

面向系统级故障诊断的高效遗传算法

邓 伟 杨小帆 吴中福

(重庆大学计算机学院 重庆 400044)

摘 要 Elhadeb 和 Ayeb 首次提出采用遗传算法来进行系统级故障诊断,其适应度函数通过比较实际症候与当前猜测故障集产生的症候得到.上述算法的一个缺点是其适应度函数只考虑了故障集随机生成的一个症候,因而会漏掉绝大多数有效的故障集.对此首先针对 PMC 模型提出结点状态与诊断图中一定症候相容时结点状态应满足的方程,然后通过设计基于该方程的适应度函数,提出针对 t -可诊断系统的遗传算法.理论分析和模拟实验均表明文中算法在迭代步数上大大地优于原算法.此外,还确认了 Elhadeb 提出的产生初始种群的方法的高效性.

关键词 系统级故障诊断;遗传算法;诊断度;PMC 模型

中图法分类号 TP306

An Efficient Genetic Algorithm for System-Level Diagnosis

DENG Wei YANG Xiao-Fan WU Zhong-Fu

(College of Computer Science, Chongqing University, Chongqing 400044)

Abstract Elhadeb and Ayeb devised a genetic algorithm for the system-level diagnosis of multi-computers, where the fitness function is calculated by comparing the given syndrome with the syndrome randomly produced by the current guess fault set. One demerit of this algorithm is that this fitness function takes only one syndrome from many possible candidates, leading to a high probability of incorrect diagnosis. In the present paper, the authors describe a set of equations that govern the statuses of the units in a system. Based on this, the authors present a new genetic algorithm for the fault diagnosis of diagnosable systems by designing a novel fitness function. Theoretical analysis and simulation result both show that the algorithm is remarkably superior to the Elhadeb-Ayeb's diagnosis algorithm in terms of the number of iterations. The initial population production process proposed by Elhadeb and Ayeb is also justified.

Keywords system-level diagnosis; genetic algorithms; diagnosability; PMC model

1 引 言

随着信息技术的高速发展,为了满足宇宙学、气象、流体力学、材料等重大领域科学计算的需要,大规模并行处理机向着高性能、大规模、多样性、多功能的方向发展.随之而来的一个问题就是系统的容

错性和维护问题,特别是在应用于某些重要部门的多机系统中^[1].由于系统的规模越来越大,从中发现故障结点的工作也越来越繁重,花费也越来越高.随着芯片集成技术的提高,目前的处理结点一般都具备了一定的通信和处理能力,那么一个很自然的想法就是充分利用结点这些能力,让它们相互测试,然后再根据测试结果和拓扑结构等条件,最终得出正

确的诊断结果,这就是系统级故障诊断的基本思想.

Preparata, Metze 和 Chien 于 1967 年首次将系统故障诊断问题归结为一个图论问题^[2]. 他们针对结点(处理器)采用点到点连接的系统提出 PMC 诊断模型,模型假设系统中每个结点只有两种状态:有故障、无故障,且假设结点的故障为永久性故障. 系统用有向图 $G(U, E)$ 表示,其中结点代表处理器,边代表处理器之间的连接,结点总数用 n 表示,结点用 $U = \{u_0, \dots, u_{n-1}\}$ 表示. 系统中所有故障结点的集合 F 成为故障集.

诊断在处理器对其它直接连接处理器测试的结果上进行,每次测试都包括一个测试者和一个被测试者. 测试者 u_i 给被测试者 u_j 发送一个测试信息,被测试者 u_j 处理完该测试信息后给测试者 u_i 返回一个测试结果,测试者 u_i 把该测试结果与自己预想的结果进行比较,如果比较后相同,则认为被测试者 u_j 无故障,把边 (u_i, u_j) 的权值 a_{ij} 定为 0; 如果比较不相同,则认为被测试者 u_j 有故障,把边 (u_i, u_j) 的权值 a_{ij} 定为 1. 也就是说,边 $(u_i, u_j) \in E$ 当且仅当结点 u_i 对结点 u_j 进行了测试. 所有测试结果的集合通过系统外的可信结点收集并称为系统的症候 S . PMC 模型假设无故障结点总是能够检测出故障结点,而故障结点检测其它结点的结果完全不可信. 用 1/0 分别表示结点状态有/无故障,则 PMC 模型具体定义见表 1.

表 1 PMC 模型

| 标号 | 测试者 u_i 状态 | 被测试者 u_j 状态 | 测试结果 a_{ij} 状态 |
|----|--------------|---------------|------------------|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0/1 |
| 4 | 1 | 1 | 0/1 |

根据诊断模型可以看出,当故障集已知时,可很容易获得故障症候. 而实际情况是,根据结点间测试可以先获得故障症候 S (即测试结果 a_{ij} 已知),然后采用一定的故障诊断算法反过来确定故障结点集合 F ,这就是系统级故障诊断要解决的问题. 诊断算法将系统中所有结点集 U 划分为三个子集:故障集 F^* 和无故障集 FF 和可疑结点集 SF . 如果 $F^* \subseteq F$ 且 $FF \subseteq U - F^*$,则称诊断算法是正确的. 如果 $SF = \emptyset$,则称诊断算法是完全的.

经典的故障诊断系统中有一个非常重要的参数:诊断度(diagnosability). 诊断度是系统允许的最大故障结点数 t ,且当系统中故障结点数不超过 t 时,存在正确并完全的诊断算法. 在满足诊断度 t 的

条件下,把正确且完全的诊断算法叫作一步 t -故障诊断,把一次至少能诊断出一个故障结点的诊断算法叫作顺序 t -故障诊断. 当 $2 \times t + 1 \leq n$ 时,系统的诊断度 t 受诊断图 G 的最小结点连通度限制^[2]. 文献[3]给出 t -故障诊断系统的特点,文献[4]给出一个 PMC 模型下通用的一步 t -故障诊断算法,文献[5]为确定性故障诊断算法的最新进展.

除了上述的确定性诊断算法, Scheinermann 提出概率故障诊断^[6],这种诊断不要求绝对正确地完成诊断任务,只要求高概率地正确完成诊断任务. 当系统具有 $O(n \cdot \log n)$ 条测试边时, Blough 等^[7] 给出一个概率趋于 1 的正确诊断算法,其主要思想是每个结点的状态(有或者无故障)通过测试它的邻接结点进行多数表决产生. 随后,文献[8]采用神经网络对算法进行了一些改进. 文献[9]对概率故障诊断作了较为详细的综述,文献[10]对故障诊断整个领域作了较为详细的综述.

