

一类带有 n 中取连续 k 系统性质的飞机机载导航设备的可靠性结构优化

邢妙娟

(中国航空工业集团公司洛阳电光设备研究所, 洛阳 471009)

摘要 n 中取连续 k 系统的失效与否,不是单纯地取决于失效部件的个数,而是取决于相邻失效部件的个数。该类系统的可靠性对飞机机载导航设备的安全性具有重要的意义,所以如何分配这 n 个部件的位置才能使得系统结构的可靠性最大是该问题的关键所在。基于该问题,以最大化系统的可靠性为目标函数,利用遗传算法来求解该系统最优的部件排列。最后以四种典型系统为例,与传统的算法进行对比,验证该遗传算法的有效性和正确性。

关键词 n 中取连续 k 系统 可靠性 遗传算法 部件

中图分类号 V241.03; **文献标志码** A

飞机机载导航设备在飞机的安全性方面具有重要的作用,通常为了提高飞机的可靠性,很多设备需要设计成冗余系统。例如,在民航机载导航系统中,一般装有三套惯性导航设备、两套多普勒雷达、两套伏尔系统、两套无线电罗盘,以及气象雷达、罗兰系统、仪表着陆设备、无线电高度表等。这类系统为 n 中取 k 系统,即当 k 个设备都失效时,整个系统才失效。然而直列的连续 n 中取连续 $k:F(G)$ 系统为该系统中的典型系统,即系统由 n 个部件组成,并排成一列,当系统中至少有 k 个连续的部件失效(工作)时,系统才失效(工作)。这种系统的失效与否,不是单纯地取决于失效部件的个数,而是取决于相邻失效部件的个数。研究该类系统的可靠性对飞机的安全系数具有重要的意义。

针对 n 中取连续 $k:F(G)$ 系统,因为各个部件的可靠度不可能相同,所以根据系统的结构函数,将同一部件分配到系统中的不同位置,会使系统具有不同的可靠性。Hwang^[1] 给出了 n 中取连续 $k:F$ 系统的可靠性计算公式。Kuo, Zhang 和 Zuo^[2] 研究

了 n 中取连续 $k:F$ 系统和 G 系统之间的关系,并详细讨论该系统的可靠性的计算。如何分配这 n 个部件的位置才能使得系统的可靠性最大的问题被称为 n 中取连续 k 系统的最优化问题。针对该问题, Kontoleon^[3] 提出了一种迭代算法,将系统中的所有位置都初始化为可靠度最低的组件,然后将剩余组件中可靠度最大的组件分派到具有最大重要度的位置,直至所有的部件分配完毕。Lin 和 Kuo^[4] (LK) 通过分析计算误差项,提出了贪婪算法(LK 算法),该算法用可靠度最低的部件初始化所有位置,然后依次将可靠度最大的部件分派到具有最大重要度的所有位置。刘琴和孙林岩^[5] 以 LK 算法为基础提出了 LKS 算法,通过对一个典型系统的数值试验和分析,指出 LKS 算法对 LK 算法的优势,并将其扩展到一般系统。Zhu, Yao 和 Kuo^[6] 提出了 LK 算法的延伸算法,通过对 8 种系统进行数值试验和分析,得出了综合使用两种算法求解最优系统可靠性,并指出了新方法的使用条件。Yao^[7] 在 LK 算法的基础上提出了三种衍生的 LK 算法,并称原来的 LK 算法为 LKA 算法,三种衍生的算法分别为 LKB、LKC、LKD,简称为 LK 类型算法。

但是以上算法在解决部件较少的分配问题时,往往得出的系统可靠性的精度不是很准确,与标准系统的可靠性相差很大。针对飞机机载导航设备

2012年12月10日收到

作者简介:邢妙娟. E-mail: duihongyan@mail.nwpu.edu.cn.

