

满足故障隔离率指标的测试序列优化差分进化算法

邱晓红 李 渤 李 靖

(江西理工大学软件学院, 南昌, 330013)

摘要: 最优测试序列的设计是故障诊断过程中必须解决的非确定多项式(Non-deterministic polynomial, NP)完全问题。基于自适应差分进化算法, 提出一种惯性速度差分进化(Inertial velocity differential evolution, IVDE)算法, 通过增加额外的惯性速度项求解复杂电子系统最优测试序列问题(Optimal test sequence problem, OTP)。为求解该优化问题设计了个体的状态与测试序列编码方式, 构建了包含故障隔离率(Fault isolation rate, FIR)等指标的个体适应度函数, 通过优化生成诊断决策树来减少测试设备和测试成本。仿真结果表明, IVDE 算法可以求得既满足 FIR 要求, 又减少测试成本的测试序列。与粒子群优化算法(Particle swarm optimizer, PSO)、遗传算法(Genetic algorithm, GA)等其他算法相比, IVDE 可以求解 OTP, 得到更好的解。

关键词: 差分进化; 进化计算; 故障隔离率; 测试性

中图分类号: TP391 **文献标志码:** A

Sequential Fault Diagnosis with Isolation Rate Requirement Using Differential Evolution Algorithm

Qiu Xiaohong, Li Bo, Li Jing

(Software School, Jiangxi University of Science and Technology, Nanchang, 330013, China)

Abstract: The optimal test sequence design for fault diagnosis is critical to NP-complete problem. An improved differential evolution (DE) algorithm with additional inertial velocity item is proposed to solve the optimal test sequence problem (OTP) in complicated electronic system. The proposed inertial velocity differential evolution (IVDE) algorithm is constructed based on adaptive differential evolution algorithm. IVDE, combined with a new individual fitness function, optimizes the test sequence sets with the index fault isolation rate(FIR) satisfied in top-down to generate diagnostic decision tree and decrease the test sets and the cost test. Simulation results show that IVDE algorithm can cut down the test cost with the satisfied FIR. Compared with the other algorithm such as particle swarm optimizer(PSO) and genetic algorithm(GA), IVDE can get better solution to OTP.

Key words: differential evolution; evolutionary computation; fault isolation rate; testability

引 言

复杂系统如航天系统或商用飞机对系统的安全性和关键任务的可靠性要求高^[1], 需要高额维护

费用。高昂的维护成本往往在设计阶段未充分考虑可测性需求。为提高可测性设计,测试序列的设计优化是故障诊断关注的重点,但它是一个具有挑战性的非确定多项式问题(Non-deterministic polynomial, NP)完全问题^[2]。现在主要采用动态规划和启发式搜索算法来求解该问题。动态规划自底向上构建测试树,需要指数增长的存储空间,计算复杂性与 $O(3^n)$ (n 为测试点的数目)相当^[2,3]。基于熵的启发式 AO* 搜索方法,提出了与/或图搜索来解决该问题^[4]。AO* 算法在扩展每个节点时,通过比较熵的大小选择当前的最佳测试点,但这容易陷入局部最优解。对中等规模的最优测试序列(Optimal test sequence problem, OTP)研究了一些实用的算法,如传统的拉格朗日松弛法、次优梯度^[5]和自下而上的测试序列生成方法^[6]。但对复杂系统,更多的学者采用智能算法如遗传算法^[7]、离散二值粒子群算法(Discrete binary particle swarm optimizer, DBPSO)^[8]和量子态粒子群算法^[9]等求解最优诊断测试策略,减小测试成本。通过增加随机冒险项,改进后的 DBPSO 能求得更好的测试序列^[10]。不少学者也探索了更多智能算法的应用,如用萤火虫算法解决软件测试问题^[11],设立蚂蚁状态转移规则和反馈的蚁群算法以克服测试序列问题的计算爆炸^[12]。但这些算法都构造了一个与测试成本不直接相关的适应度函数,也没有直接体现故障检测率和故障隔离率等可测试性指标要求^[13]。差分进化(Differential evolution, DE)算法^[14]具有其独到的优势特性,如具有外部存档功能的自适应差分进化(J adaptive differential evolution, JADE)^[14]和复合试验向量生成策略差分进化算法(Differential evolution with composite trial vector generation, CoDE)^[15],对测试的优化函数都取得了较好结果,但目前还没有将其应用到求解 OTP。本文提出一种惯性速度差分进化算法(Inertial velocity differential evolution, IVDE),同时构建差分算法个体状态与测试序列对应的编码方式用于求解 OTP,获得了更好的测试方案。本文首先介绍了惯性速度差分进化算法,分析了优化测试序列数学模型,提出了一个新的测试序列优化适应度函数,综合了测试成本与故障检测率(Fault detection rate, FDR)、故障隔离率(Fault isolation rate, FIR)等指标要求,与其他算法进行了实验比较分析,最后分析了 IVDE 求解 OTP 存在的不足。