演化算法是根据生物学上适者生存的演化原则发展起来的一种通用的问题求解方法. 它采用简单的编码技术来表示各种复杂的结构,并通过对一组编码表示进行简单的遗传操作和优胜劣汰的自然原则来知道学习和确定搜索的方向. 由于它采用种群的方式搜索,这使得它可以同时搜索解空间内的多个区域. 在赋予演化算法自组织、自适应、自学习等特征的同时,优胜劣汰的自然选择和简单的遗传操作使演化计算具有不受搜索空间限制性条件(如可微、连续、单峰等)的约束及不需要其它辅助信息(如导数)的特点. 这些特点使得演化算法不仅能获得较高的效率而且具有简单、易于操作和通用的特征,而这些特征正是演化计算越来越受到人们青睐的原因之一. 据德国 Dortmund 大学的一份研究报告,根据不完全统计演化算法已在 16 个大领域 250 个小领域中获得了应用^[11].

演化算法的一个非常重要的分支是遗传算法(genetic algorithm). 遗传算法主要由 Holland 创立并应用于优化和机器学习等问题中^[12]. 遗传算法的操作对象是一群二进制串(称为染色体、个体),即种群 $P(t)$. 每个染色体对应于问题的一个解,从初始种群出发,采用基于适应值(由适应度函数产生)比例的选择策略在当前种群中选择个体,使用杂交和变异来产生下一代种群. 如此一代代演化下去,直到满足期望的终止条件. 遗传算法作为一种鲁棒性很强的随机搜索算法较好地解决了在机器学习、过程控制、经济预测、工程优化等领域大量的优化问题^[13].

如前所述,研究系统级故障诊断的主要方法为图论和概率论方法。由于系统中所有的结点的故障状态(0/1)可以很自然地对应于染色体中的基因排列,又因为遗传算法的大量优点,Elhadeh 和 Ayebe 在文献[14]中首次采用演化算法对 PMC 模型下的一步 t -可诊断系统进行故障诊断研究,随后在文献[15-16]中采用演化算法对比较模型的系统级故障诊断进行了研究。

故障诊断算法对于给出目标症候 S^* ,要求出产生该症候的故障集 F^* 。文献[14]首先用长度为 n 的个体来表示搜索空间(即所有结点的故障状态组合),也就是说每个个体表示一个故障集 F 。然后根据故障诊断的特点设计了遗传算法的适应度函数和遗传操作符(繁殖、杂交和变异算子),通过算法的多步迭代最后求出目标故障集 F^* 。通过对该遗传算法的模拟实验,证明算法是收敛和有效的。

遗传算法中一个最重要的部分是适应度函数的设计。文献[14]中个体的适应值是根据个体随机产生一个症候 S ,比较 S 与目标症候 S^* 的相似度得到。由于一个个体 F (故障集)可以对应于多个症候,所以即使个体为所求个体时($F=F^*$),症候与目标症候也不一定相同(多数情况下, $S \neq S^*$)。

本文根据 PMC 模型下系统级故障诊断的特点,首先提出满足诊断图中一定症候时结点应满足的充分必要条件,然后根据该充分必要条件提出相应的适应度函数,该适应度函数能够保证当 $F=F^*$ 时, $S=S^*$ 。根据新的适应度函数,我们设计了新的更有效的遗传操作符。并在理论上证明本文遗传算法优于文献[14]中算法,模拟实验的结果也证明了这一点。

从第 2 节开始,首先,我们给出文中用到的一些定义。然后,给出文献[14]的大致算法,并指出其不足之处。再次,给出新的适应度函数、遗传操作符和一个改进的遗传算法同时也给出了一些相关的理论证明。最后是我们算法的实现和两个算法的比较。

2 相关定义

在本文中,对于集合 X , $|X|$ 表示该集合的元素个数。一个包括 n 个处理机的系统,可用集合 $U = \{u_0, \dots, u_{n-1}\}$ 来表示。每个结点 $u_i \in U$ 被多个其它的结点测试。对于每个结点 u_i ,定义 u_i 的被测试者集合为 $\Gamma(u_i) = \{u_j : (u_i, u_j) \in E\}$,测试者集合为 $\Gamma^{-1}(u_i) = \{u_j : (u_j, u_i) \in E\}$, u_i 的入度 $d_{in}(u_i) =$

$|\Gamma^{-1}(u_i)|$ 和出度 $d_{out}(u_i) = |\Gamma(u_i)|$ 。用 S 表示系统的症候,即 $S: E \rightarrow \{0, 1\}$,对所有的边 $(u_i, u_j) \in E$,定义 $S(u_i, u_j) = a_{ij}$ 。用 $S(u_i)$ 和 $S^{-1}(u_i)$ 表示症候中相关的被测试者集和测试者集,即 $S(u_i) = \{S(u_i, u_j) : u_j \in \Gamma(u_i)\}$, $S^{-1}(u_i) = \{S(u_j, u_i) : u_j \in \Gamma^{-1}(u_i)\}$ 。

症候 S 与故障模式 F 相容,当对所有 $(u_i, u_j) \in E$, $u_i \in U - F$ 且 $S(u_i, u_j) = 1$ 当且仅当 $u_j \in F$ 。测试图 $G(U, E)$ 为 $D_t(n)$ 设计当且仅当对任意 $u_i \in U$, $|\Gamma^{-1}(u_i)| = t$ 且任意结点不进行相互测试。Hakimi 和 Amin 在文献[3]证明了 $D_t(n)$ 为一步 t -可诊断系统。图 1 为一个 $n=9, t=4$ 的 $D_t(n)$ 设计。定义 $D_{\delta,t}(n)$ 系统为对任意两个结点 u_i 和 u_j , $j-i = \delta \cdot p \bmod n$, 其中 $p=1, 2, \dots, t$ 。很明显 $D_{\delta,t}(n)$ 系统是 $D_t(n)$ 系统。文献[2]证明当 $n \geq 2t+1$ 时, $D_{\delta,t}(n)$ 系统为 t -可诊断系统,且当 δ 和 n 互质时, $D_{\delta,t}(n)$ 同构于 $D_{1,t}(n)$ 。

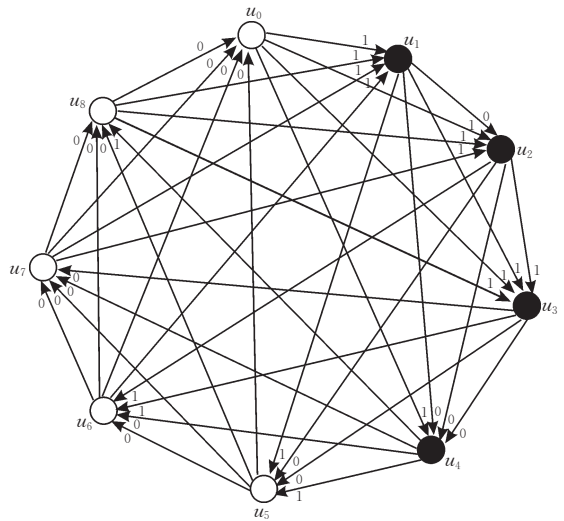


图 1 一个 $n=9, t=4$ 的 $D_t(n)$ 设计, 故障结点用实心黑点表示

用长度为 n 的字符串来表示每个个体,字符串第 i 位用 1 或 0 来代表相应结点机 u_i 有无故障。对于每个个体 v ,用 $F(v)$ 代表 v 中所有的故障处理机集合(那些为 1 的基因)。用 $Elite$ 表示种群中适应值最高的个体,用 $FT(v_i)$ 表示适应度函数。在遗传算法中,用 p_c 表示杂交概率,用 p_m 表示变异概率,用 pop_size 表示的种群大小。

