利用 GA 决策理论模型增强信息系统的生存性

丁莉娜[1], 郑 宁[1], 孔 霆[2]
DING Li-na[1], ZHENG Ning[1], KONG Ting[2]

1.杭州电子科技大学 计算机学院, 杭州 310018
2.浙江警察学院 实验中心, 杭州 310053
1.Department of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China
2.Experiment Center, Zhejiang Police College, Hangzhou 310053, China
E-mail: tianhxk_dln@yahoo.com.cn

DING Li-na, ZHENG Ning, KONG Ting.Using GA decision-theoretic model to enhance survivability of information system.Computer Engineering and Applications, 2008, 44(19): 163-165.

Abstract: Survivability controller is widely adopted into critical information system.A main function of the controller is to give policies which select corresponding sequence of multiple actions in a user-defined action set based on the reward assessment. Therefore, the quality of policy given by the controller determines its ability.The objective of this paper is to find out an efficient solution approach to determine Action Sequence(AS) achieving the maximal rewards.AS is a mixture of Knapsack Problem(KP) and Traveling Salesman Problem(TSP).This work has developed a methodology based on Genetic Algorithm(GA) for effective solving combination optimization.Special coding and effective genetic operators are designed for this genetic algorithm.Comparing with greedy algorithm simulation-based experimental results demonstrate the validity and practicability of the Genetic Algorithm.
Key words: survivability controller; Action Sequence(AS); Knapsack Problem(KP); Traveling Salesman Problem(TSP); Genetic Algorithm(GA)

摘 要: 生存控制器被广泛地应用在关键的信息系统中。生存控制器的一个重要功能是做决策, 也就是基于收益评价从用户给出的行动集合中选择相应的行动序列。因此, 决策的质量决定了控制器的能力。寻找一个有效地解决方案来确定获得最大收益的行动序列(AS)。AS 是一个背包问题(KP)和旅行商问题(TSP)的混合体。以 GA 有效解决组合优化问题的方法论为基础, 针对 AS 问题设计了特殊的编码和有效的遗传操作。通过与贪心算法进行比较, 模拟实验结果证明了遗传算法的有效性和实用性。
关键词: 生存控制器; 行动序列; 背包问题; 旅行商问题; 遗传算法
DOI: 10.3778/j.issn.1002-8331.2008.19.049 **文章编号**: 1002-8331(2008)19-0163-03 **文献标识码**: A **中图分类号**: TP393

1 引言

关键基础信息系统的生存性日益得到工业界的广泛关注。KNIGHT[1]等学者在 1999 年提出基于控制系统的解决方案, 生存控制器(下文简称控制器)应运而生。通过监视基础信息系统, 控制器对攻击、故障等意外事故做出相应的响应, 尽可能地减小这些事故造成的损失, 保证用户服务的持续进行。控制器是保障信息系统可生存性的有效工具。

Kaustubh 和 Cassandra[2,3]指出, 控制器用来决策的信息是不完全的, 而不完全信息情况下的决策过程模型是部分可观察的马尔可夫决策过程(POMDP)[4,5]。引入遗传算法到基于 POMDP 框架的控制器中, 选择行动以前, 对将来的行动序列进行评估, 在可行的行动序列中找到使收益最大或者近似最大的行动序列。在整个过程中, 这种算法支持多步的、状态相关的行动与收益规则; 并构造最优决策, 用以确保总收益的最大化。比较遗传算法和贪心算法的有效性, 模拟试验结果表明, 采用遗传算法的控制器比采用贪心算法的控制器做出的决策要好得多。

2 基于决策理论的控制器

控制器在选择和控制行动的过程中主要执行以下任务

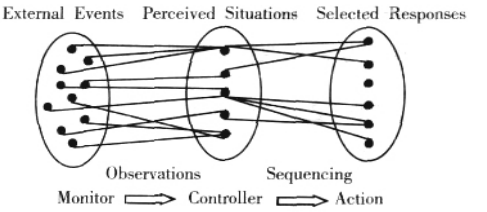

图 1 控制器的工作

基金项目: 浙江省自然科学基金(the Natural Science Foundation of Zhejiang Province of China under Grant No.Y106176); 浙江省科技厅科技计划项目(the Technology Planning Projects of the Science and Technology Office of Zhejiang Province under Grant No.2007C33058)。
作者简介: 丁莉娜(1982-), 研究生, 研究方向为信息安全; 郑宁(1961-), 研究员, 博士生导师, 研究方向为计算机应用; 孔霆(1981-), 硕士, 研究方向为图形图像, 信息安全。
收稿日期: 2007-09-28 修回日期: 2007-12-13

