第6卷 第1期 自 动 化 学 报 Vol. 6, No. 1
1980年1月 ACTA AUTOMATICA SINICA Jan., 1980

用优选法解系统可靠性最优化问题的一个简捷算法*

廖炯生
(北京控制工程研究所)

摘 要

冗余系统可靠性最优化问题在本质上属于非线性整数规划. 现有的严格解法和近似解法要求浩繁的计算,因而较简单的直接寻查法引起了重视. 本文提出,用优选法解可靠性最优化问题是值得探讨的;在一定范围内灵活掌握工程性约束条件,有可能简化可靠性最优化问题的求解,具有实际意义.

本文采用优选法的分批试验法,对文献[8]的算法作了进一步的简化,从而提出一个更简捷的算法. 文中举例说明这种算法的应用,并和文献结果进行了比较.

一、引 言

提高系统可靠性有二种最基本的办法,其一是改善每一元件的可靠性,其二是采用冗余技术(热贮备,即可靠性并联;冷贮备;n中取k的表决方式等).

用低可靠元器件组成高可靠系统的想法已经提出来二十多年了. 通过引入冗余单元(元器件或分系统),可以使得在有些单元发生故障时,整个系统还能够正常工作,或者基本上正常工作,实现所谓"故障掩蔽"和"故障软化"的作用. 随着数字逻辑电路和计算技术的发展,这种故障相容技术(容错技术)进而发展到对故障的自动诊断、隔离以至修复. 有关这方面的情况,文献[1]作了综述.

另一方面,在采用冗余单元来提高系统可靠性时,不可避免地要受到价格、体积、重量、功率等资源条件的实际限制. 如何在一组线性或非线性约束条件下,确定冗余单元数目和位置,使得系统可靠性达最大可能值,这就是最佳冗余分配,或者更一般地称为系统可靠性最优化问题.

五十年代以来,最优化理论和方法逐渐得到丰富和发展,现已成为应用数学的一个重要分支,被广泛应用于各个技术部门. 在国外,人们纷纷采用最优化理论的各种数学方法,尝试解决系统可靠性最优化问题. 但是至今只取得了有限的成功.

可靠性最优化问题在本质上属于非线性整数规划. 它比一般非线性规划问题更困难,因为冗余单元数目只能取正整数,要求的是整数最优解.

文献[2,3]对冗余系统可靠性最优化技术作了综述. 文中列举的各种方法可分为三类:

* 本文于1979年6月15日收到.

第一类是严格解法．其中动态规划法[4]一般只用于约束条件数目小于三的情况．整数规划法或者是首先把非线性目标函数和约束条件取折线近似，再用线性整数规划方法（例如“割平面法”）求解；或者转换成具有 0—1 型变量的整数规划问题，这使得变量和约束数目增加，而计算时间随之作指数增长；还有采用分枝限界法（Branch-and-Bound Technique），虽可不必转换为 0—1 变量，但这些方法都需要进行浩繁的计算．即使是用计算机来算，也还是繁．

第二类是近似解法，如：无约束极小化方法、单纯形模式搜索法等．一般只能得到非整数解，最后还要“四舍五入”取整数，当然就无法保证最优解．而且计算还是繁．

第三类是直接寻查法（Heuristic Methods），是近似解法中比较简单的一类．近几年来受到很大重视．如综述 [2] 的作者后来就专门补充综述了六种直接寻查法[5]．

在我国，七十年代以来，经华罗庚教授提倡，优选法得到推广应用[6]．我们认为，用优选法（就是直接最优化方法）解系统可靠性最优化问题，有可能用简单的办法解决上述难算的问题，因而是一个值得探索的方向．