3 Elhadeh 的演化算法概略和不足

Elhadeh 演化算法具体如下。

输入: 测试图 $G(U, E)$ 和测试得到的症候 S^*

输出: 故障结点集 F 和无故障结点集 FF

Begin

1. 首先随机产生 pop_size 个个体,形成初始种群 Pop ;
2. 对 Pop 中的每个个体 $v \in Pop$, 计算其适应度 $FT(v)$;
3. 当 $(\forall v \in Pop, FT(v) \neq 1)$ 循环以下语句

将种群 Pop 中适应度最高的个体存入 Elite;
繁殖(Pop);
变异(Pop);
杂交(Pop);
对每个个体 $v \in Pop$, 计算其适应度函数 $FT(v)$;
用 Elite 替换种群中适应度最低的个体;

4. $F \leftarrow F(v^*)$; $v^* \in Pop$ 且 $FT(v^*) = 1$ 和 $FF \leftarrow U - F$

End

对于每个个体 v , 设该个体对应的故障模式为 F , S 为与 F 相容的一个症候, 算法中的适应度函数 FT 定义如下:

$$FT(v) = \frac{\sum_{i=0}^n f(v[i])}{n} \tag{1}$$

其中 $f(v[i]) = \frac{f_{out}(v[i]) + f_{in}(v[i])}{2}$ (2)

且 $f_{in}(v[i]) = \frac{|S^{-1}(u_i) \cap (S^*)^{-1}(u_i)|}{d_{in}(u_i)}$

$$f_{out}(v[i]) = \begin{cases} 1, & d_{out}(u_i) = 0, \\ \frac{|S(u_i) \cap S^*(u_i)|}{d_{out}(u_i)}, & \text{其它} \end{cases}$$

即个体的适应度是个体表示的每一位(代表一个结点机)适应度 $f(v[i])$ 的平均, 由于每个个体 v 唯一对应于一故障模式 F , S 是 F 产生的任一相容症候, 上式中 $f(v[i])$ 即为 S 与目标症候 S^* 相同边与总边的比值. 文献[14]中提到当 $F(v) = F^*$ 时, $FT(v) = 1$. 这个结论在一般情况下是不成立的, 见以下定理 1.

引理 1. PMC 模型下, 设故障结点为 $\{vf_1, vf_2, \dots, vf_m\}$, 则与一个故障模式 F 相容的症候数

目为 2^{nf} , 其中 $nf = \sum_{i=1}^m d_{out}(vf_i)$.

证明. 由于 PMC 模型中故障结点诊断其它结点的结果是任意的, 所以一个故障模式 F 可以产生与之相容的多个症候, 设为 $\{S_1, \dots, S_i, \dots, S_n\}$, 其中症候数目 SN 与故障结点数目 m 和故障结点与其它结点相连的拓扑有关. 所有故障结点发出的测试边数目为 $nf = \sum_{i=1}^m d_{out}(vf_i)$, 由于故障结点测试其它结点的结果任取 0 或者 1, 所以症候数 $SN = 2^{nf}$. 也就是说随着故障结点数目的增加, 症候数目呈指数增加. 故命题得证. 证毕.

定理 1. 符号定义如前, 当 $F(v) = F^*$ 时,

$FT(v) = 1$ 的概率为 $1/SN$.

证明. 设 $F(v)$ 为所求的故障模式(即 $F(v) = F^*$), 目标症候为 S_i (即 $S^* = S_i$), 由引理 1 得 $F(v)$ 产生目标症候 S^* 的概率为 $1/SN$, 即当 $F(v) = F^*$ 时, 症候 $S(u_i)$ 和 $S^{-1}(u_i)$ 与 $S^*(u_i)$ 和 $(S^*)^{-1}(u_i)$ 相同的概率为 $1/SN$. 而只有 $S(u_i)$ 和 $S^{-1}(u_i)$ 与 $S^*(u_i)$ 和 $(S^*)^{-1}(u_i)$ 相同时, $FT(v) = 1$. 故命题得证. 证毕.

虽然文献[14]中存在这一点瑕疵, 但是其算法正确性证明是正确的, 也就是说算法是收敛的. 因为当 $F(v) = F^*$ 时, 虽然 $FT(v) = 1$ 的概率为 $1/SN$, 但是由于 $F(v)$ 的适应度很高, 所以 $F(v)$ 仍然保持在种群中, 在以后的演化过程中仍有机会达到 $FT(v) = 1$ 而使算法收敛, 只是收敛速度可能要慢.

4 一个改进的演化算法

定理 2. 在 PMC 模型下, 如果故障集与症候相容, 那么

- ① 当 $a_{ij} = 1$ 时, 则 u_i 和 u_j 一定满足 $(1 - u_i) \cdot (1 - u_j) = 0$;
- ② 当 $a_{ij} = 0$ 时, 则 u_i 和 u_j 一定满足 $(1 - u_i) \cdot u_j = 0$.

证明. 当 a_{ij} 为 1 时, 从表 1 中可查出 u_i 和 u_j 的状态可为 2~4 三种情况, 将这三种情况下的 u_i 和 u_j 的状态代入 $(1 - u_i) \cdot (1 - u_j)$, 结果均为 0, 等于方程右边, 所以①成立;

当 a_{ij} 为 0 时, 从表中可查出 u_i 和 u_j 的状态可为 1, 3, 4 三种情况, 将这三种情况下的 u_i 和 u_j 的状态代入 $(1 - u_i) \cdot u_j$, 结果均为 0, 等于方程右边, 所以②成立. 故定理成立. 证毕.

定理 2 中方程是基于表 1 构造的, 用该方程可以根据症候来约束故障集, 所以可以广泛地应用于故障诊断当中. 对于其它故障诊断模型可以构造类似的定理, 我们将在后续的文章中进行讨论.

下面, 在介绍了遗传算法的一些要素后, 基于以上定理给出我们的遗传算法. 众所周知, 遗传算法一般分为 4 步:

- (1) 选择合适的字串方法来表示搜索空间的个体;
- (2) 设计遗传操作符;
- (3) 选择合适的适应度函数;
- (4) 选择合适的遗传操作符的参数来控制搜索过程.

