oplus F_k \cdot (X_i^{g-1})$	OX	$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1})$
CS 94	InsertMut	$V_i^g = m_i \oplus F_k \cdot (X_a^{g-1})$	OrderBasedOX	$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1})$
CS 97	InsertMut	$V_i^g = m_i \oplus F_k \cdot (G^{g-1})$	OX	$U_i^g = c_1 \oplus CR(X_i^{g-1}, V_i^g)$
CS 95	InsertMut	$V_i^g = m_i \oplus F_k \cdot (X_a^{g-1})$	PosBasedOX	$U_i^g = c_1 \oplus CR(X_i^{g-1}, V_i^g)$
CS 39	ReverseMut	$V_i^g = m_i \oplus F_k \cdot (X_i^{g-1})$	PartMapXover	$U_i^g = c_1 \oplus CR(X_i^{g-1}, V_i^g)$
CS 98	InsertMut	$V_i^g = m_i \oplus F_k \cdot (G^{g-1})$	OX	$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1})$
CS 96	InsertMut	$V_i^g = m_i \oplus F_k \cdot (X_a^{g-1})$	PosBasedOX	$U_i^g = c_1 \oplus CR(V_i^g, X_i^{g-1})$

3.4 SaDDE与其他算法的性能对比实验与分析

为了评估 SaDDE 算法在求解大规模 CJWTA 问题上的性能,将 SaDDE 算法和相关文献最近提出的两种离散优化算法(即 HDDE^[17]和 DABC^[18])以及与求解问题相关的一种算法(即 GA-GE^[19]),在前述 6 个测试用例和更大规模的 3 个测试用例上做了算法的性能对比实验,新增加的测试用例分别为 T_INS_7(w=300,t=300); T_INS_8(w=400,t=400);T_INS_9(w=500,t=500).表 6 给出了每种算法在每个测试实例上求得的优化解的均值和标准差(计算结果为 30 次独立运行的统计数据).表 6 中,在每个测试实例上获得的最好的均值用黑体进行了标注.通过观察表 6 可以看出:在所有的测试实例上,SaDDE 算法性能优于其他算法,特别是在较大规模测试用例上,SaDDE 算法性能优越性更加明显.尽管 DABC 算法在 T_INS_1 上能够获得和 SaDDE 相同的均值,但 SaDDE 能够获得更好的标准差.

Table 6 Mean value and standard deviation of the solution fitness obtained by the compared algorithms on the 6 test instances

表 6 各种对比算法在 6 个测试实例上获得的最优解均值和标准差

INSes	Algorithms							
	SaDDE		HDDE		GA-GE		DABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
T_INS_1	1.13E-001	0.00E+000	1.16E-001	6.19E-004	1.16E-001	1.83E-003	1.13E-001	6.48E-004
T_INS_2	3.63E-002	3.71E-004	3.80E-002	5.25E-004	4.07E-002	3.86E-004	3.84E-002	4.32E-005
T_INS_3	1.31E-002	1.24E-004	1.58E-002	3.38E-004	1.68E-002	1.25E-004	1.57E-002	3.12E-004
T_INS_4	8.65E-003	4.64E-006	1.13E-002	3.00E-005	1.21E-002	2.59E-004	1.14E-002	1.64E-006
T_INS_5	5.93E-003	7.77E-006	8.22E-003	5.93E-005	8.58E-003	1.27E-004	8.21E-003	1.03E-004
T_INS_6	5.46E-003	2.81E-005	7.50E-003	6.38E-005	7.88E-003	1.72E-004	7.44E-003	1.24E-004
T_INS_7	4.24E-003	1.82E-005	5.63E-003	1.89E-005	5.79E-003	2.72E-005	5.63E-003	1.67E-005
T_INS_8	3.65E-003	8.10E-006	4.47E-003	1.99E-005	4.58E-003	1.31E-005	4.48E-003	1.83E-005
T_INS_9	3.19E-003	3.39E-007	3.70E-003	3.92E-006	3.75E-003	1.68E-005	3.70E-003	7.14E-006

SaDDE 及其对比算法在典型测试实例 T_INS_1,T_INS_6,T_INS_8 和 T_INS_9 上适应度随评估次数变化收敛曲线如图 2 所示.为简短起见,省略了在其他几个测试实例上各种算法的收敛曲线图.为了便于观察,图中横坐标取评估次数的自然对数.