而言,该系统中的冗余部件数量相对较少(低于 10),不适合运用以上的算法。遗传算法是模拟自然界中生物的遗传和进化过程而形成的一种自适应全局优化概率搜索算法,它可以打破这种局限性,获得全局最优解。本文将在以前的研究成果基础上基于遗传算法(based genetic algorithm, BGA 算法)来解决 n 中取连续 k 系统的最优化问题,并将 BGA 的计算结果与 LK 类型算法的计算结果相比较,分析这五种算法的优劣和使用范围。

1 n 中取连续 k 系统可靠性

直列的 n 中取连续 $k:F$ 系统的可靠性的计算公式如式(1)。

$$h[p(n), k]_F = \sum_{i=n-k+1}^n p_i \left(\prod_{j=i+1}^n q_j \right) h[p(i-1), k]_F \quad (1)$$

式(1)中 $p_i (i = 1, 2, \dots, n)$ 表示按一定顺序排列的部件 i 的可靠性, $h[p(n), k]_F$ 表示在给定的 p_i 下直列的 n 中取连续 $k:F$ 系统的可靠性, $q_i = 1 - p_i$ 表示部件 i 失效的概率。并且式(1)有以下条件:

$$h[p(u), v]_F = 1 (u < v); \prod_{i=1}^n q_i = 1 (i < n) \quad (2)$$

对于直列的 n 中取连续 $k:G$ 系统的可靠性,根据文献^[2]的方法,只要将式(1)中的 p_i 和 q_i 交换位置并用 1 减去所得到的结果即可得到系统的可靠性,即:

$$1 - h[p(n), k]_G = \sum_{i=n-k+1}^n q_i \left(\prod_{j=i+1}^n p_j \right) h[p(i-1), k]_G \quad (3)$$

式(3)中 $h[p(n), k]_G$ 代表直列的 n 中取连续 $k:G$ 系统的可靠性。

2 基于遗传算法的可靠性结构优化

2.1 算法的提出

该算法的目标函数是计算 n 中取连续 k 系统的可靠性,如式(1)和式(3)所示。不同的部件排列使模型具有不同的可靠性。因此,计算 n 中取连续 k 系统的可靠性时,首先要确定它的可靠性函数。其解空间是部件不同排列的集合,若一个系统具有 n 个部件,则解空间的大小为 $n!$ 。下面叙述基于遗传

算法用于解决 n 中取连续 k 系统最优可靠性的的算法(Based Genetic-Algorithm, BAG)。

2.1.1 编码方法

我们用 $N = \{1, 2, \dots, n\}$ 表示 n 个位置的向量,一个部件被指派到位置 $i (i \in N)$,用 π_i 表示。这 n 个部件都被分派到系统中时,会形成一个排列,用 $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ 表示其中的一个排列, $\pi(i, j)$ 表示通过在一个排列 π 中交换位置 i 和 j 处的部件形成的一个新排列。根据模型的性质, BGA 算法采用符号编码方法。在建立模型时,已经对 n 个部件按可靠性的顺序进行了编号,那么每一个部件的编号对应一个编码符号。部件的一个分派 π 被编码为染色体,每一个整数 $\pi_i (i = 1, 2, \dots, n)$ (即部件编号)在染色体中是一个基因。每个种群包含 m 个染色体,且在遗传算法中种群的规模 m 是常数,初始群体包含 m 个随机产生的染色体,保证了种群的多样性。遗传算法结束后输出的最优染色体就是该模型的最优分派。

2.1.2 适应度尺度变换

模型的目标是使系统的可靠性最大,而部件的每一种分派 π 可能使系统具有不同的可靠性。因此, BGA 使用每条染色体的可靠性大小在种群中所占的比例 $f_t (t = 1, 2, \dots, m)$ 作为染色体的适应度。同时,染色体 π_t 的适应度变换尺度,定义为 F_t ,可用带有参数 a 和 b 的线性变换函数表示:

$$F_t = af_t + b \quad (4)$$

设 $f_{\max}, f_{\min}, f_{\text{avg}}$ 分别表示变换之前 m 个染色体的最大、最小、平均适应度值, $F_{\max}, F_{\min}, F_{\text{avg}}$ 分别表示变换之后 m 个染色体的最大、最小、平均适应度值。参数 a 和 b 满足条件:

$$a = \frac{(S_f - 1)f_{\text{avg}}}{(f_{\text{avg}} - f_{\min})S_f + (f_{\max} - f_{\text{avg}})}; b = (1 - a)f_{\text{avg}} \quad (5)$$

式(5)中 $S_f > 1$ 为变换因子, $f_{\text{avg}} = F_{\text{avg}}, F_{\max} = S_f f_{\min}$, 且 a 始终大于 1, b 的值不确定。