4.1 字串表示

用 1 或 0 结点表示有或无故障,那么对于一个有 n 个结点的系统,就用一个长度为 n 、每个字节为 0/1 的字串来表示. 如用染色体(100100010)表示 9 个结点的系统,其中故障模式 $F = \{u_0, u_3, u_7\}$. 由于 $D_i(n)$ 系统规定系统中故障结点数目 $|F| \in [1, t]$, 所以 1 的位数大于 t 或者没有为 1 的位的染色体为无效的个体.

4.2 初始种群

文献[14]中提出两种产生初始种群的方法: (1) 随机产生初始个体, 进而形成种群; (2) 任意指定一个结点为无故障, 然后根据 PMC 模型和测试图的症候产生一个初始个体. 文献[14]通过实验对比, 得出方法 2 大大优于方法 1. 在 δ 和 n 互质的 $D_{\delta, t}(n)$ 系统(即 $D_{1, t}(n)$ 系统)下, 这是可以理论证明的问题.

$D_i(n)$ 中由于 t 为 $n/2$ 的下取整, 而故障数目为 $[1, t]$ 之间的一个随机变量, 所以故障数目的均值为 $t/2 \approx n/4$, 即在 $D_{1, t}(n)$ 每个结点的故障概率约为 $1/4$. 由于个体中每一位对应于一个结点, 所以个体中每一位为故障的概率为 $1/4$, 即个体中每一位为无故障的概率为 $3/4$.

方法 1 每一位的选择为独立同分布的, 由于个体中每个结点正确的概率为 $3/4$, 那么个体正确的概率为 $(3/4)^n$, 该概率随着 n 的增加越来越小.

本文采用的方法类似于文献[14]的方法 2. 由于我们研究遗传算法在系统级故障诊断的最终目标是得到一个较为通用故障诊断遗传算法, 这样, 产生的个体中可能存在不能确定的位, 这些位我们就随机指定了. 具体产生过程如下.

算法 1. 随机指定一位无故障产生初始种群(称为指定无故障结点方法).

1. 随机指定个体中一个结点 i 无故障.
2. 根据该结点的出边, 按测试边的权值确定相连结点的状态, 根据该结点的入边, 若该边的起始结点认为 i 有故障, 则起始结点有故障; 按宽度优先原则, 对确定的无故障结点继续以上过程; 直到所有的无故障结点循环完毕.
3. 对未确认的结点随机指定状态.
4. 如果生成的个体未满足 $D_i(n)$ 系统的条件(故障数目在 $[1, t]$ 之间), 回到第 1 步.

首先证明该算法产生的个体是满足 $D_i(n)$ 系统的条件的.

在第 1 步时, 假如 i 为故障结点. 如果个体的每一位状态满足随机分布, 又因每个结点的故障状态独立, 且故障概率为 $1/4$, 那么第 3 步生成的个体故

障数目 x 满足二项分布 $x \sim b(n, 1/4)$, 则 x 在 $[1, n/2]$ 中的概率 P_F 为 $\sum_{k=1}^{n/2} C_n^k \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{n-k}$, 该概率是趋进于 1 的. 因为 n 远大于故障概率 $1/4$, 由二项分布的柏松近似可得

$$P_F \approx \sum_{k=1}^{n/2} \frac{(n/4)^k}{k!} e^{-n/4}.$$

当 n 从 5 增加到 100 时, P_F 值的变化如图 2 所示. 从图中可以看出, 当结点数目 n 大于 11 时, 单次采用上述方法产生的个体满足 $D_i(n)$ 系统的条件的概率超过 90%. 当结点数目 n 大于 25 时, 满足 $D_i(n)$ 系统的条件的概率超过 99%. 所以该方法产生的绝大部分个体均能在 4 步成功结束.

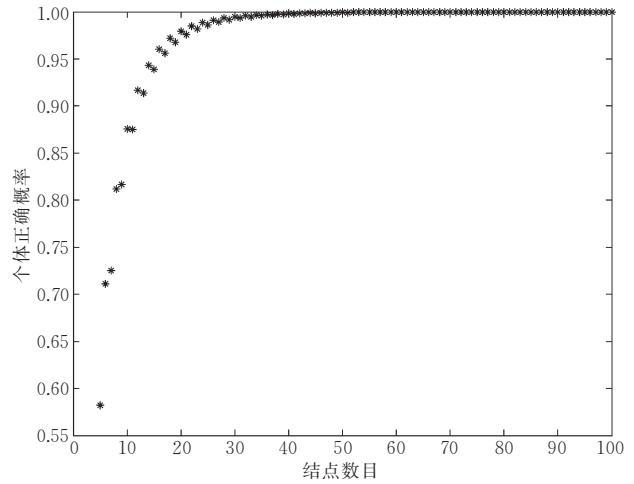


图 2 单次执行指定结点无故障方法产生的个体满足 $D_i(n)$ 系统的条件的概率 P_F

下面讨论该算法产生的个体是目标个体的概率.

在 $D_{1, t}(n)$ 系统下, 一个无故障结点可以通过测试边确定所有其余结点的状态, 那么上述算法中任选无故障结点根据 PMC 模型和测试图的症候产生个体的过程为确定性的, 所以产生个体正确的概率取决于任选结点无故障的概率. 如果第 1 步时 i 为无故障结点, 那么根据以上步骤产生的个体一定是正确的, 即个体正确的概率为 1. 又因为 i 为故障结点的概率为 $1/4$, 第 4 步重新循环的概率为 $(1 - P_F)$, 所以按上述算法产生的个体为正确的概率为

$$3/4 \cdot P_F + [(1 - P_F)/4] \cdot 3/4 \cdot P_F + [(1 - P_F)/4]^2 \cdot 3/4 \cdot P_F + \dots = 3P_F/(3 + P_F).$$

即当 n 很大时, $P_F \approx 1$, 那么正确个体的概率为 $3/4$.

指定无故障结点方法的复杂度分析如下: 第 1 步需常数时间; 第 2 步的最坏时间复杂度为 $O(|E|) = O(n^2)$; 第 3 步为 $O(n)$; 第 4 步应该分为

两种情况,未满足 $D_i(n)$ 系统的条件则重新执行,满足则结束. 由于 i 为无故障结点 (概率 $3/4$) 时,第 4 步生成的个体一定满足条件,只有 i 为故障结点 (概率 $1/4$) 时,第 4 步生成的个体以概率 $(1-P_F)$ 不满足条件. 所以总复杂性 $\Omega \leq (3/4 \times O(n^2) + 1/4 \times \Omega)$, 解此不等式得 $\Omega \leq O(n^2)$.