美国 IEEE 可靠性汇刊主编 R. A. Evans 在 1976 年曾经指出[7]：可靠性最优化问题中约束条件有二种不同的性质，其一是纯数学的，如概率必须在闭区间 [0, 1] 上；其二是工程性的约束，如系统的最高价格和重量等．前者的界限是绝对的（我们还可以补充说，冗余单元数目必须取正整数，这一条物理约束也是绝对的），但工程性约束的界限则被人为地绝对化了．这样做的目的本来是为了形成数学模型、简化问题．实际适得其反，人们自己给自己造成了求系统可靠性最优化问题的严格最优整数解这样一个很大的数学困难．因此，我们认为，在一定范围内灵活掌握工程性约束条件，对于简化可靠性最优化问题的求解，具有实际意义．在理论上，最优化问题的约束条件是否应当有一定程度的模糊，也是可以探讨的．

根据上述观点，我们采用优选法的分批试验法，对现有最简单的一种直接寻查法（Sharma 等提出的算法[8]），作了进一步的简化，从而提出一个更简捷的算法．从优选法的观点看来，Sharma 算法属于“单点”试验性质，我们的算法属于“多点”试验性质，所以效率高得多．系统越大，这种繁简程度的差别也越大．

二、算 法

J. Sharma 和 K. V. Venkateswaran 在 1971 年提出使系统可靠性取极大值的一个直接算法．这是现有多种直接寻查法中最简单的一种．它基于这样的直观想法：每次在可靠性串联系统中把不可靠性最大的一级并联上一只冗余元件，并检验约束条件．在不超出约束条件的范围内，继续这样做．通过这样的一系列步骤，可以达到使系统可靠性接近最优值的目的．这个方法不能保证数学上严格的最优解，但非常简便，易于排成计算机程序，便于工程应用．

由于这个算法最简单，也由于严格的和其它近似的解法计算浩繁，所以这个算法引起了广泛的注意和讨论．有趣的是所有对 Sharma 算法的改进意见[9-14]都是使它复杂化．能不能在继续简化的方向再前进呢？

我们下面采用优选法的分批试验法，来进一步简化问题的解法，从而提出一个更简捷的算法.

设一系统有 k 级串联. 其中第 i 级有 n_i 个并联冗余元件，各自的失效概率（不可靠性）同为 q_i. 所有各级和所有元件是统计独立的.

因为 k 级可靠性串联系统是保证完成一定的物理功能所必需的，所以 $n_i \geqslant 1$,（$i = 1, 2, \cdots, k$）.

整个系统的不可靠性为

$$Q(n) = 1 - \prod_{i=1}^{k} (1 - q_i^{n_i}), \tag{1}$$

n 为非负整数向量，$n \equiv (n_1, n_2, \cdots, n_k)$，它代表各级元件数的集.

约束条件为

$$\sum_{i=1}^{k} G_{ji}(n) \leqslant B_j, \quad j = 1, 2, \cdots, r. \tag{2}$$

问题是在约束条件（2）之下，选 n 使 $Q(n)$ 为最小.

若 $q_i \ll 1$，则

$$Q(n) \approx q(n) \equiv \sum_{i=1}^{k} q_i^{n_i} \tag{3}$$

此时问题简化为在约束（2）之下选 n 使 $q(n)$ 为最小.

从优选法观点看来，这个问题类似于在 $Q_i \equiv q_i^{n_i}$ 的一定取值范围内做试验，以检验是否达到了约束（2）中任一条件；如未达到，可取更小的 Q_i 值（相应更大的 n_i 值）再试. 这样继续进行到达到约束（2）为止. Q_i 越小越好. Sharma 算法相当于"单点试验"，在系统较大时"试验步数"还是较多的. 我们现在用分批法作"多点试验"来提高效率. 在这里目标函数是单调的，当 n_i 增大时 $q_i^{n_i}$ 单调减小，所以我们可以用平分法来分批.