(图 1)[6]:(1)收集潜在问题的观察结果;(2)决定目标系统的状态;(3)确定当前状态下采取的行动的收益;(4)选择使总的收益最大化的一个或多个行动;(5)执行选择的行动。

生存控制器的一个重要功能是做决策,也就是基于收益评价从用户给出的行动集合中选择相应的行动序列。因此,决策的质量决定了控制器的能力。构造一个确保总收益最大化的决策是值得思考的问题。

3 问题描述和表示

3.1 POMDP

1 个 POMDP 被定义为 1 个(S,A,O,p(·|s,a),q(·|s),r(s,a))的六元组,其中:S 代表有限的状态的集合;A 代表行动的集合;O 代表 1 个有限的观察 o 的集合;p 代表转移概率函数,例如 p(s'|s,a),其中 s',s∈S,a∈A,代表在状态 s 的时选择行动 a 转移到状态 s 的概率;q(o|s,a)表示任何情况下,系统通过行动 a 到达状态 s,观察到 o 的概率。本文假设与所选择的行动无关,也就是,q(o|s,a)=q(o|s),q(o|s)代表系统在状态 s 下观察到 o 的概率;r(s,a)代表报酬函数,它代表在状态 s 时,采取行动 a 获得的收益。可以为正也可以为负。

POMDP 可以看成是基于信念状态的 MDP。信念状态 π=[π(1),π(2),…,π(S)],表示系统处在状态集合中各状态的概率分布。POMDP 的值(最大累积收益)可以用函数 V(π)表示:

$$V(\pi)=\max_{a\in A}\{\pi(a)+V(\pi^{\pi,a})\} \quad (1)$$

这里,r(a)=[r(s,a),s∈S]^T,代表选择行动 a 的收益列向量;$\pi^{\pi,a}$代表下一个信念状态。

接收到 1 个观察将引起信念状态的更新。公式(2)是贝叶斯原理的直接应用,当 o 正确观察,目标系统处于 s 状态的概率等于在状态 s 下观察到 o 的概率占所有状态下观察到 o 的概率的比例:

$$\pi(s)=\frac{\pi(s)q(o|s)}{\sum_{s'\in S}\pi(s')q(o|s')} \quad (2)$$

执行 1 个行动也将引起信念状态的更新:

$$\pi(s)=\sum_{s'\in S}\pi(s')p(s|s',a) \quad (3)$$

因此,给定当前的信念状态 π,选择行动 a,和观察结果 o,下一个信念状态可以用以下公式来计算:

$$\pi^{\pi,a,o}(s)=\frac{q(o|s)\sum_{s'\in S}p(s|s',a)\pi(s')}{\sum_{s'\in S}q(o|s')\sum_{s''\in S}p(s'|s'',a)\pi(s'')} \quad (4)$$

假设当前信念状态为 π,采取行动 a,则观察到 o 的概率:

$$\gamma^{\pi,a}(o)=\sum_{s\in S}q(o|s)\sum_{s'\in S}p(s|s',a)\pi(s') \quad (5)$$

综合公式(4)和公式(5)可得:

$$\pi^{\pi,a}(s)=\frac{\sum_{o\in O}\gamma^{\pi,a}(o)\pi^{\pi,a,o}(s)}{\sum_{o\in O}\gamma^{\pi,a}(o)} \quad (6)$$

3.2 KP,TSP 和 AS

TPS 和 KP 是近代组合优化领域中的经典难题,具有广泛的应用背景和重要的理论价值,它已经被证明属于 NP 难题。AS 是 KP 和 TSP 的混合体。

AS 和 KP:它们都是最大化问题。并且它们都是 0-1 规划问题,任何一个物体(行动)可以被选择也可以不被选择,一旦选择将会对收益产生影响。由于在 AS 中,当前选择的行动会对下一个行动产生影响,它们最大的不同是,AS 是一个排列问题,而 KP 是一个组合问题。

AS 和 TSP:它们都是与排列问题,与序列有关。不同之处是:(1)在 TSP 中,每一个城市都必须被遍历一次,但是在 AS 中,并不需要选择行动集合中的每一个行动;(2)在 TSP 中,每两个城市的距离是固定的,但是在 AS 中,选择某个行动所获得的收益是动态的,与状态相关的;(3)AS 是最大化问题,TSP 是最小化问题。