1 改进的惯性速度差分进化算法

假设初始种群 $P_0 = \{x_1, \dots, x_r, \dots, x_{N_p}\}$, 其中 N_p 为初始种群个体数, 个体 x_r 是状态为 N 维的向量, $x_r = [x_{r,1}, \dots, x_{r,d}, \dots, x_{r,N}] \in \mathbf{R}^N$, 最小化目标函数 $f(x_r)$, 其解空间为 $\Omega = \prod_{r=1}^N [x_{r,\min}, x_{r,\max}]$ 。DE 算法可用于求解此类优化问题^[14]。假设在 k 代为个体 x_r 创建一个变异向量 $v_r = [v_{r,1}, \dots, v_{r,d}, \dots, v_{r,N}] \in \mathbf{R}^N$, 变异操作直接影响整个算法的性能。一种具有优良性能的称为“DE/current-to-pbest/1”变异操作^[14] 表达为

$$v_{r,d}^{k+1} = x_{r,d}^k + F_r \times ((x_{pbest,d}^k - x_{r,d}^k) + \lambda \times (x_{r_1,d}^k - x_{r_2,d}^k)) \tag{1}$$

式中: $v_{r,d}^{k+1}$ 为新产生的个体变异状态分量, 上标 k 为第 k 次迭代或第 k 代, 下标 r 为第 r 个个体(粒子), 下标 d 为第 d 维分量。其中 r_1 和 r_2 为从范围 $[1, N_p]$ 随机选出的不同于 r 的整数, 参数 F_r 称为变异因子。 $x_{pbest,d}^k$ 为随机从适应度最优的前 100 $p\%$ ($p \in (0, 1]$) 个体中选择出的优秀个体的 d 维分量。经过变异操作后, 对 $x_{r,d}$ 和 $v_{r,d}$ 两项执行交叉操作, 产生测试向量 $u_{r,d}$

$$u_{r,d} = \begin{cases} v_{r,d} & \text{rand}(1) \leq C_R \quad \text{或} \quad d = \text{rand}(D) \\ x_{r,d}^k & \text{其他} \end{cases} \tag{2}$$

式中: $r = 1, 2, \dots, N_p; d = 1, 2, \dots, N; \text{rand}(N)$ 为从 $1 \sim N$ 随机选择出的一个整数, $\text{rand}(1)$ 为在区间 $[0, 1]$ 均匀分布的随机数, $C_R \in [0, 1]$ 为交叉操作控制参数。因为使用了随机函数 $\text{rand}(\cdot)$, 测试向量 u_r 一般不同于目标向量 x_r 。如果测试分量值 $u_{r,d}$ 出界, 则采用式(3)进行修正, 则有

$$u_{r,d} = \begin{cases} \min\{x_{d,\max}, 2x_{d,\min} - u_{r,d}\} & \text{或} \quad u_{r,d} < x_{d,\min} \\ \max\{x_{d,\min}, 2x_{d,\max} - u_{r,d}\} & \text{或} \quad u_{r,d} > x_{d,\max} \end{cases} \tag{3}$$

然后进入选择操作, 即选择具有更好性能的个体 u_r^k 进入下一代 x_r^{k+1} , 如

$$\mathbf{x}_r^{k+1} = \begin{cases} \mathbf{u}_r^k & f(\mathbf{u}_r^k) < f(\mathbf{x}_r^k) \\ \mathbf{x}_r^k & \text{其他} \end{cases} \quad (4)$$

为记录这一成功更新的参数,定义一更新状态标志向量 $\mathbf{F}_u = (F_{u,1}, \dots, F_{u,r}, \dots, F_{u,N})$,其中的元素为

$$F_{u,r}^{k+1} = \begin{cases} 1 & f(\mathbf{u}_r^k) < f(\mathbf{x}_r^k) \\ 0 & \text{其他} \end{cases} \quad (5)$$

为了避免陷入局部最优解,学习 PSO 的优点^[16],增加速度的惯性加权,在差分算法中引入速度项,式(1)写成式(6,7),即

$$\mathbf{vel}_{r,d}^{k+1} = \omega_{r,d}^k \times \mathbf{vel}_{r,d}^k + F_r^k \times ((x_{\text{pbest},d}^k - x_{r,d}^k) + \lambda \times (x_{r1,d}^k - x_{r2,d}^k)) \quad (6)$$