同文献 [8] 一样，我们从所有 $n_i = 1$ 的情形出发. 首先对于

$$Q_i \geqslant \frac{1}{2} q_{i\cdot\max}$$

的各级各加一个并联元件（$q_{i\cdot\max}$ 为元件不可靠性之最大者），并检验约束条件（2）. 在约束许可范围内持续进行这种步骤，每一步都把上次的 Q_i 界限平分作为新的界限. 正好达到约束（2）的任一条件时即停止. 如进行到第 m^* 步时超过（2）的任一条件，则把第 $m^* - 1$ 步和第 $m^* - 2$ 步之间的 Q_i 界限平分，并把第 $m^* - 1$ 步中 Q_i 低于此界限的各级各取去一只并联元件，重新检验约束（2）. 这样来回调试，直到求得最优解.

详细的步骤如下：

第一程　对于所有 i，取 $n_i = 1$. 这是一个 k 级串联系统，每级至少必有一个元件，它应不超出约束（2）.

第二程　在第 m 步，对所有 $Q_i \geqslant \frac{1}{2^m} q_{i\cdot\max}$ 的级各加一只并联元件，得到第 $m + 1$ 步各级的元件数.（$m = 1, 2, 3, \cdots$）

检验约束条件（2）. 只要尚未达到（2）的任一界限，就重复进行这一步骤. 如果：

A. 正好达到（2）的任一限度，则停止. 此时的 n_i 即最佳冗余元件数.

B. 如进行到第 m^* 步时超出（2）中任一条件，则按第三程进行.

第三程　将第 m^*-1 步中

$$Q_i < \left(2-\frac{1}{2^s}\right)\left(\frac{1}{2^{m^*-1}}\right) q_{i\cdot\max}$$

的各级各取去一只并联元件，s 为按本程连续进行的步数，$s=1, 2, 3, \cdots$.

检验约束 (2)，只要还是超出任一约束，就继续按本步骤进行．如果：

A．正好达到约束 (2) 的任一条件，则停止，此时有最优解；

B．如按本程进行 s^* 步之后已达不到 (2) 的任一限度，则转回第二程进行，将

$$\left(2-\frac{1}{2^{s^*}}\right)\left(\frac{1}{2^{m^*-1}}\right) q_{i\cdot\max}$$

逐次平分作为新的 Q_i 界限；

C．如果"增之一分则太长，减之一分则太短"，即在某一级单独加上一只并联元件时超出约束 (2)，而取去这一只元件时又达不到约束界限，则按取去这一只元件处理，并且在以后步骤中不再考虑这一级．

整个程序进行到正好达到约束 (2) 的任一条件为止，或进行到没有任一级需要继续考虑为止．所得结果即最优解．

三、举　　例

例一　五级系统如表 1 所示：

表　1

级　数	1	2	3	4	5
元件可靠性	0.9	0.75	0.65	0.8	0.85
元件价格	5	4	9	7	7
元件重量	8	9	6	7	8
约束条件　总价格 ≤132；　总重量 ≤142					

其解列于下表 2：

表　2

n_i					$Q_i\equiv q_i^{n_i}$					总重	总价	Q_i 界限
n_1	n_2	n_3	n_4	n_5	1	2	3	4	5			
1	1	1	1	1	0.1	0.25	0.35	0.20	0.15	38	32	$\geqslant\frac{1}{2}q_{i\max}$
	↓	↓	↓									
1	2	2	2	1	0.1	0.0625	0.1225	0.04	0.15	60	52	$\geqslant\frac{1}{4}q_{i\max}$
↓		↓		↓								
2	2	3	2	2	0.01	0.0625	0.042875	0.04	0.0225	82	73	$\geqslant\frac{1}{8}q_{i\max}$
	↓											
2	3	3	2	2	0.01	0.015625	0.042875	0.04	0.0225	91	77	$\geqslant\frac{1}{16}q_{i\max}$
		↓	↓	↓								
2	3	4	3	3	0.01	0.015625	0.015006	0.008	0.003375	112	100	$\geqslant\frac{1}{32}q_{i\max}$
	↓	↓										
2	4	5	3	3	0.01	0.003906	0.005252	0.008	0.003375	127	113	$\geqslant\frac{1}{64}q_{i\max}$
↓			↓									
3	4	5	4	3	0.001	0.003906	0.005252	0.0016	0.003375	142	125	

最佳冗余元件数 $n=(3, 4, 5, 4, 3)$.