如果对指定无故障结点方法的第 2 步加以修改,添加根据出边的限制来决定其他结点等技巧,可以使 i 为无故障结点时不存在不确定结点;而且通过所有出入边决定的结点状态是否冲突可以判断出 i 为故障结点的状态. 那么当 i 为无故障结点时复杂度为 $O(n^2)$;当判断 i 为故障结点时,使用经典的确定性的故障诊断算法 $O(n^{2.5})$ [10,12],这样可使 t 可诊断系统的实际运行时间降低;并且通过增加个体的方法可使算法的复杂性进一步降低. 详细的算法和分析,我们将于下一步研究中给出.

我们又进行了实验仿真. 在满足 $D_i(n)$ 系统条件下,每次随机产生目标故障集 F^* 和相应的症候 S^* ,按以上步骤生成种群并比较种群中个体与 F^* 相同的概率. 我们固定种群大小为 20,通过对不同结点数目 n 的比较,得到下图 (图 3). 我们可以看到大多数的个体是正确的. 经过对下图的统计,得出个体为待求个体 F^* 的概率大约为 0.85.

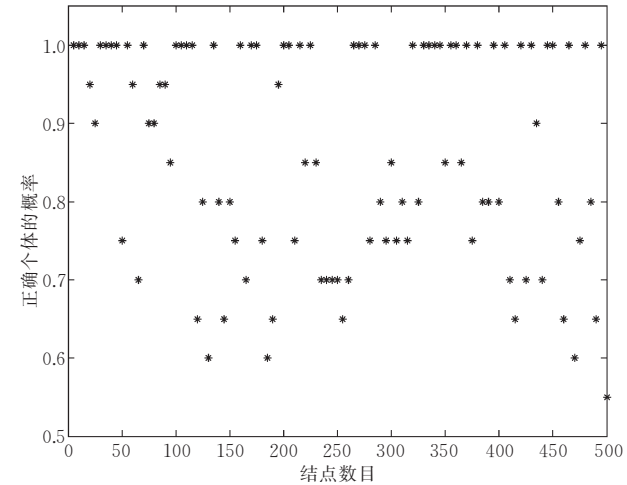


图 3 指定无故障结点方法产生个体的正确率

在概率故障诊断中多数表决方法 (如果结点的多数测试边认为无故障,则结点状态为无故障) 得到很广泛的应用. 我们将指定无故障结点方法与多数表决方法进行了比较. 在种群数目为 20, 结点从 5 增加到 500 的情况下,多数表决方法产生个体的正确率为 0.02,而指定无故障结点方法的正确率为 0.85. 由于多数表决方法产生的正确个体太少,所以在图 4 中我们只给出多数表决方法产生个体的

每一位与待求个体 F^* 比较匹配的概率,其均值约为 0.87.

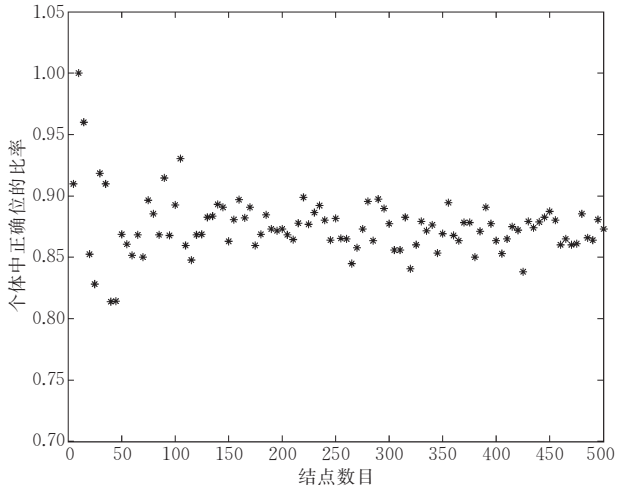


图 4 多数表决方法产生的个体每一位正确的概率

4.3 适应度函数

本文中的适应度函数与文献[14]中适应度函数的区别在于 $f_{in}(v[i])$ 和 $f_{out}(v[i])$ 定义的不同,文献[14]是根据症候直接比较得到的,而我们是根据结点 $v[i]$ 和 $v[j]$ 的值是否满足定理 2 中方程得到的.

我们先根据 S^* 中任意一条边 (u_i, u_j) 的权值 a_{ij} 定义结点 u_i 到 u_j 的约束方程 $F(u_i, u_j)$:

$$F(u_i, u_j) = \begin{cases} (1 - u_i) \times (1 - u_j) = 0, & a_{ij} = 1 \\ (1 - u_i) \times u_j = 0, & a_{ij} = 0 \end{cases}$$

设 $nf_{in}(v[i]) = |\{(u_j, u_i) : u_j \in \Gamma^{-1}(u_i) \text{ 且 } F(u_j, u_i) \text{ 成立}\}|$, $nf_{out}(v[i]) = |\{(u_i, u_j) : u_j \in \Gamma(u_i) \text{ 且 } F(u_i, u_j) \text{ 成立}\}|$.

然后定义 $f_{in}(v[i])$ 和 $f_{out}(v[i])$ 如下

$$f_{in}(v[i]) = nf_{in}(v[i]) / d_{in}(u_i) \quad (3)$$

$$f_{out}(v[i]) = \begin{cases} 1, & d_{out}(u_i) = 0 \\ \frac{nf_{out}(v[i])}{d_{out}(u_i)}, & \text{其它} \end{cases} \quad (4)$$

而 $f(v[i])$ 和 $FT(v)$ 的定义同式 (1), (2).

定理 3. 对于如 (1) (2) (3) (4) 定义的适应度函数 $FT(v)$ 而言,当且仅当一个染色体 v 表示的故障模式 $F(v) = F^*$ 时, $FT(v) = 1$.

证明.

充分性. 当 $F(v) = F^*$ 时, S^* 与 $F(v)$ 相容,即在 PMC 模型下,症候 S^* 中任意边 (u_j, u_i) 的权值 a_{ij} 一定可由 u_i 和 u_j 的故障状态产生. 由定理 2, 当 a_{ij} 的权值为 0 (或 1) 时, u_i 和 u_j 的故障状态满足等式 $(1 - u_i) \cdot u_j = 0$ (或 $(1 - u_i) \cdot (1 - u_j) = 0$), 即满足方程 $F(u_i, u_j)$. 故对症候 S^* 的任意两个结点 u_i 和

u_j , 如果存在测试边 (u_j, u_i) , 则 u_i 和 u_j 的状态一定满足方程 $F(u_i, u_j)$. 也就是说 $d_{in}(u_i) = |\{(u_j, u_i) : u_j \in \Gamma^{-1}(u_i) \text{ 且 } F(u_j, u_i) \text{ 成立}\}| = nf_{in}(v[i])$, $d_{out}(u_i) = |\{(u_i, u_j) : u_j \in \Gamma(u_i) \text{ 且 } F(u_i, u_j) \text{ 成立}\}| = nf_{out}(v[i])$. 那么由 $f_{in}(v[i])$ 和 $f_{out}(v[i])$ 的定义易得 $f_{in}(v[i]) = f_{out}(v[i]) = 1$, 由 (1)(2) 定义可得 $f(v[i]) = 1, FT(v) = 1$.

