

基于人工免疫算法和蚁群算法求解旅行商问题

胡纯德 祝延军 高随祥

(中国科学院研究生院,北京 100039)

E-mail:huchunde@126.com

摘要 人工免疫算法具有快速随机的全局搜索能力,但对于系统中的反馈信息利用不足,往往做大量无谓的冗余迭代,求解效率低。蚁群算法具有分布式并行全局搜索能力,通过信息素的积累和更新收敛于最优路径上,但初期信息素匮乏,求解速度慢。该文提出一种基于人工免疫算法和蚁群算法的混合算法,采用人工免疫算法生成信息素分布,利用蚁群算法求优化解。将该算法用于求解旅行商问题进行计算机仿真,结果表明,该算法是一种收敛速度和寻优能力都比较好的优化方法。

关键词 人工免疫算法 蚁群算法 旅行商问题

文章编号 1002-8331-(2004)34-0060-04 文献标识码 A 中图分类号 TP301.6

A Hybrid Algorithm Based on Artificial Immune Algorithm and Ant Colony Algorithm for Solving Traveling Salesman Problem

Hu Chunde Zhu Yanjun Gao Suixiang

(Graduate School of the Chinese Academy of Sciences, Beijing 100039)

Abstract: Artificial Immune Algorithm has the ability of doing a global searching quickly and stochastically. But it can't make use of enough system output information, and hence has to do a large redundancy repeat searching for the optimal solution, which reduces the efficiency of algorithm. Ant Colony Algorithm converges on the optimal path through pheromone accumulation and renewal, and has the ability of parallel processing and global searching. But its convergence speed is slow because of poor pheromone on the path early. In this paper we propose a hybrid algorithm based on Artificial Immune Algorithm and Ant Colony Algorithm. It adopts Artificial Immune Algorithm to give pheromone to distribute and makes use of Ant Colony Algorithm to give the optimal solution. The computer simulation results show that the proposed algorithm is better than the previous two algorithms on the convergence speed and ability of searching for approximate global optimal solution for solving Traveling Salesman Problem.

Keywords: Artificial Immune Algorithm(AIA), Ant Colony Algorithm(ACA), Traveling Salesman Problem(TSP)

1 引言

旅行商问题(Traveling Salesman Problem, TSP)是具有重要意义的组合优化难题,已被证明属于 NPC 问题。它可简述为:给定一个边赋权图 $G=(V, E)$, 寻找 G 的 Hamilton 圈 C , 使得 C 的总权 $\omega(C)$ 最小。TSP 问题描述简单却难以求解,因而一直作为衡量各种优化算法性能的标准。近年来,人们从仿生学的机理中受到启发,提出了许多用于求解 TSP 问题的新方法,如:禁忌搜索算法、遗传算法、模拟退火算法、人工免疫算法和蚁群算法等。然而,面对 TSP 问题的复杂性,每种算法都表现出各自的优势和缺陷。

人工免疫算法(Artificial Immune Algorithm, AIA)是近几年才提出的一种随机优化方法,它模拟生物免疫系统,用亲和力来描述抗体与抗原之间的匹配程度,用排斥力来描述两个抗体之间的相似程度,依据抗体与抗原之间的亲和力以及抗体与抗体之间的排斥力来选择抗体。在用 AIA 求解优化问题时,一个抗体用一个字符串表示,满足约束条件的最优解即是抗原,

候选解即是抗体。抗体与抗原之间的亲和力反映了候选解与最优解的接近程度,也即反映候选解对目标函数和约束条件的满足程度;抗体与抗体之间的排斥力反映了不同候选解之间的异同,也即反映了抗体的多样性。保持抗体的多样性可以防止算法陷入局部最优解。AIA 具有快速随机的全局搜索能力,但对于系统中的反馈信息利用不足,当求解到一定范围时往往做大量无谓的冗余迭代,求解效率低。