$$\mathbf{v}_{r,d}^{k+1} = x_{r,d}^k + \mathbf{vel}_{r,d}^k \quad (7)$$

式中: $\mathbf{vel}_{r,d}^k$ 为第 k 代第 r 个个体第 d 维分量的速度; $\omega_{r,d}^k$ 为速度的惯性因子。变异因子 F_r^k 与 x_r 关联,服从 Cauchy 分布^[14],即 $F_r = \text{randc}(\mu_F, 0, 1)$, $\text{randc}(\mu_F, 0, 1)$ 是位置参数为 μ_F ,尺度参数为 0.1 的 Cauchy 分布函数。若 $F_r \geq 1$,设置为 1,若 $F_r \leq 0$,则重新产生。位置参数 μ_F 初始值为 0.5,并且更新 $\mu_F = (1 - c) \times \mu_F + c \times \text{mean}_L(S_F)$, S_F 为获得成功变异个体的集合, $c \in (0, 1)$ 为正常数,函数 $\text{mean}_L(\cdot)$ 为如式(8)计算的 Lehmer 均值函数,则有

$$\text{mean}_L(S_F) = \frac{\sum_{F_r \in S_F} F_r^2}{\sum_{F_r \in S_F} F_r} \quad (8)$$

类似地,交叉概率 $C_{R,r}$ 也采用类似的方法更新,计算方法为

$$C_{R,r} = \text{randn}(\mu_{CR}, 0, 1) + C_{R,rw} \quad (9)$$

$$\mu_{CR} = (1 - c) \times \mu_{CR} + c \times \text{mean}_A(S_{CR}) \quad (10)$$

式中: $\text{randn}(\mu_{CR}, 0, 1)$ 是均值为 μ_{CR} 、方差为 0.1 的正态随机函数, $C_{R,rw}$ 为随适应度函数值变化的加权值。 S_{CR} 为交叉操作成功的个体集合,均值 μ_{CR} 初始值取为 0.5,更新计算公式如式(10),式中 $c \in (0, 1)$ 是一小于 1 的正常数, $\text{mean}_A(\cdot)$ 为算术平均值函数。令 $f_{\min} = \min_{x_r \in P} \{f(x_r)\}$,并根据个体目标函数值进行升序排序 $f_{r,\text{sort}} = \text{sort}_{x_r \in P}^{\text{Ascending}} \{f(x_r)\}$,则算法的其他参数更新为

$$C_v(r) = \sin\left(\frac{f_{r,\text{sort}}}{N_p} \pi\right) \quad (11)$$

$$C_{R,rw}^k = \frac{1 + \alpha_1 k}{1 + \alpha_2 k} \times 0.041 \times C_v(r) \quad (12)$$

$$\mathbf{F}_v^{k+1} = 0.382 \times \mathbf{F}_v^k + 0.618 \times \mathbf{F}_u^k \quad (13)$$

$$\mathbf{w}_r^k = \frac{1 + \alpha_1 k}{1 + \alpha_2 k} \times (0.1 \text{rand}(0, 1) + 0.618) \times \mathbf{F}_v^k \quad (14)$$

$$p^k = \frac{1 + \alpha_1 k}{1 + \alpha_2 k} \times p^0 \quad (15)$$

式中: k 为迭代次数或进化代数, \mathbf{F}_v 用于存储状态进化标志向量 \mathbf{F}_u ,其初始值设为 0。 \mathbf{w}_r 为式(6)中速度向量的惯性因子向量。 p^0 为“100p%”的初始概率值^[14]。 α_1, α_2 为随进化代数增加进行调节的滤波器控制参数。IVDE 具有惯性加权的 PSO 算法的优点,其惯性加权因子 $\omega_{n,d}^k$ 也随迭代次数增加衰减。所以, $\frac{1 + \alpha_1 k}{1 + \alpha_2 k}$ 在 k 很大时,其值近似为 1/6。

2 测试序列优化数学模型

最优测试序列问题就是在满足 FDR 和 FIR 条件下,以最小的测试代价,设计出一个能够有效隔离故障状态的测试序列策略。OTP 的参数通常定义为五元组 $(S, \mathbf{P}, T, C, \mathbf{D})$ ^[2],其中 $S = \{s_0, s_1, \dots, s_m\}$ 为与系统相关的统计独立的故障状态集合。 s_0 代表“无故障”状态, $s_i (0 < i < m + 1)$ 代表不同的故障状态。

$P=[p(s_0), p(s_1), \dots, p(s_m)]$ 为故障状态集 S 对应的先验概率。 $T=\{t_1, t_2, \dots, t_n\}$ 为 n 个用于检测故障状态的可靠测试, 每个测试 t_j 能检查到相应的故障状态子集。 $C=\{c_1, c_2, \dots, c_n\}$ 为测试代价集, c_j 对应测试 t_j 所耗的时间、人力和物力等。 $D=[d_{ij}]$ 为 $m \times n$ 维二值关联矩阵, 其中 $d_{ij}=1$, 表示 t_j 能检测到故障 s_i , 否则 $d_{ij}=0$ 。一个简单的 OTP 参数如表 1 所示。