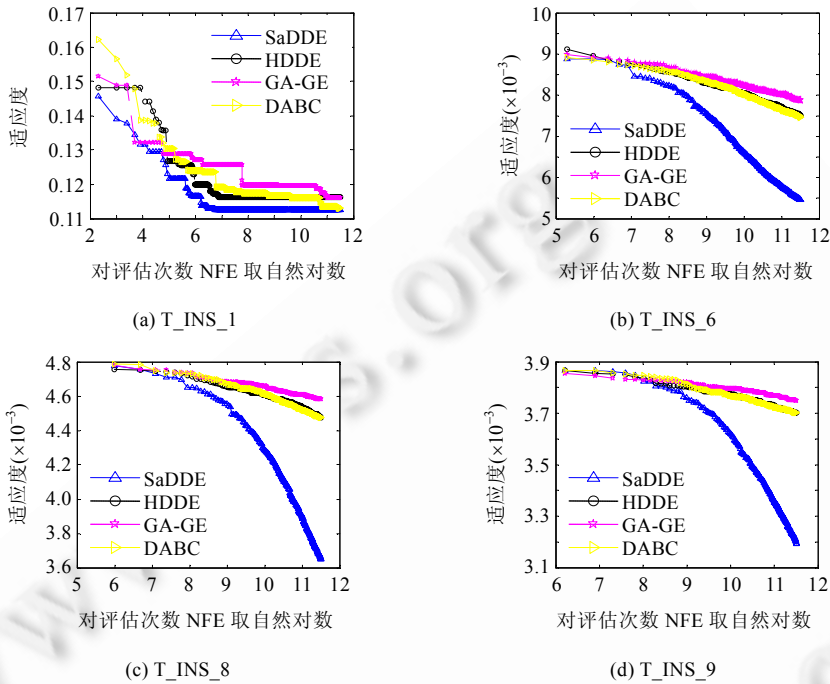


Fig.2 Convergence curves of the compared algorithms on test instances of T_INS_1, T_INS_6, T_INS_8, T_INS_9

图 2 对比算法在测试实例 T_INS_1,T_INS_6,T_INS_8 和 T_INS_9 上的收敛曲线图

通过观察图 2 可知:在小规模测试用例上,SaDDE 算法的收敛速度和其他算法相差无几,求解质量稍有优势;但在大规模测试用例和较大规模测试用例上,SaDDE 算法性能明显优于其他几种算法.由图 2 各子图的对比可见:随着问题规模的增大,SaDDE 的优越性越来越明显.通过对图 2 分析可以得出以下结论:

- (1) 总体而言,SaDDE 的求解速度和求解质量优于其他算法;
- (2) SaDDE 算法的求解有效性和效率随着求解规模的增加而越来越明显;
- (3) 一般智能优化算法随着问题规模的增大性能迅速下降,但 SaDDE 仍能保持较好的求解特性.这一现象表明,多策略并行搜索技术是克服单一智能优化算法随求解实例规模增大而自身性能下降问题的有效途径之一.

4 结束语

在进化算法研究领域存在着多种多样的候选解产生策略,但是如何根据实际问题为算法确定最优策略通常是很困难的,目前还没有行之有效的策略选择理论指导原则.因此,为了克服策略选择的困难,研究人员提出了一些多策略自适应进化算法,这些算法通常采用某种策略自适应产生机制和参数更新方法实现多策略的自适应选择,以增强算法的普适性和鲁棒性.通过对这些算法的分析可知,候选解产生策略池是这类算法的重要结构之一,然而策略候选池的设计存在着 3 个主要问题.为了解决这 3 个主要问题,提出了基于相对排列顺序的标度法(RPOSM)和基于 RPOSM 的层次分析法.实验部分首先做了策略选择实验和策略池大小寻找实验,通过实验确定了 SaDDE 算法结构和参数.为了评估所设计算法的性能,做了算法对比实验,实验结果表明:SaDDE 算

法在求解 CJWTA 问题上十分高效,在有限的评估次数内,在求解质量和求解速度上,SaDDE 可以获得比 HDDE, DABC 和 GA-GE 更好的求解结果;而且随着所求问题规模的增长,SaDDE 的优势越来越突出.本文重点在于解决策略池设计中的 3 个主要问题.其中:第 1 个问题基本得到了解决;对于第 2 个和第 3 个问题,本文提供了解决办法,但只是解决了有无的问题,这两个问题还没有得到彻底的解决,进一步的研究工作将以此为重点.另一方面,在算法中没有考虑启发式信息、基于威胁度的整数编码方案和个体修复过程等,这也是进一步的研究内容.

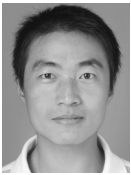
References:

- [1] Xue Y, Zhuang Y, Zhang YY, Ni SR, Zhao XJ. Multiple UCAV cooperative jamming air combat decision making based on heuristic self-adaptive discrete differential evolution algorithm. *Acta Aeronautica et Astronautica Sinica*, 2012,34(2):343–351 (in Chinese with English abstract). [doi: 10.7527/S1000-6893.2013.0039]
- [2] Bi XJ, Xiao J. Classification-Based self-adaptive differential evolution with fast and reliable convergence performance. *Soft Computing*, 2011,15(8):1581–1599. [doi: 10.1007/s00500-010-0689-5]
- [3] Gong WY, Cai ZH, Ling CX, Li H. Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Trans. on Systems Man and Cybernetics Part B—Cybernetics*, 2011,41(2):397–413. [doi: 10.1109/tsmcb.2010.2056367]
- [4] Xu B, Zhuang Y, Xue Y, Wang Z. Self-Adaptive learning based immune algorithm. *Journal of Central South University of Technology*, 2012,19(4):1021–1031. [doi: 10.1007/s11771-012-1105-3]
- [5] Wang Y, Li B, Weise T, Wang JY, Yuan B, Tian QJ. Self-Adaptive learning based particle swarm optimization. *Information Sciences*, 2011,181(20):4515–4538. [doi: 10.1016/j.ins.2010.07.013]
- [6] Wang Y, Zhou JZ, Zhou C, Wang YQ, Qin H, Lu YL. An improved self-adaptive PSO technique for short-term hydrothermal scheduling. *Expert Systems with Applications*, 2012,39(3):2288–2295. [doi: 10.1016/j.eswa.2011.08.007]
- [7] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. on Evolutionary Computation*, 2009,13(2):398–417. [doi: 10.1109/TEVC.2008.927706]
- [8] Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, 1997,1(1):67–82. [doi: 10.1109/4235.585893]
- [9] Saaty TL. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 1990,48(1):9–26. [doi: 10.1016/0377-2217(90)90057-1]
- [10] Saaty TL. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 1977,15(3):234–281. [doi: 10.1016/0022-2496(77)90033-5]
- [11] Liu CY, Wang DS. AHPCC: A high performance computer system evaluation model based on HPCC and analytic hierarchy process. *Ruan Jian Xue Bao/Journal of Software*, 2007,18(4):1039–1046 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1039.htm> [doi: 10.1360/jos181039]
- [12] Pan QK, Tasgetiren MF, Liang YC. A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering*, 2008,55(4):795–816. [doi: 10.1016/j.cie.2008.03.003]
- [13] Tasgetiren MF, Pan QK, Suganthan PN, Liang YC. A discrete differential evolution algorithm for the no-wait flowshop scheduling problem with total flowtime criterion. In: *Proc. of the IEEE Symp. on Computational Intelligence in Scheduling*. Piscataway: IEEE, 2007. 251–258. [doi: 10.1109/scis.2007.367698]
- [14] Tasgetiren MF, Pan QK, Liang YC, Suganthan PN. A discrete differential evolution algorithm for the total earliness and tardiness penalties with a common due date on a single-machine. In: *Proc. of the IEEE Symp. on Computational Intelligence in Scheduling*. Piscataway: IEEE, 2007. 271–278. [doi: 10.1109/scis.2007.367701]
- [15] Davis L. Applying adaptive algorithms to epistatic domains. In: *Proc. of the Int'l Joint Conf. on Artificial Intelligence*. Scottish: British Computer Society, 1985. 161–163.
- [16] Singh V, Choudhary S. Genetic algorithm for traveling salesman problem: Using modified partially-mapped crossover operator. In: *Proc. of the Int'l Conf. on Multimedia, Signal Processing and Communication Technologies*. Piscataway: IEEE, 2009. 20–23. [doi: 10.1109/mspct.2009.5164164]
- [17] Pan QK, Wang L, Gao L, Li WD. An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. *Information Sciences*, 2011,181(3):668–685. [doi: 10.1016/j.ins.2010.10.009]

- [18] Pan QK, Tasgetiren MF, Suganthan PN, Chua TJ. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 2011,181(12):2455–2468. [doi: 10.1016/j.ins.2009.12.025]
- [19] Lee ZJ, Su SF, Lee CY. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Trans. on Systems Man and Cybernetics Part B—Cybernetics*, 2003,33(1):113–121. [doi: 10.1109/tsmcb.2003.808174]

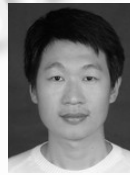
附中文参考文献:

- [1] 薛羽,庄毅,张友益,倪思如,赵学健.基于启发式自适应离散差分进化算法的多UCAV协同干扰空战决策.航空学报,2012,34(2):343–351. [doi: 10.7527/S1000-6893.2013.0039]
- [11] 刘川意,汪东升.基于HPCC和层次分析法的高性能计算系统评价模型.软件学报,2007,18(4):1039–1046. <http://www.jos.org.cn/1000-9825/18/1039.htm> [doi: 10.1360/jos181039]



薛羽(1981—),男,山东济宁人,博士,讲师,CCF会员,主要研究领域为电子对抗,智能计算,物联网.

E-mail: xueyu_123@nuaa.edu.cn



常相茂(1982—),男,博士,讲师,主要研究领域为无线传感器网络.

E-mail: xiangmaoch@nuaa.edu.cn



庄毅(1956—),女,教授,博士生导师,主要研究领域为分布计算,可信计算,网络安全.

E-mail: zhuangyi@nuaa.edu.cn



王洲(1975—),男,高级工程师,主要研究领域为系统总体设计,系统仿真实验.

E-mail: hi.wz@tom.com



顾晶晶(1983—),女,博士,讲师,主要研究领域为物联网,智能计算,无线传感器网络定位.

E-mail: gujingjing@nuaa.edu.cn