蚁群算法(Ant Colony Algorithm, ACA)是由意大利学者 M.Dorigo, A.Colomi, V.Manizzo 等在 90 年代初提出的一类新的模拟进化算法,它通过信息素的积累和更新来寻求最优解。它的特点是模拟自然界中蚂蚁的群体行为。科学家们发现,蚂蚁有能力在没有任何提示下找到从巢穴到食物源的最短路径,并且能随环境的变化而变化,适应性地搜索新的路径,产生新的选择。经研究发现,其根本原因是蚂蚁在寻找食物源时,在其走过的路上释放一种特殊的分泌物—信息素,后来的蚂蚁选择该路径的概率与当时这条路径上该物质的强度成正比。当一定

基金项目:国家自然科学基金项目(编号:10171095);国家 863 计划重大专项(编号:2002AA103061)资助

作者简介:胡纯德(1973-),男,硕士研究生,主要研究方向为算法设计。祝延军(1973-),男,硕士研究生,主要研究方向为算法设计。高随祥(1962-),男,数学系教授,主要研究方向为组合最优化。

路径上通过的蚂蚁越来越多时,其留下的信息素轨迹也越来越多,后来的蚂蚁选择该路径的概率也越高,从而更增加了该路径的信息素强度。而强度大的信息素会吸引更多的蚂蚁,从而形成一种正反馈机制。通过这种正反馈机制,蚂蚁最终可以发现最短路径。特别是,当蚂蚁巢穴与食物源之间出现障碍物时,蚂蚁不仅可以绕过障碍物,而且通过蚁群信息素轨迹在不同路径上的变化,经过一定时间的正反馈,最终收敛到最短路径。ACA 具有分布式并行全局搜索能力,但由于初期信息素匮乏,致使求解速度缓慢。

该文提出的算法汲取了人工免疫算法和蚁群算法的优点,采用人工免疫算法生成信息素分布,利用蚁群算法求优化解,优势互补,在收敛速度和寻优能力两方面较原有算法都有明显改善。当用于求解 TSP 问题时,该算法在收敛速度上优于 ACA,在寻优能力上优于 AIA,是一种收敛速度和寻优能力都比较好的优化方法。

2 基于人工免疫算法和蚁群算法的混合算法

2.1 算法的设计思想

基于人工免疫算法和蚁群算法的混合算法,其基本思想是算法前过程采用 AIA,充分利用 AIA 的快速性、随机性、全局收敛性,寻找较优的可行解,算法后过程采用 ACA,利用前过程中 AIA 获得的较优可行解,产生初始信息素分布,然后充分利用 ACA 的并行性、正反馈性,提高求解效率。

2.2 算法中人工免疫算子的构造

人工免疫算子有字符换位算子、字符串移位算子、字符串逆转算子和优质字符串的保留等几种。对于 TSP 问题,AIA 的人工免疫算子的构造如下:

2.2.1 字符换位算子,可分为单对字符换位算子和多对字符换位算子

单对字符换位操作是随机取两个正整数 i, j ($1 < i, j \leq n, i \neq j$),以一定的概率 p_c 交换抗体 $A=(c_1, c_2, \dots, c_n)$ 中的一对字符 c_i, c_j 的位置;多对字符换位操作是预先确定一个正整数 u_c ,在抗体 $A=(c_1, c_2, \dots, c_n)$ 中随机取 r ($1 < r \leq u_c$) 对字符作字符串换位操作。

2.2.2 字符串移位算子,可分为单个字符串移位算子和多个字符串移位算子

单个字符串移位操作是随机取两个正整数 i, j ($1 < i, j \leq n, i \neq j$),在抗体 $A=(c_1, c_2, \dots, c_n)$ 中取一个字符子串 $A_1=(c_i, c_{i+1}, \dots, c_{j-1}, c_j)$,以一定的概率 p_s 依次往右移动 A_1 中的各个字符,最右边的一个字符则移动到最左边的位置;多个字符串移位操作是预先确定一个正整数 u_s ,在抗体 $A=(c_1, c_2, \dots, c_n)$ 中随机取 r ($1 < r \leq u_s$) 个字符子串作字符串移位操作。