表 1 小型 OTP 参数例表

Tab. 1 OTP parameters for small scale example

故障状态	t_1	t_2	t_3	t_4	t_5	p
s_0	0	0	0	0	0	0.01
s_1	0	1	0	0	1	0.08
s_2	0	0	1	1	0	0.28
s_3	1	0	0	1	1	0.20
s_4	1	1	0	0	0	0.30
s_5	1	1	1	1	0	0.13
C	1.0	4.0	3.0	3.0	5.0	

本文假定系统同时仅发生一个故障。OTP 问题就是在给定的 (S, P, T, C, D) 与所需的故障隔离率 (如 $FIR \geq FIR_{target}$) 等条件下, 设计一个测试序列, 应用 T 中的测试准确识别出 S 中状态的发生, 并且最大限度地减少如式 (16) 描述的测试成本, 即

$$J = P^T A C = \sum_{i=0}^m \sum_{j=1}^n a_{ij} p(s_i) c_j \tag{16}$$

式中: $A=[a_{ij}]$ 为 $(m+1) \times n$ 维二值关联矩阵。 $a_{ij}=1$ 表示用了测试 t_j 来辨别故障状态 s_i , 否则 $a_{ij}=0$ 。 FIR_{target} 为系统要求的最低故障检测率。

3 IVDE 求解 OTP 问题

3.1 测试序列编码

当使用 IVDE 来解决最优测试排序问题, 首先必须解决个体状态向量与测试序列的编码问题。定义个体状态向量 $x=[x_1, \dots, x_i, \dots, x_N]$ 为测试序列, 其维数等于测试数目, 对应测试集 T 。 x_i 的值对应于第 j 个试验。对每一个体向量的分量值降序排序, 可以作为一个诊断序列。例如, 一个 5 维向量 $x=[8.1, -6.5, -0.9, 6.2, -7.3]$, 如图 1 所示递减顺序, 它的诊断序列的结果为 $T_s=\{1, 4, 3, 2, 5\}$ 。在测试空间, 这些值表示测试集 T 的测试序列, 所以向量 x 将给出诊断策略 $\{t_1 \rightarrow t_4 \rightarrow t_3 \rightarrow t_2 \rightarrow t_5\}$ 。该测试序列可视为如表 1 所示范例的 OTP 解。其测试序列构建的测试树如图 2 所示。

x	x_1	x_2	...	x_i	...	x_N
测试序列	j

测试集	t_1	t_2	t_3	t_4	t_5
x	8.1	-6.5	-0.9	6.2	-7.3
测试序列	1	4	3	2	5

图 1 个体状态对应测试序列的编码

Fig. 1 Individual state for test sequence coder

图 2 中, 如果假设 $FIR_{target}=0.9$, 所有的故障状态集 $S=\{s_0, s_1, s_2, s_3, s_4, s_5\}$ 。测试序列 $\{t_1 \rightarrow t_4 \rightarrow t_3 \rightarrow t_2 \rightarrow t_5\}$ 的第 1 个测试 t_1 , 根据关联矩阵 D 中第 t_1 列值 $\{0, 0, 0, 1, 1, 1\}$ 将 S 分成两模糊集: 使得 $t_1=1$ 的

$\{s_0, s_1, s_2\}$ 和 $t_1=0$ 的 $\{s_3, s_4, s_5\}$ 。类似地,使用第 2 个测试 t_4 测试这两个模糊集。 $\{s_0, s_1, s_2\}$ 分成 $\{s_0, s_1\}$ 和 $\{s_2\}$ 。 $\{s_3, s_4, s_5\}$ 被分离为 $\{s_4\}$ 和 $\{s_3, s_5\}$ 。共有 4 个故障状态子集合:两个模糊集和两个已隔离出的故障状态。所以计算此时的 $FIR = \text{隔离出单个故障状态概率之和} = p(s_2) + p(s_4) = 0.58$ 。如果 $FIR \geq FIR_{\text{target}}$, 则说明测试序列已经足够, 停止继续使用下一个测试。否则, 使用下一个测试, 继续测试分离所有故障子集。所以继续使用第 3 个测试 t_3 , 但 t_3 不能将模糊集 $\{s_0, s_1\}$ 分成两个子集, t_3 仅将 $\{s_3, s_5\}$ 成功地分离到故障状态 $\{s_3\}$ 和 $\{s_5\}$ 。现计算 $FIR = 0.58 + p(s_3) + p(s_5) = 0.91$, 该值大于 $FIR_{\text{target}} = 0.9$, 满足了设计要求。因此针对该范例有效的测试序列长度 $N_t = 3$, 代入式(16), 可计算出测试代价 $J = 4.99$ 。