系统可靠性为 0.985.

与文献 [8, 4, 14] 的结果相同.

例二 三级串联系统，第一级可在四个不同可靠性水平的元件中选用一个；第二级可采用并联冗余元件；第三级采取 (x_3+1) 个元件表决取 2 的模式工作，即 x_3+1 个元件中有 2 个或 2 个以上元件正常时该级正常. 现要求在三个非线性约束条件下选 $n=(x_1, x_2, x_3)$ 使系统可靠性为最大值[13].

约束条件：

$$G_1=4\exp\left\{\frac{0.02}{1-R_1(x_1)}\right\}+5x_2+2(x_3+1)\leqslant 45,$$

$$G_2=5+\exp\left(\frac{x_1}{8}\right)+3(x_2+e^{x_2/4})+5(x_3+1+e^{x_3/4})\leqslant 70,$$

$$G_3=10+8x_2\cdot e^{x_2/4}+6x_3\cdot e^{x_3/4}\leqslant 240.$$

系统可靠性：

$$R_s=\prod_{i=1}^{3}R_i(x_i),$$

其中

$$R_1(x_1)\equiv 0.88, 0.92, 0.98, 0.99, \text{当 } x_1=1, 2, 3, 4;$$

$$R_2(x_2)\equiv 1-(1-0.81)^{x_2};$$

$$R_3(x_3)\equiv\sum_{K=2}^{x_3+1}\binom{x_3+1}{K}\times 0.77^K\times(1-0.77)^{x_3+1-K}.$$

据此可预先算出 R_i 值如下表 3：

表 3

x_i	1	2	3	4	5	6
$R_1(x_1)$	0.88	0.92	0.98	0.99	—	—
$R_2(x_2)$	0.81	0.9639	0.993141	0.998697	0.999752	—
$R_3(x_3)$	0.5929	0.865634	0.959727	0.988583	0.996877	0.999166

例二的解列于下表 4.

最优解为 (3, 3, 5),

系统可靠性为 0.970018,

与文献 [13] 的结果相同.

四、讨 论

例一是线性约束条件下的可靠性最优化问题，文献 [4] 在 1958 年用动态规划解例一问题用计算机花五分钟时间. 文献 [8] 的算法比它简单得多，解例一问题只用 15 步计算. 而本文算法进一步简化成七步计算. 在系统很大时，这种繁简程度的差别更大.

例二是在非线性约束条件下，综合采用并联冗余、表决式冗余，以及选择不同可靠性水平的元件的办法时，求解可靠性最优化问题. 用本文算法需 14 步计算. 而用文献[13

表 4

x_i			$Q_i = 1 - R_i(x_i)$			G_1	G_2	G_3	Q_i 界限
x_1	x_2	x_3	$1 - R_1$	$1 - R_2$	$1 - R_3$				
1	1	1	0.12	0.19	0.4071	11.73	29.41	27.98	⩾0.20355
1	1	2	0.12	0.19	0.134366	13.73	36.23	40.06	⩾0.101775
2	2	3	0.08	0.0361	0.040273	23.14	47.82	74.49	⩾0.050888
3	2	3	0.02	0.0361	0.040273	28.87	47.99	74.49	⩾0.025444
3	3	4	0.02	0.006859	0.011417	35.87	60.40	126.04	⩾0.012722
4	3	4	0.01	0.006859	0.011417	**54.56**	60.59	126.04	<0.019083
4	3	4			同	上			<0.022264
3	3	4	0.02	0.006859	0.011417	35.87	60.40	126.04	⩾0.011132
3	3	5	0.02	0.006859	0.003123	37.87	69.26	165.48	⩾0.005566
3	4	5	0.02	0.001303	0.003123	42.87	**74.06**	201.65	<0.008349
3	3	5	0.02	0.006859	0.003123	37.87	69.26	165.48	⩾0.004175
3	3	5			同	上			⩾0.002088
3	3	6	0.02	0.006859	0.000834	39.87	**79.22**	222.16	<0.003132
3	3	5	0.02	0.006859	0.003123	37.87	69.26	165.48	