2.2.3 字符串逆转算子,可分为单个字符串逆转算子和多个字符串逆转算子

单个字符串逆转操作是随机取两个正整数 i, j ($1 < i, j \leq n, i \neq j$),在抗体 $A=(c_1, c_2, \dots, c_n)$ 中取一个字符子串 $A_1=(c_i, c_{i+1}, \dots, c_{j-1}, c_j)$,以一定的概率 p_i 使 A_1 中的各个字符首尾倒置;多个字符串逆转操作是预先确定一个正整数 u_i ,在抗体 $A=(c_1, c_2, \dots, c_n)$ 中随机取 r ($1 < r \leq u_i$) 个字符子串作字符串逆转操作。

2.2.4 优质字符串的保留。

如果若干个抗体与抗原之间的亲和力都很大,且这些抗体中包含了一个相同的字符子串,则称这个字符子串为优质字

串。如果抗体中存在优质字符串,则在抗体产生过程中以概率 p_o 使该优质字符串不受破坏,即把该优质字符串当成一个字符看待,称为优质字符串的保留。

2.3 算法基本步骤

根据该文算法的设计思想,其基本步骤可描述如下:

步骤 1 输入问题和确定抗体的编码表示。

输入问题的目标函数和约束条件,作为 AIA 的抗原。AIA 的抗体采用自然数编码方式,一个字符串代表一个候选解。对于 n 个城市的 TSP 问题,设其城市编码分别为 $1, 2, \dots, n$,并且把商人出发城市编为第 1 号,其它城市可随意编号。把这 n 个城市的编号任意排列成一个长度为 n 的字符串都可以形成一个抗体,因此抗体空间包含 $n!$ 个抗体。为了缩小抗体空间,提高搜索效率,将每个人工抗体(字符串)的第一个字符固定为出发城市的编号 1。这样,每个抗体只有 $(n-1)$ 个字符可任意排列,抗体空间就只包含 $(n-1)!$ 个抗体。

步骤 2 产生初始抗体并进行预处理。

在一般情况下,可按上述抗体编码方式,在解空间中随机产生 N 个抗体作为初始抗体,构成初始抗体群,其中 N 为抗体群中抗体的数目。考虑到 TSP 问题的任何一条路径都是闭合路径,从任一城市出发,要到达的下一个城市选择为未到过的城市中距该城市最近的一个。为了提高搜索效率,先对每一个初始抗体进行预处理,然后才开始算法的迭代计算。设初始抗体 $A=(c_1, c_2, \dots, c_n)$,其中 $c_1=1$ (即 c_1 代表商人出发城市的编号),对 A 进行预处理的步骤是:

(1)随机取正整数 r ($1 \leq r \leq n$)。若 $r=c_1=1$,则令 $A'=(c'_1, c'_2, \dots, c'_n)=A$,转(3);若 $r=c_k \neq 1$ ($1 < k \leq n$),则转(2)。

(2)对初始抗体 A 中的各个字符依次循环左移位 $(k-1)$ 次,每次移位时,使 c_{i+1} 移到 c_i ($i=1, 2, \dots, i-1$) 的位置,且使 c_k 移到 c_1 的位置。设移位以后初始抗体 A 变为 A' ,则 $A'=(c_k, c_{k+1}, \dots, c_n, c_1, c_2, \dots, c_{k-1})$ 。令 $A'=(c'_1, c'_2, \dots, c'_n)$,其中 $c'_1=c_k$,余类推。

(3)随机取 $c'_m=c'_1$,令 $C=\{c'_2, c'_3, \dots, c'_n\}$,若对 C 中任一元素 c'_k ,都有 $d(c'_m, c'_1) \leq d(c'_m, c'_k)$, $c'_1 \in C$,则把 c'_1 置于 A' 中 c'_{m+1} 的位置,此时 A' 变为 A'' ,令 $A''=(c''_1, c''_2, \dots, c''_n)$ 。然后从 C 中删除元素 c'_1 ,再取 $c'_m=c''_1$,重复上述步骤,直到 C 中的元素全部被删除为止。设这一步完成以后初始抗体 A 变为 B 。