2.1.3 终止条件

种群的遗传代数不能超过事先制定的临界值(如 200 代)。当

$$(f_{\text{avg}} - f_{\min}) \times S_f + (f_{\max} - f_{\text{avg}}) < \alpha \quad (6)$$

这里 α 是一个很小的正数, BGA 就终止执行。

2.1.4 选择操作

从父代染色体中选择 m 个染色体, 组成 $m/2$ 对父染色体。本文使用比例选择法, 则一个染色体被选作父染色体的概率是: $F_i / (F_1 + F_2 + \dots + F_m)$; $t = 1, 2, \dots, m$ 。

2.1.5 交叉操作

这里 BGA 使用三点交叉在一对父染色体上执行, 假设以概率 p 进行交叉, 这三个交叉点在一对父染色体之间产生一个匹配部分, 并在交叉点执行点到点的交叉互换工作。例如一对父染色体 π_1, π_2 :

$$\pi_1 = 4 \mid 3 \ 2 \mid 5 \ 6 \mid 1; \quad \pi_2 = 2 \mid 5 \ 4 \mid 1 \ 3 \mid 6。$$

以染色体 π_1 为例, 三个随机交叉点为 3, 2 和 4。则基因 3 和 5, 2 和 4, 1 和 6 交换位置, 可以产生的一个子代染色体: $\pi_1' = 2 \ 5 \ 4 \ 3 \ 1 \ 6$ 。

2.1.6 变异操作

在交叉操作后, 假设以概率 p' 在每条染色体上执行变异操作, 产生子代种群 $P(t+1)$, 变异操作随机选择两个位置并交换这两个位置的基因。以上述的交叉结果 π_1' 作为变异的染色体, 且第 2 个位置和第 6 个位置被随机选择为变异位置, 那么这两个位置的基因 5 和 6 交换位置, 产生的子代染色体: $\pi = 2 \ 6 \ 4 \ 3 \ 1 \ 5$ 。

2.1.7 最优保存策略

首先计算变异操作后的子代种群 $P(t+1)$ 中每个个体的适应度, 然后用最优染色体替换子代种群 $P(t+1)$ 中适应度最小的染色体, 完成种群的重组, 形成新的子代种群 $P(t+1)$ 。用父代的一条最优染色体替换子代的一条最差染色体, 这样可以将产生的最优染色体直接复制到子代, 还有利于保持种群的多样性, 有利于寻找全局最优解。

则 BAG 算法的具体步骤如下:

(1) 产生一个具有 m 个个体的初始种群, 这里 m 个个体就是部件的 m 种排列;

(2) 如果当前种群满足终止条件, 停止程序, 输出最优解和最优分派;

(3) 根据式(4)计算每条染色体的适应度变换尺度;

(4) 执行最优保存策略, 选出当前种群的一条最优染色体;

(5) 在当前种群中执行选择、交叉、变异操作, 产生 m 个子代染色体, 并用当前种群的最优染色体替换子代染色体中的一条最差染色体;

(6) 把下一代种群作为当前种群, 返回到步骤 2。

2.2 可靠性结构优化

假设某类飞机机载导航设备包含了一个直列的 n 中取连续 $k:G$ 系统。下面我们分析该系统可靠性最大的最优部件排列。随着飞机寿命的增加, 部件的可靠性会越来越低, 所以这里我们设定三种不同类型的部件, 它们的可靠性分别在区间 $[0.8, 0.99]$ (类型 1)、 $[0.01, 0.2]$ (类型 2)、 $[0.01, 0.99]$ (类型 3) 上, 其中类型 1 代表高可靠性的部件, 一般出现在飞机寿命的初期阶段; 类型 2 代表低可靠性的部件, 一般出现在飞机寿命的末期阶段; 类型 3 代表任意可靠性的部件, 一般出现在飞机寿命的中期阶段。本文将以 $G1:7$ 中取连续 $2:G$ 系统、 $G2:8$ 中取连续 $2:G$ 系统、 $G3:7$ 中取连续 $3:G$ 系统、 $G4:8$ 中取连续 $3:G$ 系统为例, 分别利用文献 [7] 提到的 LKA、LKB、LKC、LKD 算法和本文的 BGA 算法对这四个系统进行结构优化, 并对得出的系统可靠性结果进行比较分析。