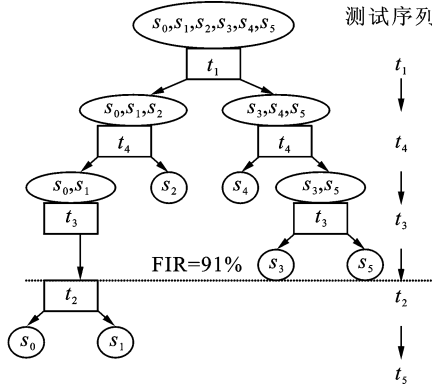


图 2 测试序列隔离故障流程

Fig. 2 Fault isolation flow diagram by test sequence

3.2 IVDE 求解 OTP 问题的适应度函数

假设个体向量 x 对应一诊断序列, 根据上述方法可以建立诊断树, 确定它的故障隔离矩阵 $A = [a_{ij}]$, 计算出 FDR 和 FIR, 并通过式(16) 计算出其诊断成本, 由此可以建立适应度函数。将 FDR, FIR 和测试成本等纳入适应度函数 $f(x)$ 中, 则有

$$f(x_r) = \sum_{i=0}^m \sum_{j=1}^n a_{ij} p(s_i) c_j + \alpha(FDR_{\text{target}} - FDR) + \beta(FIR_{\text{target}} - FIR) + \gamma N_t \quad (17)$$

式中: N_t 为测试序列集中用于检测和隔离故障状态所使用的测试个数; α, β, γ 为 3 个加权系数; α, β 越大, 故障检测率和故障隔离率越接近于指定的 $FDR_{\text{target}}, FIR_{\text{target}}$ 值。其他参数的意义如式(16)。本文中主要采用下列参数: $FDR \geq FDR_{\text{target}}, \alpha = 1$, 否则 $\alpha = -100$; $FIR \geq FIR_{\text{target}}, \beta = 1$, 否则 $\beta = -100$; $\gamma = 0.0001, FDR_{\text{target}} = 100\%$ 。值得注意的是式(17) 给出的适应度函数值是为了求最小值, 即适应度值越小, 其个体生存度越大。

4 实验结果与分析

结合测试序列编码和适应度函数, IVDE 求解 OTP 问题的算法在 Matlab 环境下编程加以验证。仿真实验程序在 Intel 双核 2.93 GHz CPU、Windows7 操作系统的 PC 上运行。IVDE 算法基本参数设定

为: 状态维数 $N = \text{测试数目}$, $\Omega = \prod_{r=1}^N [x_{i, \min}, x_{i, \max}]$, $x_{r, \min} = -10, x_{r, \max} = 10$ 。在不特别说明时, 运算迭代次数的最大数目是 1 000, 种群大小 $N_p = 100$, 即每次求解问题时所允许的适应度函数计算最大次数 $FE_{S_{\max}} = 10^5, c = 0.1, p^0 = 0.35, \alpha_1 = 0.00011, \alpha_2 = 0.00067$ 。 μ_{CR} 和 μ_F 的初始值都设为 0.5。选择 3 个范例来验证 IVDE 求解 OTP。应用任一算法求解问题时, 都独立运行 25 次。

4.1 APOLLO 发射前检测

阿波罗发射前检测的例子总是用来测试诊断算法^[2]。在该系统中, 有 10 个故障状态和 15 个测试

所要求的 $FIR_{target} = 100\%$ 。该系统总假定在失败状态,即无故障条件 $p(s_0)$ 的概率。系统的故障状态、测试的关联矩阵 D 、故障发生概率 P 及测试成本等参数见文献[2]。独立运行 25 次,用 IVDE 求得的最佳解决方案如表 2 所示,表中 T_{S_1} 和 T_{S_8} 有相同的测试成本和适应度。将表 3 所示结果与启发式算法(Heuristic evaluation function, HEF) [2] 结果相比,两者相同。IVDE 运行 25 次,几乎得到 25 种不同的最佳测试序列,但具有几乎相同的测试成本。

表 2 APOLLO 发射前检测优化序列

Tab. 2 Optimal solution on APOLLO prelaunch checkout

优化测试序列集合 T_s	成本	适应度
$T_{S_1} = \{t_{15}, t_7, t_8, t_9\}$		
$T_{S_2} = \{t_{11}, t_3, t_{13}, t_{12}\}$		
$T_{S_3} = \{t_5, t_8, t_6, t_1\}$		
$T_{S_4} = \{t_{15}, t_{12}, t_{11}, t_{10}\}$	3.400 0 ± 1.81E-15	3.400 4 ± 1.81E-15
$T_{S_5} = \{t_8, t_9, t_4, t_{15}\}$		
$T_{S_6} = \{t_9, t_{11}, t_3, t_2\}$		
$T_{S_7} = \{t_{12}, t_{13}, t_9, t_{15}\}$		
$T_{S_8} = \{t_5, t_8, t_{10}, t_{11}\} \dots\dots$		

4.2 反坦克系统

反坦克系统主要用于击毁重装甲军车导弹系统。该系统有 13 个故障状态和 12 项测试。故障状态概率和测试成本等参数见文献[13]。IVDE 独立运行 25 次,求得的最佳解决方案如表 3 所示。