(4)对初始抗体 B 中的各个字符依次循环右移位若干次,直到抗体中第一个字符为 1 为止,移位方法与(2)中的类似,但方向相反。

步骤 3 计算亲和力和排斥力。

对于 TSP 问题,可定义抗体 B 与抗原 G 之间的亲和力 $App(B)=1/(T_B-T_C)$ 。其中 T_B, T_C 分别为抗体 B 与抗原 G 对应的旅行路线的总长度, T_C 也是所求的最短路线的总长度。因在计算结束之前并不知道 T_C 的大小,可用一适当大的正数 T ($T < T_C$) 代替 T_C 。又因计算 $App(B)$ 需作除法运算,可定义 $App(B)=T_M-T_B$ 。其中 T_M 为较大的正数,且要求 T_M 大于任意抗体对应的旅行路线的总长度,则可避免在计算 $App(B)$ 时作除法运算。因此,构造抗体 B 与抗原 G 之间的亲和力 $App(B)=T_M-T_B$ 并计算 $App(B)$ 。抗体与抗原之间的亲和力反映抗体与抗原之间的匹配程度, $App(B)$ 越大,说明抗体 B 与抗原 G 之间的匹配越好。

对于 TSP 问题,可定义抗体 B_1 与抗体 B_2 之间的排斥力 $Re p(B_1, B_2)=|T_{B1}-T_{B2}|$ 并计算 $Re p(B_1, B_2)$ 。其中 T_{B1}, T_{B2} 分别为

抗体 B_1 与抗体 B_2 对应的旅行路线的总长度。抗体与抗体之间的排斥力反映抗体与抗体之间的差距, $\text{Re } p(B_1, B_2)$ 越大, 说明抗体 B_1 与抗体 B_2 之间的差距越大。计算抗体群中所有抗体与当前最佳抗体之间的排斥力。

步骤 4 产生新抗体并计算其亲和力和排斥力。

构造适当的人工免疫算子, 通过人工免疫算子的作用概率 p_c, p_s, p_i, p_o 和预先确定的正整数 u_c, u_s, u_i 产生新抗体; 同时, 计算各个新抗体的亲和力 $\text{App}(B)$ 和新抗体间的排斥力 $\text{Re } p(B_1, B_2)$ 。若新抗体中有与抗原相匹配的抗体, 或已满足预定的停机条件则停机, 从而获得较优的可行解。否则转步骤 5。

步骤 5 抗体选择。

按照“优胜劣汰”的自然选择机制, 在新产生的若干个抗体中, 选择出 N 个与抗原匹配得较好的抗体构成新的抗体群, 转步骤 4。

步骤 6 初始化参数 $\tau_c, \tau_0, m, \rho, \alpha, \beta, Q$, 根据步骤 4 获得的较优可行解, 生成信息素初始分布, 将 m 只蚂蚁置于 n 个结点。

这里, τ_c 是一个根据具体求解规模给定的信息素常数, τ_0 是 AIA 求解结果转换的信息素值, m 是蚁群中蚂蚁的数量, ρ ($0 \leq \rho < 1$) 为信息素轨迹的残留系数, α ($\alpha \geq 0$) 为边 (i, j) 信息素轨迹强度 τ_{ij} 的相对重要性, β ($\beta \geq 0$) 为边 (i, j) 能见度 η_{ij} 的相对重要性, η_{ij} 为边 (i, j) 长度 d_{ij} 的倒数, Q 为蚂蚁循环一周所释放的总信息量。在初始时刻, 根据下式设置信息素的初值:

$$\tau_{ij}(0) = \begin{cases} \tau_c + \tau_0, & \text{若边 } (i, j) \text{ 在步骤 4 获得的较优可行路径上} \\ \tau_c, & \text{否则} \end{cases}$$