必要性. 令 F^* 对应染色体为 $F(v)$, 假设存在一个不同于 c 的染色体 $F(v')$ 使得 $FT(v') = 1$. 那么 $F(v)$ 与 $F(v')$ 至少存在一位不相同, 不失一般性, 假设该位为 $v[i]$, 即 $v[i] \neq v'[i]$ (表示 u_i 状态在 $F(v)$ 和 $F(v')$ 中分别取不同的值). 由于 $FT(v') = FT(v) = 1$, 则由式 (3) 得 $nf_{in}(v'[i]) = nf_{in}(v[i]) = d_{in}(u_i)$. 由 $nf_{in}()$ 定义可知, 对任意 $u_j \in \Gamma^{-1}(u_i)$ 且 $F(u_j, u_i)$ 成立, 又因为 $v[i] \neq v'[i]$, 所以所有 u_i 的测试者 $\Gamma^{-1}(u_i)$ 为故障 (注: 因为如果有一个测试者无故障, 在 PMC 模型下, 无故障结点能够正确识别其它结点, 那么 u_i 要么为 0, 要么为 1, 不可能在不同染色体中表现出两种不同状态, 这与 $v[i] \neq v'[i]$ 矛盾). 因为所有 u_i 的测试者为故障, 那么由 $D_t(n)$ 的特点, 每个结点的测试者为 t 个, 系统中故障结点数最多为 t , 即系统的所有故障结点数目为 t , 且均属于 u_i 的测试者集 $\Gamma^{-1}(u_i)$. 那么 u_i 的状态只能为无故障, 因为 u_i 的状态为故障将使系统的故障结点数目超过 t , u_i 的状态只能为无故障与假设 $v[i] \neq v'[i]$ 矛盾, 故必要性得证, 故命题得证. 证毕.

我们将修改文献[14]中算法的适应度函数得到的新算法与原算法进行比较, 实验结果表明, 改进的遗传算法的迭代步数大大优于原算法. 在 $D_t(n)$ (其中 $5 \leq n \leq 200$) 系统下, 采用改进算法的决大多数迭代步数为 0, 而原算法的迭代步数均值大于 40.

4.4 演化操作

本文遗传算法大的框架可参看前述的 Elhadeb 演化算法, 但是由于我们采用了不同的适应度函数, 所以在具体的遗传操作又有所不同, 同时对文献[14]中的具体遗传操作进行了一些分析.

4.4.1 繁殖

繁殖是一个基本的遗传操作, 它通过选择种群中具有较高适应度的个体形成下一代的种群. 具体的繁殖操作我们采用轮盘式选择, 轮盘式选择是遗传算法中使用最多的繁殖策略^[13]. 我们采用如下的方法构造轮盘:

1. 累加种群中所有个体的适应度;
2. 计算每个个体 $v_i (i=0, \dots, pop-size-1)$ 在轮盘中的选择概率 p_i ;

3. 计算每个个体 $v_i (i=0, \dots, pop-size-1)$ 的累加选择概率 q_i ;

4. 具体选择过程就是在轮盘中随机选择 $pop-size$ 次, 进而形成新种群. 每次通过如下方法得到一个个体:

- 4.1 产生一个 $[0, 1]$ 之间的随机数 r ;
- 4.2 当 $r < q_0$ 时选择第一个个体 v_0 ; 否则根据 $(q_{i-1} < r \leq q_i)$ 选择第 i 个个体 v_i .

很明显, 通过轮盘形成的新种群中有些适应度高的个体可能出现多次. 最后用上一代中的最优个体 Elite 替换新种群中的最差个体.

4.4.2 变异

文献[14]首先由每个个体的适应度形成选择轮盘, 然后从种群中按概率 p_m 随机选取个体, 再根据个体中每一位的适应度选择变异位置 i^h , 进行变异操作 (0 变 1, 1 变 0). 如果变异后个体为 1 的位 (有故障结点) 不在 $[1, t]$ 的范围内, 如果超过 t , 说明 i^h 以前为 0, 那么将 i^h 的 Tester 中适应度最低且为 1 的位 j^h 改变, 如果 j^h 不满足, 则在 j^h 的 Tester 中继续以上操作. 对变异后个体中没有为 1 的位的情况, 类似处理.

由于变异操作是遗传算法迭代中最常用的操作, 所以我们希望对其进行优化. 为叙述方便, 称文献[14]中变异方法为变异方法 1, 本文中变异方法称为变异方法 2. 变异方法 1 当故障结点数目不在 $[1, t]$ 的范围内时, 通过循环修改入边中最差结点得到最后个体. 本文中当故障结点超过 t 时, 根据每位的适应度随机选择一位进行变异 (变异后故障结点数目为 t), 这样可以重用选择以前的变异位置 i^h 时的数据结构和结果; 当故障结点为 0 时, 根据每位的适应度随机选择一位进行变异 (变异后个体就会有一位故障结点). 当然方法 2 中用到的适应度是根据改进的适应度函数获得.

我们采用多数表决产生初始种群, 并将种群大小 pop_size 从 3 增加到 50, 结点数目 n 从 5 增加到 120, 对不同变异方法 (其余部分算法相同) 的 $D_t(n)$ 系统进行了仿真实验, 如图 5 所示. 经统计, 变异方法 1 的迭代平均步数为 239.2, 变异方法 2 的迭代平均步数为 226.3. 从图中也可看出, 方法 2 的效率较高.

4.4.3 杂交

基于每个个体的适应值形成选择轮盘, 根据杂交概率 p_c 和种群大小 pop_size 从轮盘中随机选择 $p_c \cdot pop_size$ 个个体, 然后再从轮盘中随机选择每个个体的配对个体. 在杂交过程中, 采用单点式杂交, 即随机地在两个父串上选择一个杂交点, 然后交

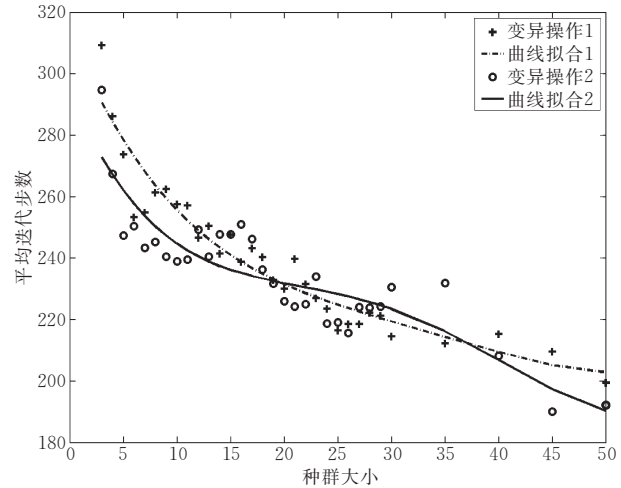


图 5 两种不同变异方法的比较

换这两个串对应的子串. 如果杂交后的两个个体不满足为 1 位的数目在区间 $[1, t]$ 时, 就在父串上选择一个新的杂交点重新进行杂交操作.