表 3 反坦克系统^[13]测试序列优化结果比较

Tab. 3 Comparisons on anti-tank system^[13] OTP

FIR 目标		≥95%	≥90%	≥85%	≥80%	≥75%	≥70%
FIR		100%	94%	89%	83%	79%	73%
SADPSO	测试点数目	11	10	10	9	9	8
	成本	4.86	4.63	4.56	4.32	4.14	3.9
FIR		100%	94%	89%	83%	79%	73%
VDE	测试点数目	11	10	10	9	9	8
	成本	4.75	4.52	4.44	4.21	4.02	3.9

如表 3 所示,IVDE 方法的性能与文献[13]的离散二值粒子群算法(Discrete binary particle swarm optimizer, DBPSO)、自适应测试优化算法(Self-adaptive test optimizing DPSO, SADPSO)进行比较,在不同的故障隔离率指标要求下,IVDE 求得测试点数目和故障隔离率与 SADPSO 一样,但测试成本更小,IVDE 比 SADPSO 更有效。在文献[13]中, SADPSO 是讨论的最好算法,所以 IVDE 用于求解 OTP 问题更具有优势。

4.3 超外差接收机

雷达系统的超外差接收机^[2]比较复杂,求解其测试序列总是用于评估新的算法性能。该系统有 36 个不同的测试和 22 个故障状态。其故障状态和测试关系矩阵 D 、故障先验概率 P 和测试成本见文献[2],用该例子进一步验证 IVDE 算法。IVDE 算法使用相同的参数,除了 $N=36$ 外,其最优结果如表 4 所示。与其他算法^[7-10]相比,求得的结果如表 5 所示。

在表 4 中,所提出的 IVDE 方法的性能与 SADPSO^[13]相比较。在相同的 FIR 约束下,如表 4 中的第

1 行所示 90%, 80%, 70% 和 60%, 算法产生的测试序列应具有 FIR 应大于该值。表 4 的其他行显示了实际所需要的测试数目、测试成本和实际的 FIR。例如, IVDE 产生了测试成本为 3.25 和 FIR = 93.53% (大于 90%), 并且只需要 6 个不同的测试, 但 SADPSO 需要 9 个不同的测试才能达到近似的目标。IVDE 算法求得的只要 6 个不同的测试就能获得 FIR > 90% 测试策略如图 3 所示。为了实现 100% FIR, IVDE 需要 15 个测试隔离所有的故障状态, 其优化计算结果如表 5 所示。表 5 将不同的进化方法如遗传算法 (Genetic algorithm, GA)^[7], DPSO-AO^{*}^[8], 量子粒子群算法 (Quantum-behaved particle swarm optimization, QPSO)^[9], DBPSO-AO^{*}^[10] 求得的结果进行比较。表 5 的 T_{S_1} , T_{S_2} 和 T_{S_3} 有相同的测试成本和适应度。IVDE 得到了测试成本为 3.347 的最优解, 仅比 DPSO-AO^{*} 算法^[8] 求得的 3.02 要大, 比其他方法的测试成本都要小。虽然 IVDE 没有得到低于测试成本 3.02^[8] 的解决方案, 但认为该文的结果不尽合理, 因为其选择测试集包含 t_8 两次。所以在这些方法中, IVDE 应该是较好的解决方案。

表 4 超外差接收机^[13]测试序列优化结果比较
Tab. 4 Comparisons on super heterodyne receiver^[13]

算法	FIR 目标	≥90%	≥80%	≥70%	≥60%
	FIR	94.65%	80.26%	74.69%	60.26%
SADPSO	测试点数目	9	13	10	10
	成本	3.26	3.13	3.04	2.68
	FIR	93.53%	80.02%	74.85%	60.04%
IVDE	测试点数目	6	13	4	9
	成本	3.25	3.16	2.84	2.68

表 5 超外差接收机测试序列优化结果比较
Tab. 5 Optimal solution on superheterodyne receiver

算法	优化测试序列集合 T_s	成本	适应度
IVDE	$T_{S_1} = \{t_{34}, t_8, t_{19}, t_{30}, t_{28}, t_{26}, t_{29}, t_{32}, t_{31}, t_7, t_5, t_{21}, t_{22}, t_{10}, t_{14}\}$	3.347 4	3.348 9
	$T_{S_2} = \{t_{34}, t_8, t_{19}, t_{30}, t_{28}, t_{29}, t_{32}, t_{31}, t_{26}, t_{11}, t_{21}, t_6, t_{23}, t_{15}, t_{22}\}$		
	$T_{S_3} = \{t_{34}, t_8, t_{19}, t_{30}, t_{28}, t_{26}, t_{29}, t_{32}, t_{31}, t_{11}, t_{21}, t_6, t_{22}, t_7, t_{35}\}$		
DPSO-AO [*] ^[8]	$t_{30}, t_{26}, t_{31}, t_{34}, t_8, t_{11}, t_4, t_{21}, t_{22}, t_{12}, t_{28}, t_8, t_{19}$	3.02	
GA ^[7]	$t_{34}, t_{28}, t_8, t_{26}, t_{31}, t_{32}, t_{19}, t_{21}, t_{27}, t_5, t_{14}, t_{22}, t_{12}, t_{13}$	3.952 6	
DBPSO-AO [*] ^[10]	$t_{34}, t_7, t_8, t_{26}, t_{32}, t_{14}, t_{21}, t_{19}, t_{31}, t_{28}, t_{13}, t_{35}, t_{29}, t_{22}, t_{36}$	3.52	
QPSO ^[9]	$t_{32}, t_{34}, t_{28}, t_{31}, t_8, t_{26}, t_{13}, t_{14}, t_5, t_{12}, t_{21}, t_{27}, t_{19}, t_{22}$	3.390 4	