为了衡量 BGA 解决直列的 n 中取连续 $k:G$ 系统模型的精度, 本文利用标准系统可靠性 (standardized system reliability, SSR):

$$SSR = \frac{R_{\text{heu}} - R_{\text{wor}}}{R_{\text{opt}} - R_{\text{wor}}} \quad (6)$$

式(6)中, R_{heu} 表示通过 BGA 求得的系统可靠性, R_{opt} 和 R_{wor} 分别表示通过枚举法得到的最优和最差系统可靠性。考虑到一次试验可能有偏差, 对于一个直列的 n 中取连续 $k:G$ 系统模型, 本文使用 100 个实例, 每一个实例中部件的可靠性随机产生, 即随机产生 100 组部件可靠性。每个模型需要在三种类型的部件下试验, 则每个模型需要 $3 \times 100 = 300$ 个实例。对于一个模型在一种类型部件下的试验, 可得到 100 个 SSR, 我们使用平均标准系统可靠性 (mean standardized system reliability, MSSR) 评价这些数据, 并称这

样的数值试验为一个试验。同时,本文使用标准差 σ 衡量一个试验中 100 个数据的离散程度,使用 SSR 的最大值和最小值作为辅助评价数据。

在该五种算法下,三类部件的计算结果如表 1 所示。从表中可以看出,这五种算法根据不同的部件类型都能找到系统可靠性的最优解,这说明每种算法适合于不同的部件类型。从表 1 中的数据可知,每种算法的方差都比较小,接近于 0,这说明这五种算法算出的系统的可靠性波动都比较小。表 1 中的数据显示了,在类型 1 部件上 LKB 算法和 LKC 算法在 G3 和 G4 系统上的计算结果相同。从 MSSR 值的大小看,BGA 算法最适合于类型 3 部件,LKA 算法适用于类型 2 部件,LKB 算法适用于类型 1 部件。

表 1 五种算法对四种系统的计算结果

系统	算法	部件类型		
		1	2	3
G1	LKA	0.953 615/0.024 591	0.999 123/0.001 991	0.989 395/0.011 929
		0.898 703/1.000 000	0.986 884/1.000 000	0.946 774/1.000 000
	LKB	0.998 345/0.003 139	0.994 646/0.005 281	0.995 160/0.012 570
		0.983 830/1.000 000	0.979 782/1.000 000	0.914 566/1.000 000
	LKC	0.997 774/0.011 054	0.997 774/0.024 549	0.975 737/0.057 091
		0.910 238/1.000 000	0.911 378/1.000 000	0.626 220/1.000 000
	LKD	0.803 144/0.093 962	0.983 561/0.019 579	0.864 381/0.123 878
		0.526 632/0.978 583	0.912 482/1.000 000	0.148 105/0.999 914
	BGA	0.999 877/0.000445	0.998 976/0.001 875	0.999 687/0.000 682
		0.996 709/1.000 000	0.989 734/1.000 000	0.995 227/1.000 000
G2	LKA	0.946 315/0.020 731	0.999 147/0.001 725	0.989 926/0.010 876
		0.893 542/0.993 284	0.986 233/1.000 000	0.951 237/1.000 000
	LKB	0.999 952/0.000 190	0.988 737/0.011 648	0.994 990/0.008 925
		0.998 427/1.000 000	0.949 159/1.000 000	0.955 639/1.000 000
	LKC	0.973 469/0.016 893	0.998 991/0.004 368	0.985 659/0.028 840
		0.932 896/1.000 000	0.969 692/1.000 000	0.822 012/1.000 000
	LKD	0.767 054/0.065 828	0.973 469/0.016 893	0.980 820/0.014 292
		0.644 635/1.000 000	0.932 896/1.000 000	0.922 813/1.000 000
	BGA	0.994 573/0.007 226	0.984 307/0.014 397	0.998 054/0.003 147
		0.969 493/1.000 000	0.941 058/1.000 000	0.980 586/1.000 000
G3	LKA	0.991 888/0.007 073	0.996 011/0.005 447	0.996 148/0.005 447
		0.969 847/1.000 000	0.968 283/1.000 000	0.968 283/1.000 000
	LKB	0.999 976/0.000 243	0.997 745/0.003 667	0.998 195/0.003 833
		0.997 557/1.000 000	0.982 911/1.000 000	0.982 484/1.000 000
	LKC	0.999 976/0.000 243	0.983 885/0.018 056	0.998 195/0.003 833
		0.997 557/1.000 000	0.926 514/1.000 000	0.982 484/1.000 000
	LKD	0.951 872/0.042 136	0.989 152/0.014 369	0.904 745/0.084 383
		0.764 659/0.999 789	0.926 514/1.000 000	0.616 713/1.000 000
	BGA	0.994 607/0.013 165	0.999 132/0.001 873	0.998 686/0.002 590
		0.918 936/1.000 000	0.989 299/1.000 000	0.983 358/1.000 000
G4	LKA	0.981 643/0.010 446	0.994 267/0.007 465	0.994 189/0.006 258
		0.952 756/0.999 165	0.964 532/1.000 000	0.972 408/1.000 000
	LKB	0.999 987/0.000 118	0.994 384/0.007 552	0.996 837/0.005 694
		0.998 818/1.000 000	0.961 494/1.000 000	0.964 804/1.000 000
	LKC	0.999 987/0.000 118	0.982 339/0.022 731	0.994 582/0.019 146
		0.998 818/1.000 000	0.891 179/1.000 000	0.842 415/1.000 000
	LKD	0.884 944/0.102 114	0.997 399/0.002 970	0.832 687/0.153 096
		0.570 524/0.999 583	0.984 679/1.000 000	0.233 088/1.000 000
	BGA	0.988 915/0.012 983	0.985 014/0.017 061	0.996 482/0.005 976
		0.933 291/1.000 000	0.912 241/1.000 000	0.984 679/1.000 000