5 算法实现与分析

由于在 $D_t(n)$ 系统下, 如果采用指定个体无故障方法产生初始种群, 改进遗传算法的迭代步数太少以至于无法进行参数比较与分析; 另一方面, 对于其它拓扑(如 Hypercuber、Mesh 等常用的稀疏拓扑)和高故障概率诊断, 采用指定个体无故障方法不一定同 $D_t(n)$ 系统一样以很高的概率产生正确个体, 为了下一步研究的需要和保证算法的通用性, 我们采用多数表决来产生初始种群并进行算法分析.

5.1 算法分析

本文算法总框架参照了前面提到的 Elhadeh 演化算法, 但是本文算法的适应度函数和变异操作与其有区别. 由于计算每个个体适应度需要遍历所有的测试边, 所以计算每个的适应度需要 $O(|E|)$, 最坏情况下, 变异操作需要 $O(n \cdot pop-size)$, 这均与 Elhadeh 演化算法的复杂度一样. 所以根据文献[14], 本文算法的最坏复杂度为 $O(|E| \cdot pop-size (\ln pop-size)^2 / \ln r)$, 平均复杂度为 $O(|E| \cdot pop-size \cdot \ln pop-size / \ln r)$, 其中 r 代表个体的适应率. 由定理 1 和定理 3 可知, 当个体为目标个体时, 多数情况下(概率为 $SN - 1/SN$), 采用本文适应度函数的个体适应率大于文献[14]的个体适应率; 同理可证, 当个体部分位与目标个体对应位相同时, 采用本文适应度函数的个体适应率大于文献[14]的个体适应率, 所以采用本文算法的 $\ln r$ 大于文献[14]

算法的 $\ln r$, 即本文算法时空复杂性优于文献[14]算法.

本文算法的正确性可由遗传算法在搜索空间的随机搜索特点(保证算法能够收敛)和定理 3(保证诊断算法的正确性和完全性)保证.

5.2 参数选择

(1) 种群大小 $pop-size$ 的选择

从图 5 可以看出, 不管采用何种变异操作, 随着种群数目的增加, 算法的平均迭代次数将逐渐下降, 这对于算法的并行实现是一个很有利的优点, 但是随着种群数目的增加, 算法的时空复杂度将以 $pop-size \cdot (\ln \cdot pop-size)$ 的速度增加, 所以必须进行折中, 通过实验, 我们发现在单机运行情况下, 种群数目在 7~10 之间比较理想.

(2) 变异概率 p_m 的选择

我们采用多数表决产生初始种群, 并将种群固定为 10, 杂交概率为 0.01, 结点数目 n 从 5 增加到 110, 在不同变异概率 p_m 下对 $D_t(n)$ 系统进行了多次仿真实验, 最后, 算法平均步数的获取如图 6 所示. 从图中可以看出, 个体最佳的变异概率在 0.5 左右, 这与文献[14]中算法的变异概率一致.

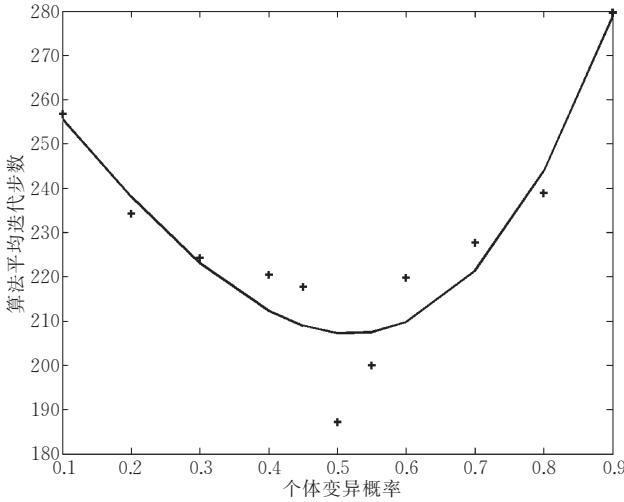


图 6 不同变异概率 p_m 下的平均执行步数

(3) 杂交概率的选择

我们采用多数表决产生初始种群, 并将种群固定为 10, 变异概率为 0.5, 结点数目 n 从 5 增加到 110, 在不同杂交概率 p_c 下对 $D_t(n)$ 系统进行了多次仿真实验, 最后, 算法平均步数的获取如图 7 所示. 从图中可以看出, 当杂交概率大于 0.1 时, 随着杂交概率的增加, 系统的平均迭代步数在减少, 这与文献[14]的杂交概率为 0.01 时系统的迭代步数最小有所不同.

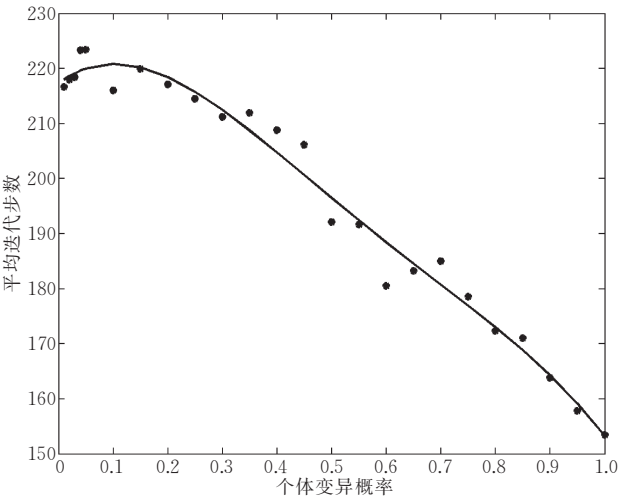


图 7 不同杂交概率 p_c 下的平均执行步数

5.3 算法比较

我们采用文献[14]的算法参数,即种群大小为 7,变异概率为 0.5,杂交概率为 0.01,初始种群用指定结点无故障方法产生,结点数目从 5 增加到 300,记录两个算法的迭代次数如图 8.改进算法(用圆圈表示)大部分在产生初始种群后就结束,平均迭代次数为 1.1,而原算法的平均迭代次数为 45.75,可见改进算法的效率明显高于原算法.