为了证明 IVDE 的优势, 将 IVDE 求解超外差接收机 OTP 的结果与其他进化算法, 如综合学习粒子群算法 (Comprehensive learning particle swarm optimizer, CLPSO)^[17], JADE^[14] 和 CoDE^[15] 进行比较。CLPSO, JADE 和 CoDE 等算法的 Matlab 代码和它们的参数参见文献[15]。其适应度函数如式(17), 运行 25 次的平均结果和标准偏差如表 6 所示, 收敛曲线如图 4 所示。从表中可以看出, 所有算法求得结果都已接近最优解, 差别不大, 但 IVDE 求得的结果最好, 平均收敛速度也最好, 具有收敛性和全局最优解的折中, 这意味着 IVDE 具有比其他算法更好的功能。

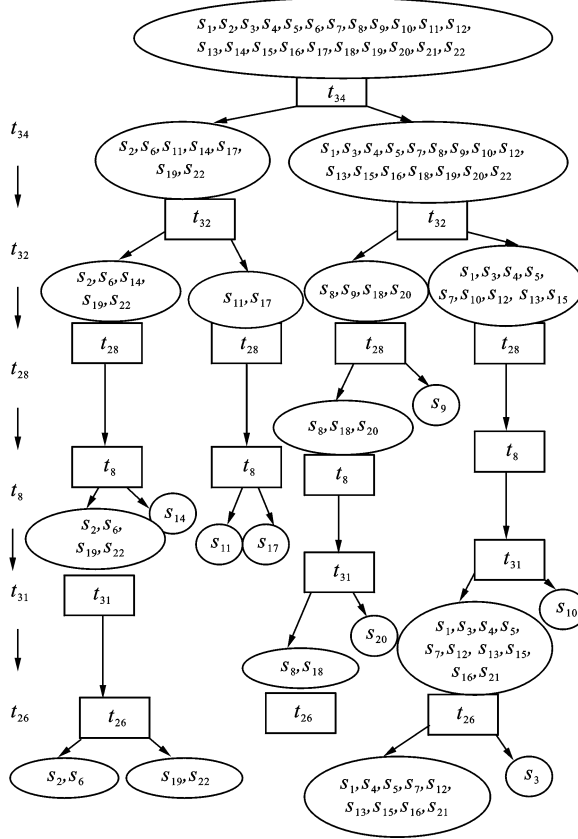


图 3 由 IVDE 求得的测试序列(6 个测试, FIR>90%)

Fig. 3 Solution of 6 tests to achieve FIR>90% by IVDE

表 6 超外差接收机不同方法求得平均测试代价比较

Tab. 6 Average test cost of different methods on super-heterodyne receiver

算法	优化适应度均值与方差
CLPSO	3.353 1±1.80E-03
CoDE	3.351 3±9.06E-16
JADE	3.354 5±4.98E-03
IVDE	3.349 0±7.56E-04

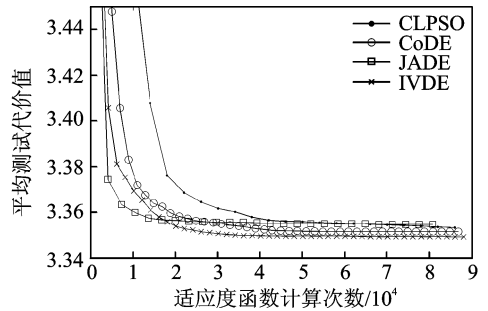


图 4 各种优化算法收敛曲线比较图

Fig. 4 Convergence graph of different algorithms

5 结束语

针对测试序列优化问题,提出惯性速度差分进化算法,通过附加的惯性速度因子项,可避免陷入局部最优解,从而求得更好的结果。结合故障隔离率要求,提出了适合差分进化个体状态向量的编码方式,进而提出了新的个体适应度函数计算方法,可用于解决大型系统的测试序列优化问题。通过对一系列问题进行数值计算验证,在平均收敛速度和获取最优解方面,IVDE 体现了收敛速度和全局最优解的折中优势,比其他启发式进化算法如 GA^[13], PSO^[7-10] 得到了更好的结果。但本文还没有深入分析 IVDE 特性与各参数关联的变化规律、求解测试序列优化问题的编码优缺点,还没有研究如何将 IVDE 应用到