步骤 7 计算每只蚂蚁的选择概率, 根据选择概率移动每只蚂蚁到下一结点。

设蚂蚁 k 在 t 时刻的转移概率为 $p_{ij}^k(t)$, 可行顶点集为 U , 则按下式的概率转移规则决定 t 时刻蚂蚁 k 由位置 i 转移到位置 j :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in U} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta}, & \text{若 } j \in U \\ 0, & \text{否则} \end{cases}$$

步骤 8 经过 n 个时刻, m 只蚂蚁遍历个结点, 完成一次循环。

设 $\Delta\tau_{ij}^k$ 为蚂蚁 k 在本次循环中在边 (i, j) 上留下的单位长度轨迹信息量, Z_k 为蚂蚁 k 在本次循环中所走路程的长度。此时, 根据下式对各个路径信息素更新:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}^k$$

$$\text{其中, } \Delta\tau_{ij}^k = \sum_{k=1}^m \Delta\tau_{ij}^k,$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{Z_k}, & \text{若蚂蚁 } k \text{ 在本次循环中经过边 } (i, j) \\ 0, & \text{否则} \end{cases}$$

步骤 9 进行递归循环, 直到满足算法的停止条件为止。

蚂蚁完成一个搜索周期, 进入下一个循环。反复进行递归循环, 直到周游计数器(总循环次数) NC 达到预设值或所有蚂蚁都走同一周游路径便停止计算, 输出近似最优解。

3 仿真实验

为了验证该算法的效果, 从下面两个 TSP 问题进行了仿

真实验, 都取得了比较满意的结果。

3.1 小规模 TSP 问题的实验比较

以 9 个城市的 TSP 问题为例: 设 9 个城市为 1, 2, 3, 4, 5, 6, 7, 8, 9。它们之间的距离如表 1 所示:

表 1 9 个城市之间的距离

	1	2	3	4	5	6	7	8	9
1	0	3	1	12	7	5	6	3	4
2	5	0	7	4	1	8	1	2	3
3	3	5	0	3	1	6	4	9	1
4	7	1	9	0	7	8	9	5	6
5	8	6	6	1	0	13	5	1	10
6	1	4	7	3	2	0	5	2	9
7	9	7	8	8	7	1	0	1	8
8	12	11	3	8	7	1	6	0	5
9	5	3	6	5	1	9	12	4	0

AIA 中, 有关参数取为: $p_c=0.2, p_s=0.3, p_i=0.4, p_o=0.5, u_c=u_s=u_i=[n1/4], [\cdot]$ 表示取整运算, 每次抗体选择时保留与抗原匹配得最好的抗体, 并对随机产生的初始抗体进行预处理, 递归迭代 20 次。ACA 中, 有关参数取为: $\tau_c=20, \tau_0=1, m=3, \rho=0.7, \alpha=0.5, \beta=0.9, Q=100$, 蚁群总循环次数 $NC=20$ 。分别采用 AIA、ACA 和该算法对上述 TSP 问题进行实验比较, 重复运行 10 次, 计算结果如表 2 所示:

表 2 小规模 TSP 问题的比较结果

算法	名称	平均解	最优解	最差解	运行时间(s)
1	AIA	14.8	9	20	33.3
2	ACA	13.4	9	17	56.8
3	该文算法	13.1	9	16	39.8

3.2 大规模 TSP 问题的实验比较

为了更好地验证、比较文章所提出的算法的效果, 选用通用的 TSPLIB(www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95) 中的 eil51TSP 问题为例进行仿真实验。

AIA 中, 有关参数取值同上, 每次抗体选择时保留与抗原匹配得最好的抗体, 并对随机产生的初始抗体进行预处理, 递归迭代 10000 次。ACA 中, $\tau_c=100, \tau_0=4$, 其它参数取自文献[6], 蚁群总循环次数 $NC=510000$ 。分别采用 AIA、ACA 和该算法对 eil51TSP 问题进行实验比较, 重复运行 20 次, 部分搜索结果取自文献[5], 计算结果如表 3 所示:

表 3 大规模 TSP 问题的比较结果

算法	名称	平均解	最优解	最差解	运行时间(s)
1	AIA	431.12	426	452	1.26×10^5
2	ACA	428.06	426	—	2.10×10^5
3	该文算法	427.52	426	442	0.92×10^5

从表 2 和表 3 中的实验数据, 可以清楚地看到: 该文算法在收敛速度上明显优于 ACA, 在寻优能力上明显优于 AIA。

4 结论

该文对于 TSP 问题的求解, 提出了一种新的算法——基于人工免疫算法和蚁群算法的混合算法, 经过仿真实验获得以下结论:

该算法前过程采用人工免疫算法生成信息素分布, 算法后过程利用蚁群算法求优化解, 汲取了两种算法的优点, 在收敛速度上优于蚁群算法, 在寻优能力上优于人工免疫算法, 是收敛速度和寻优能力都比较好的一种新的优化方法。

(收稿日期:2004年7月)

参考文献

1. Dorigo M, Bonabeau E, Theraulaz G. Ant algorithm and stigmergy[J]. Future Generation Computer Systems, 2000; 16(8): 851~871
2. Tsai CF, Tsai CW. A new approach for solving large traveling salesman problem using evolution ant rules[C]. In: Proc of the 2002 Int, 1 Joint Conf on Neural Networks, IJCNN 2002 Honolulu: IEEE Press, Vol 2, 2002: 1540~1545
3. Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm[J]. IEEE Trans on Evolutionary Computation, 2002; 6(4): 321~328
4. 丁建立, 陈增强, 袁著祉. 遗传算法与蚂蚁算法的融合[J]. 计算机研究与发展, 2003; 40(9): 1351~1356
5. [日]Drik Slama, [美]Jason Garbis, [澳]Perry Russell. CORBA 企业解决

方案[M]. 北京: 机械工业出版社, 2001

6. 詹士昌, 徐婕, 吴俊. 蚁群算法中有关算法参数的最优选择[J]. 科技通报, 2003; 19(5): 381~386
7. 丁永生, 任立红. 人工免疫系统: 理论与应用[J]. 模式识别与人工智能, 2000; 13(1): 52~59
8. 李茂军, 舒宜, 童调生. 旅行商问题的人工免疫算法[J]. 计算机科学, 2003; 30(30): 80~82
9. 朱燕飞, 蔡永昶, 李中华等. 人工免疫算法在过程数据分析中的应用[J]. 计算机工程与应用, 2004; 40(6): 205~207
10. 韩健, 张乐, 蔡瑞英. 基于人工免疫算法的入侵检测系统[J]. 南京工业大学学报(自然科学版), 2004; 26(1): 48~51
11. 郑日荣, 毛宗源. 一种改进的人工免疫算法[J]. 计算机工程与应用, 2003; 39(33): 55~57
12. 王磊, 肖人彬. 基于免疫记忆的人工免疫算法模型及其应用[J]. 模式识别与人工智能, 2002; 15(4): 385~391

(上接 35 页)

$$S.t: A_{q,i} + G_i = S_i + G_{i-1} + \Delta_{q,i}, \quad i=1, \dots, T, q=1, \dots, Q \quad (9)$$

$$W_{q,i} - W_{q,i-1} - A_{q,i} = -(M_i - \psi_{q,i}) \quad i=1, \dots, T, q=1, \dots, Q \quad (10)$$

$$W_{q,i} \geq W_{q,i-1} + A_{q,i} - M_i$$

$$W_{q,i} - \beta \cdot r_{q,i} \leq 0$$

$$W_{q,i} - (W_{q,i-1} + A_{q,i} - M_i) + \beta \cdot r_{q,i} \leq \beta \quad i=1, \dots, T, q=1, \dots, Q \quad (11)$$

$$G_i \in Z_+, A_{q,i} \in Z_+, W_{q,i} \in Z_+, \psi_{q,i} \in Z_+, r_{q,i} = 0, 1 \quad (12)$$

$$\text{初始条件 } G_0 = G_{T+1} = 0, W_{q,0} = W_{q,T+1} = 0, k=1, \dots, K; q=1, \dots, Q$$