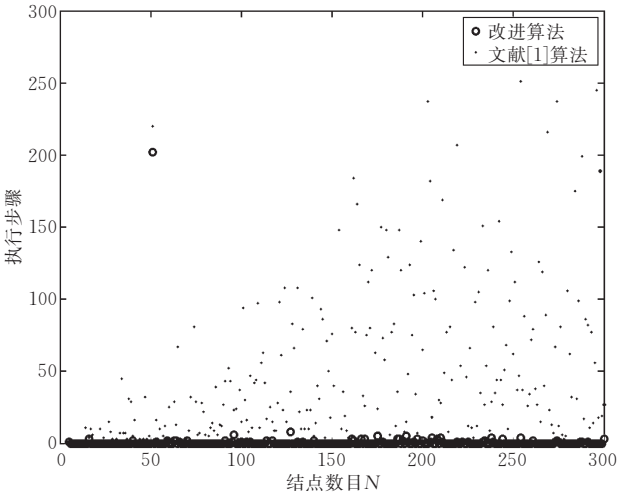


图 8 在文献[14]参数条件下比较改进算法与原算法执行步数

6 结 论

在文献[14]中首次提出采用演化算法来进行系统级故障诊断,其适应度函数通过比较故障模式随机产生的任一症候与目标症候得到. 由于一个故障集可对应于多个症候,而 Elhadeif 的方法只使用其中的一个随机的症候,所以该适应度函数将漏掉绝大多数有效的个体. 本文首先针对 PMC 模型提出

结点状态与诊断图中一定症候相容时结点状态应满足的方程,然后设计基于该方程的适应度函数. 理论分析和模拟实验均表明本文算法在迭代步数上大大地优于原算法. 另外,针对随机产生的不同结点数目的 $D_t(n)$ 测试图,对遗传算法中的参数进行了选择,并分析了遗传算法中初始种群的产生方法和变异算子的选择.

以后的研究方向是在其它模型(如比较模型)和在其它常用拓扑(如 Hypercube、Mesh 等)下研究高效的并行演化算法.

致 谢 匿名审稿人对论文初稿进行了认真审阅,并提出了若干宝贵建议. 在此,作者表示感谢!

参 考 文 献

[1] Chen T H. Fault Diagnosis and Fault Tolerance—A Systematic Approach to Special Topics. Berlin: Springer-Verlag, 1992

[2] Preparata F P, Metze G, Chien R T. On the connection assignment problem of diagnosable systems. IEEE Transactions on Computers, 1967, 16(12): 848-854

[3] Hakimi S L, Amin S L. Characterization of connection assignment of diagnosable system. IEEE Transactions on Computers, 1974, 23(1): 86-88

[4] Dahbura A T, Masson G M. An $O(n^{2.5})$ fault identification algorithm for diagnosable systems. IEEE Transactions on Computers, 1984, 33(6): 486-492

[5] Deng Wei, Yang Xiao-Fan, Wu Zhong-Fu. An efficient diagnosis algorithm of BC graphs. Acta Electronica Sinica, 2006, 34(4): 698-701(in Chinese)
(邓伟,杨小帆,吴中福. 一种高效的基于 BC 图的 t/k -诊断算法. 电子学报, 2006, 34(4): 698-701)

[6] Scheinerman E R. Almost sure fault tolerance in random graphs. Society for Industrial and Applied Mathematics (SIAM) Journal on Computing, 1987, 16(6): 1124-1134

[7] Blough D, Sullivan G, Masson G M. Efficient diagnosis of multi-processor system under probabilistic models. IEEE Transactions on Computers, 1992, 41(9): 1126-1136

[8] Yang X F, Chen T H et al. A new scheme for the fault diagnosis of multiprocessor systems//Proceedings of the 5th Asian Test Symposium (ATS'96). National Tsing Hua University Hsinchu, Taiwan, China, 1996. Washington DC: IEEE Computer Society, 1996: 286-294

[9] Barborak M, Malek M, Dahbura A. The consensus problem in fault-tolerant computing. ACM Computing Surveys, 1993, 25(2): 171-220

[10] Dabura A T. System-level diagnosis: A perspective for the third decade//Tewksburg S et al eds. Concurrent Computations: Algorithms, Architectures and Technology. New York: Plenum Press, 1988

[11] Back T, Hoffmeister F, Schwefel H P. Application of evolutionary algorithms. University of Dortmund; The Systems Analysis Research Group, Department of Computer Science, Technical Report SYS-2/92, 1993

[12] Holland J H. Adaptation in Natural and Artificial System. Michigan: University of Michigan Press, 1975

[13] Goldberg D E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989

[14] Elhadeif M, Ayeb B. An evolutionary algorithm for identifying faults in *t*-diagnosable systems//Proceedings of the 19th

IEEE Symposium on Reliable Distributed Systems. Nurnberg, Germany. Washington, DC: IEEE Computer Society, 2000; 74-83

[15] Elhadeif M, Ayeb B. An evolutionary algorithm for generalized comparison-based self-diagnosis of multiprocessor systems. Applied Artificial Intelligence, 2002, 16(1): 73-95

[16] Elhadeif M, Abrougui K, Das S, Nayak A. A parallel probabilistic system-level fault diagnosis approach for large multiprocessor systems. Parallel Processing Letters, 2006, 16(1): 63-80



DENG Wei, born in 1975, Ph. D. , lecturer. His research interests include fault tolerance, grid computing, etc.

YANG Xiao-Fan, born in 1964, professor, Ph. D. supervisor. His current research interests include system-level fault diagnosis, fault-tolerant computing, difference equations, neural networks and biological computing, etc.

WU Zhong-Fu, born in 1938, professor, Ph. D. supervisor. His research interests are distance education, grid computing and network security.

Background

The system-level diagnosis of multicomputers, which is intended for identifying faulty processors by conducting tests on processors and interpreting the test outcomes, is an important topic of research from fault-tolerant computing. The central task of system-level diagnosis is to develop efficient diagnosis algorithms. It is known that some typical fault identification problems are NP-hard. As a result, various heuristic diagnosis algorithms have been proposed.

This paper presents a new diagnosis algorithm by elaborately designing the genetic heuristics. Both theoretical analysis and simulation result show that the algorithm is superior

to a recently proposed diagnosis algorithm.

The work of this paper is a constituent part of a large project entitled "Parallel Computing and Fault Tolerance", which is supported by Program of Educational Ministry of China for New Century Excellent Talents (Grant No. NCET-05-0759), Doctorate Foundation of Educational Ministry of China (20050611001), and Natural Science Foundation of Chongqing CSTC (2006BB2231, 2005BB2191).

The research group has published more than 20 academic papers in distinguished international journals towards this direction of research.