求解多重故障问题的测试序列优化问题,这些都有待进一步研究。

参考文献:

- [1] 张华春, 岳海霞, 禹卫东. 环境一号 C 卫星 SAR 闭环测试系统的设计与实现[J]. 数据采集与处理, 2014, 29(4): 654-660.
Zhang Huachun, Yue Haixia, Yu Weidong. Design and implementation of HJ-1-C satellite SAR closed-loop test system[J]. *Journal of Data Acquisition and Processing*, 2014, 29(4): 654-660.
- [2] Pattipati K R, Alexandridis M. Application of heuristic search and information theory to sequential fault diagnosis[J]. *IEEE Trans on SMC*, 1990, 20(4): 872-887.
- [3] Raghavan V, Shakeri M, Pattipati K R. Optimal and near optimal test sequencing algorithms with realistic test models [J]. *IEEE Trans on SMC*, 1999, 29(1): 11-27.
- [4] Hu Z, Zhang S, Yang Y, et al. Test sequencing problem considering life cycle cost based on the tests with non-independent cost[J]. *Chemical Engineering Transactions*, 2013, 33(3): 253-258.
- [5] Tu Fang, Pattipati K R, Deb S, et al. Computationally efficient algorithms for multiple fault diagnosis in large graph-based systems[J]. *IEEE Trans on SMC*, 2003, 33(1): 73-85.
- [6] Kundakcioglu O E, ünlüyurt T. Bottom-up construction of minimum-cost AND/OR trees for sequential fault diagnosis[J]. *IEEE Trans on SMC*, 2007, 37(5): 621-629.
- [7] 于劲松, 徐波, 李行善. 基于遗传算法的序贯诊断测试策略生成[J]. 系统仿真学报, 2004, 16(4): 833-836.
Yu Jinsong, Xu Bo, Li Xingshan. Generation of test strategy for sequential fault diagnosis based on genetic algorithms[J]. *Chinese Journal System Simulation*, 2004, 16(4): 833-836.
- [8] 蒋荣华, 王厚军, 龙兵. 基于 DPSO 的改进 AO* 算法在大型复杂电子系统最优序贯测试中的应用[J]. 计算机学报, 2008, 31(10): 1835-1840.
Jiang Ronghua, Wang Houjun, Long Bing. Applying improved AO* based on DPSO algorithm in the optimal test sequencing problem of large scale complicated electronic system[J]. *Chinese Journal of Computers*, 2008, 31(10): 1835-1840.
- [9] Lian Guangyao, Huang Kaoli, Chen Jianhui, et al. Optimization method for diagnostic sequence based on improved particle swarm optimization algorithm[J]. *Chinese Journal of Systems Engineering and Electronics*, 2009, 20(4): 899-995.
- [10] Qiu Xiaohong, Liu Jun, Qiu Xiaohui. Random DBPSO algorithm application in the optimal test-sequencing problem of complicated electronic system[C]//2nd International Conference on Computer and Automation Engineering, ICCAE 2010, NJ, Singapore; IEEE, 2010: 107-111.
- [11] Srivatsava P R, Mallikarjun B, Yang X S. Optimal test sequence generation using firefly algorithm[J]. *Swarm and Evolutionary Computation*, 2013, 8(1): 44-53.
- [12] Pan Jialiang, Ye Xiaohui, Xue Qiang. A new method for sequential fault diagnosis based on ant algorithm[C]//Second International Symposium on Computational Intelligence and Design. Changsha, China; IEEE, 2009: 44-48.
- [13] Yang Chenglin, Yan Junhao, Long Bing, et al. A novel test optimizing algorithm for sequential fault diagnosis[J]. *Microelectronics Journal*, 2014, 45(6): 719-727.
- [14] Zhang J, Sanderson A C. JADE: Adaptive differential evolution with optional external archive[J]. *IEEE Trans Evolut Comput*, 2009, 13(5): 945-958.
- [15] Wang Yong, Cai Zixing, Zhang Qingfu. Differential evolution with composite trial vector generation strategies and control parameters[J]. *IEEE Trans Evolutionary Computation*, 2011, 15(1): 55-66.
- [16] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proceedings of IEEE Conference on Neural Networks. IV, Piscataway, NJ: IEEE Press, 1995: 1942-1948.
- [17] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE Trans Evolutionary Computation*, 2006, 10(3): 281-295.

作者简介:



邱晓红(1967-),男,教授,研究方向:智能计算和系统仿真, E-mail: jxauqiu@163.com。



李渤(1979-),男,讲师,研究方向:智能计算。



李靖(1969-),女,助理实验员,研究方向:计算机信息处理技术。