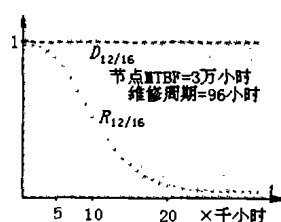
仍成立。

5 结语

基于需求随机的随机样本处理方法, 该文给出了需求随机型单机场地面等待问题的具体描述, 并建立了问题的数学模型, 而且对该模型进行了改进。相关定理表明了该文所建模型的合理性和正确性。文章的研究结果对求解空中交通流量管理问题有一定参考价值。(收稿日期: 2004年8月)

(上接 46 页)

其中节点可靠度 $R = e^{-T/MTBF}$, 这里 T 为节点工作周期。计算 16 取 12 集群系统可靠度 $R_{12/16}$ 在不同工作周期内的可靠性变化曲线如图 4 所示。

图 4 $R_{12/16}/D_{12/16}$ 时间曲线图

在维修周期 T_{repair} 为 96 小时的情况下, 维修周期内节点可靠度为 $R_{T=96} = e^{-96/MTBF}$, 系统可信度 $D_{k/n}$ 为:

$$D_{k/n} = \sum_{i=k}^n \frac{n!}{(n-i)!i!} R_{T=96}^i (1-R_{T=96})^{n-i}$$

16 取 12 集群系统不间断运行的可信度 $D_{12/16}$ 为:

$$D_{12/16} = 0.99999998588$$

从可信度计算公式可以看到, $D_{k/n}$ 仅仅和系统维修周期相关, 和系统工作周期无关, 在可维护性成立的条件下, 系统在整个生命周期内具备不衰退的、十分可观的不间断运行可信度。

参考文献

1. Andreatta G et al. Aircraft Flow Management under Congestion[J]. Transportation Science, 1987; (21): 249~253
2. Hoffman R, Ball M. A Comparison of Formulations for the Single-Airport Ground Holding Problem with Banking Constraints[R]. Technical research report, T R 98-44, <http://www.isr.umd.edu>, 1997
3. S Antonio Alonso, Laureano F Escudero, Maria Teresa Ortuño. A Stochastic 0-1 Program Based Approach for the Air Traffic Flow Management Problem[J]. European Journal of Operational Research, 2000; 120: 47~62
4. Giovanni Andreatta, Lorenzo Brunetta, Guglielmo Guastalla. From ground holding to free flight: An exact approach[J]. Transportation Science, 2000; 34(4): 394~401
5. Hoffman R. Integer Programming Models for Ground-Holding in Air Traffic Flow Management[D]. America: University of Maryland, 1997

对 $D_{12/16}$ 的计算结果说明, 这一系统在工作过程中的不间断运行能力是值得信赖的。单纯采取可靠性措施, 很难达到这一指标。这就是文章基于可信度设计系统的目的。

7 结论

事实上, 单纯依赖多重冗余的高可靠设备构建关键实时计算系统, 成本十分昂贵。以可信度理论为指导, 综合采取技术(故障沉默和恢复等)、后勤保障(可靠迅速的备件支援)、人员(经过培训的现场维修人员)等方面的措施, 用一般商业化(COTS)设备以低廉成本构建起来的实时集群系统, 能够实现关键计算应用要求的苛刻不间断工作能力。除了地面系统外, 这一技术也适应于有后勤储备的船载实时计算系统。

(收稿日期: 2004年5月)

参考文献

1. Algirdas Avizienis, Los Angeles. Fundamental Concepts of Dependability. <http://www.cert.org/research/isw/isw2000/papers/56.pdf>
2. Algirdas Avizienis, Los Angeles. Fundamental Concepts of Computer System Dependability. <http://www.cs.virginia.edu/~jck/cs651/papers/laprie.taxonomy.pdf>
3. James L Peterson. Petri Net Theory and the Modeling of Systems. <http://www.cis.umassd.edu/~hxr/courses/cis602/papers/petrinet/>