注:表格中的每组数据,第一行的第一个数表示 MSSR,第二个数表示 100 个 SSR 的标准差;第二行的第一个数据表示 100 个 SSR 的最小值,第二个数据表示其最大值。

为比较 MSSR 值即 100 组实例最优解的期望 μ 与方差 σ 之间的关系,我们计算其比值 σ/μ 。图 1 以部件类型为分组,描绘了各种算法下 σ/μ 的值,其中横坐标表示不同的四种系统,纵坐标表示 σ/μ 的值。从图 1(a)和(c)中可以很容易地看出,在类型 1 和类型 3 部件中,LKD 算法的值较大,其余算法的值都很小,这说明 LKD 算法得出的系统可靠性的期望最差。在图 1(b)中,针对类型 2 部件,LKC 算法的值较大,其得出的系统可靠性的期望最差。在类型 2 部件中,所有的值都很小。通过表 1 可以看出,LKD 算法的计算结果,包括 MSSR 值和标准差,往往不如其它 4 种算法,这与图 1 相对应。从图 1 中的柱状图可知,每组实例的最优解的标准差与它们的平均值相比很小,最优解的波动性也很小。因此,在小型系统试验中,将不再考虑标准差,只比较 MSSR 值。

表 2 说明了在 $3 \times 4 = 12$ 个试验中,一种算法优于其它算法的次数。表中数据之和大于 12,这是因为在类型 1 部件上 LKB 算法和 LKC 算法在 G3 和 G4 系统上的计算结果相同,且这些结果高于其他算法得到的结果。从表 2 中还能得知,BGA 算法最适合于类型 3 部件,LKA 算法适用于类型 2 部件,LKB 算法适用于类型 1 部件。虽然在类型 1 和类型 2 部件上 BGA 算法还不如 LKB 算法和 LKA 算法,但在类型 2 和类型 3 部件上却优于这两种算法,这与表 1 中的数据相对应。总体看来,BGA 算法优于其它算法,其次是 LKB 算法,然后是 LKA 算法和 LKC 算法,LKD 算法最差。

表 2 五种算法性能比较

部件类型	算法				
	LKA	LKB	LKC	LKD	BGA
类型 1	0	3	2	0	1
类型 2	2	0	0	1	1
类型 3	0	1	0	0	3
合计	2	4	2	0	5