4 遗传算法设计

遗传算法[7,8]与问题的特定领域无关,是解决复杂系统优化问题的通用框架。它被广泛地应用在组合优化问题中,提供了一个以有限代价解决搜索和优化的有效途径。论文引入遗传算法来解决 AS 问题,并针对 AS 问题设计了特殊的编码和有效的遗传操作。

4.1 适应度函数

适应度函数的核心是目标系统的基于信念状态的模型。采用 POMDP 的值作为适应度,公式(1)定义了 POMDP 的值,它是一个从当前信念状态开始的动态规划的递归式。

4.2 编码

采用路径表示法对 AS 进行编码,在路径表示法中,按照每个行动被选择的先后次序来排列。没有被选择的行动将不出现在排列中。由于究竟会选择多少个行动是不可预知的,所以路径的长度也是待定的。例如,给定行动集合 A={$a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9$},代表行动序列 $a_3,a_1,a_2,a_4,a_8,a_7,a_6$ 的个体将被编码为(3,1,2,4,8,7,6)。

4.3 交叉算子

许多学者针对路径表示法提出了自己的交叉算子,例如,PMX[9],OX[10],CX[11]等。但上述交叉算子并不适合 AS 问题,因为路径可能没有包含 A 中所有的元素,并且它的长度是不确定的。本文设计了针对 AS 问题的交叉算子,分成部分交叉和基于贪心算法的排序两个步骤:

(1)部分交叉

将 A 划分成两个分离的子集 A_1 和 A_2,$A_1\cap A_2=\varnothing$。规定:父个体 p_1 决定子个体中所有属于 A_1 的行动的排序,也就是说,如果父个体中有 $a_i,a_j\in A_1$,$a_i>a_j$(a_i 排在 a_j 前面),则子个体中 $a_i>a_j$ 也必定成立。如果 p_1 没有选择 $a\in A_1$,则它的子个体也不会包含 a。同样地,规定父个体 p_2 决定其子个体中所有属于 A_2 的行动的排序。

假设 $A_1=\{a_1,a_2,a_3,a_4\}$,$A_2=\{a_5,a_6,a_7,a_8,a_9\}$,双亲:$p_1$=(3,4,8,7,1,2,6,9,5),$p_2$=(3,5,8,4,2,7,6),则子代必须满足 $a_3>a_4>a_1>a_2$ 和 $a_5>a_8>a_7>a_6$(图 2)。但是这 8 个行动的完整排序却是未知的。

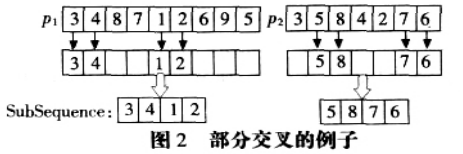

图 2 部分交叉的例子

(2)基于贪心算法的排序

在部分交叉算子的基础上,使用贪心算法来寻找一个完整的行动排序。首先,把通过部分交叉算子得到的 2 个子序列放在 2 个数组中,$X_1=\{3,4,1,2\}$,$X_2=\{5,8,7,6\}$。选择 a_3 作为结果序列中的第 1 个行动。假设当前信念状态为 $\pi_{current}$,那么 $F=\pi_{current}(a_3)$。将 a_3 和 $\pi_{current}$ 代入公式(6)中计算出下一个信念状态 π_{next};接下来,选择下一个行动。由于 X_1 和 X_2 中的行动排序已经固定了,那么下一个行动要么是 a_4 要么是 a_5。令 $\pi_{current}=\pi_{next}$,选择当前信念状态下可获得较大收益的行动。如果 $\pi_{current}r(a_5)>\pi_{current}(a_4)$,将 a_5 从 X_2 中移出并添加到结果集合中。更新适应度:$F=F+\pi_{current}(a_5)$。根据 $\pi_{current}$ 和 a_5 计算下一个信念状态 π_{next}。重复以上操作直到 X_1 和 X_2 为空,可以得到结果序列(3,5,4,1,8,2,7,6)。如果选择 a_5 作为结果序列的第一个行动,通过以上操作可以得到另一个子个体(5,3,4,8,7,6,1,2)。