的算法，需计算目标函数和约束条件之间的“平衡系数”α 取 14 个值的情形，仅以 $\alpha = 0.5$ 的情形就要计算 9 步．最后还要比较 14 个 α 值之下的计算结果去选取最优解．显然本文算法更为简单有效得多．

从本文例子来看，例一正好达到一个约束条件，例二则不是“正好”的情况．假设我们认为达到约束条件中任一界限的 ±5% 的范围内即可认为合格，那么在本文例二中，只要算九步，可以省去五步计算，而不影响结果．当然，把约束界限放宽 ±5% 有时会使结果有不大的变化(好些或差些)，这在工程上是可以允许的．它的效果是可以把求最优解的问题大大简化．

参 考 文 献

[1] 含英，利用故障相容技术提高空间电子设备的可靠性，国外空间控制技术 1977 年第四期，1—19.
[2] F. A. Tillman, C. L. Hwang, W. Kuo, Optimization Techniques for System Reliability with Redundancy, *A Revitw, Trans. IEEE,* R-**26** (1977), No. 3, 148—155.
[3] K. B. Misra, On Optimal Reliability Design: A Review, Proc. IFAC 6th Congress, pt. III, 1975.
[4] R. Bellman, S. Dreyfus, Dynamic Programming and the Reliability of Multicomponent Devices. *Operations Research,* **6** (1958), No. 2, 200—206.
[5] W. Kuo, C. L. Hwang, F. A. Tillman, A Note on Heuristic Methods in Optimal System Reliability, *Trans. IEEE,* R-**27** (1978), No. 5, 320—324.
[6] 华罗庚，优选法平话及其补充，国防工业出版社(1971).
[7] R. A. Evans, Optimization Constraints, *Trans. IEEE,* R-**25** (1976), No. 2, 65.
[8] J. Sharma, K. V. Venkateswaran, A Direct Method for Maximizing the System Reliability, Ibid, R-**20** (1971), No. 4, 256—259.
[9] K. K. Aggawal et al., A New Heuristic Criterion for Solving a Redundancy Optimization

Problem, Ibid, R-24(1975), No. 2. 86—87.
[10] K. K. Aggarwal, Redundancy Optimization in General Systems, Ibid, R-25(1976), No. 5, 330—332.
[11] H. Sivaramakrishnan, A. D. Narasimhalu, Correction to "Redundancy Optimization in General Systems", Ibid, R-26(1977), No. 5, 345.
[12] K. Gopal, et al., An Improved Algorithm for Reliability Optimization, Ibid, R-27(1978), No. 5, 325—328.
[13] Y. Nakagawa, K. Nakashima, A Heuristic Method for Determining Optimal Reliability Allocation, Ibid, R-26(1977), No. 3, 156—161.
[14] A. D. Narasimhalu, H. Sivaramakrishnan, A Rapid Algorithm for Reliability Optimization of Parallel Redundent Systems, Ibid, R-27(1978), No. 4, 261—263.

A SIMPLIFIED HEURISTIC ALGORITHM FOR RELIABILITY OPTIMIZATION

LIAO JIONG-SHENG

(Beijing Institute of Control Engineering)

ABSTRACT

The optimization problem for system reliability with redundancy is essentially nonliear integer programming. For solving this kind of problems, the existing exact and approximate methods require a considerable amount of computation time. Consequently, simpler heuristic methods attract one's attention.

It is pointed in this paper that solving reliability optimization problems by direct optimization technique might be worth to be considered. By suitably relaxing the engineering constraints it is possible and practical to simplify the solution of reliability optimization problems.

Using the technique of batch test, the algorithm in [8] is further simplified, and a rapid heuristic algorithm for optimal redundancy allocation is presented. Applications to this algorithm were illustrated by two examples, and compared with other reference.