3 结论

本文的 12 个试验都在 MATLAB7.11 上进行,计算机的配置为 Intel(R) Core(TM)2 的 3GHz 处理

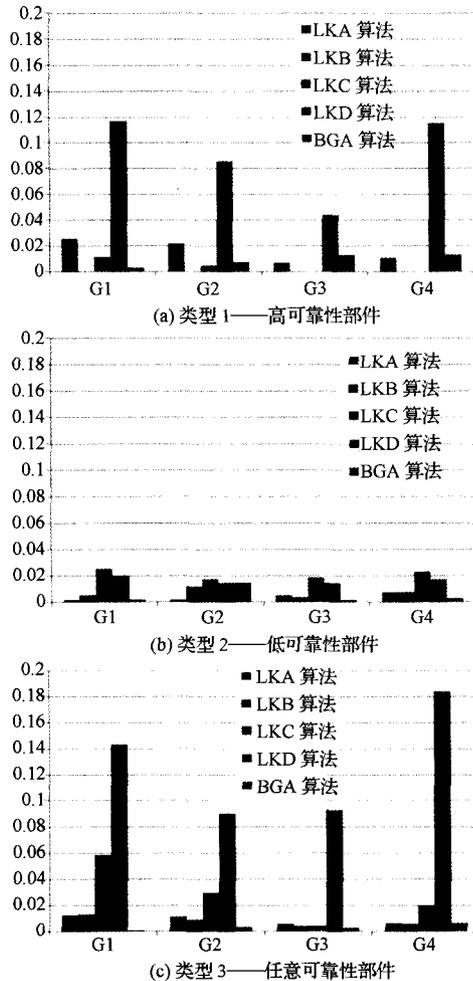


图1 三种类型部件下不同算法的 σ/μ 的值

器,2GB内存。我们得出了BGA算法最适合于类型3部件,LKA算法适用于类型2部件,LKB算法适用于类型1部件。虽然在类型1和类型2部件上BGA算法还不如LKB算法和LKA算法,但在类型2和类型3部件上却优于这两种算法。但是,BGA算法的计算速度在该计算机上的运行速度为100 min,LK类型算法为60分钟,从速度上看,BGA算法明显低于LK类型算法,但是与枚举法相比(随着 n 的增加而增加),已经快了很多,这就为将重要度引入遗传算法来提高模型求解的计算速度提供了基础,这也是我们下一步要展开的工作。

参考文献

- 1 Hwang F K. Fast solution for consecutive- k -out-of- n : F system. IEEE Transactions on Reliability, 1982; 31(5): 447—448
- 2 Kuo W, Zhang W, Zuo M J. A consecutive- k -out-of- n : G system; the mirror image of a consecutive- k -out-of- n : F system. IEEE Transactions on Reliability, 1990; 39(2): 244—253
- 3 Kontoleon J M. Optimal link allocation of fixed topology networks. IEEE Transactions on Reliability, 1979; 28(2): 145—147
- 4 Lin F H, Kuo W. Reliability importance and invariant optimal allocation, Journal of Heuristics, 2002; 8(2): 155—171
- 5 刘琴, 孙林岩. 求解系统可靠性优化中指派问题的启发式算法. 运筹与管理, 2011; 20(6): 15—18
- 6 Zhu X, Yao Q, Kuo W. Birnbaum importance in solving component assignment problems. IIE Transaction, 2011; 43(5): 633—646
- 7 Yao Q. Birnbaum importance patterns and their application in the component assignment problem, Doctoral Dissertations, University of Tennessee, USA, 2011

Reliability Structural Optimization of Airborne Navigation Equipment with the Consecutive- k -out-of- n System

XING Miao-juan

(China Aviation Industry Corp Luoyang Electro-optic Equipment Research Institute, Luoyang 471009, P. R. China)

[Abstract] Whether the consecutive- k -out-of- n system can function properly depends on not only the number of the failed components, but also the number of adjacent failed components. The importance of such systems is significant for the safety of airborne navigation system. Therefore the key issue is that how to arrange the positions of n components to obtain the maximum reliability of system structure. With aiming at maximizing the reliability of the system, the genetic algorithm is used to obtain the optimum component arrangement. In the end, compared with the traditional algorithms, the algorithm's effectiveness and accuracy is verified, with 4 typical systems as examples.

[Key words] consecutive- k -out-of- n system reliability genetic algorithm component