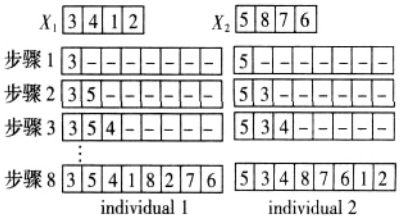

图 3　全排列的例子

以上遗传操作步骤如下:

步骤 1　双亲经过部分交叉操作,得到下一代的 2 个子序列:$X_1=\{a_{1,1},a_{1,2},a_{1,3},\cdots,a_{1,n}\}$,$X_2=\{a_{2,1},a_{2,2},a_{2,3},\cdots,a_{2,m}\}$;

步骤 2　将 $a_{1,1}$ 作为结果序列的第 1 个行动。假设当前信念状态为 $\pi_{current}$,则适应度 $F=\pi_{current}(a_{1,1})$。将 $\pi_{current}$ 和 $a_{1,1}$ 代入公式(6)中,计算出下一个信念状态 π_{next}。令 i=2,j=1;

步骤 3　如果 $i\leq n$ 且 $j\leq m$,比较 $\pi_{current}(a_{1,i})$ 和 $\pi_{current}(a_{2,j})$:如果 $\pi_{current}(a_{1,i})$ 大,则将 $a_{1,i}$ 加入结果序列中,$F=F+\pi_{current}(a_{1,i})$,根据 $\pi_{current}$ 和行动 $a_{1,i}$ 计算,令 i=i+1;否则将 $a_{2,j}$ 加入结果序列中,$F=F+\pi_{current}(a_{2,j})$,根据 $\pi_{current}$ 和行动 $a_{2,j}$ 计算 π_{next},令 j=j+1;

步骤 4　如果 $i=n+1$ 且 $j\leq m$,则将 $a_{2,j}$ 加入结果序列中,$F=F+\pi_{current}(a_{2,j})$,根据 $\pi_{current}$ 和行动 $a_{2,j}$ 计算 π_{next},令 j=j+1;

步骤 5　如果 $j=m+1$ 且 $i\leq n$,则将 $a_{1,i}$ 加入结果序列中,$F=F+\pi_{current}(a_{1,i})$,根据 $\pi_{current}$ 和行动 $a_{1,i}$ 计算 π_{next},令 i=i+1;

步骤 6　转步骤 3,直至 $i=n+1$ 且 $j=m+1$;

如果将 $a_{2,1}$ 作为结果序列的第 1 个行动,通过以上遗传操作可以得到另一个子个体。

4.4　变异算子

采用选择个体中任意 2 个基因并交换它们的值的操作来代替传统的变异算子,交换以后,仍然可以得到 1 个合法的序列。例如,(3,5,4,1,8,2,7,6)通过交换 4 和 8 可以变异为(3,5,8,1,4,2,7,6)(图 4)。

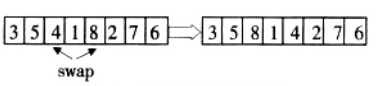

图 4　变异操作的例子

5　实验结果

选择简单的 EMN[12]系统运行控制器算法,实验通过向该 EMN 系统注入 10 000 个故障来评估控制器做出的决策的质量。这里提供两种控制器算法,一种是本文中构造的遗传算法,另一种是贪心算法。贪心算法基于当前的信念状态选择可获得最大收益的行动,并且每次只选择一个行动。在遗传算法中,使用以下默认参数:初始种群大小 S=50,交叉概率 $P_c=0.8$,变异概率 $P_m=0.1$,以及最大遗传代数 G=100。目标系统每一次实验运行 10 天,每一种算法分别运行 100 次。

图 5 显示当向目标系统注入故障时,平均无故障时间的异常概率函数。

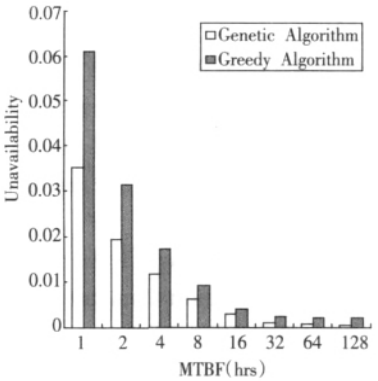

图 5　故障注入后的异常概率函数

表 1 给出了算法的详细测试结果。收益指从一个故障恢复过来的平均收益值;残余时间指平均每个故障在目标系统中残留的时间;算法时间指执行一次算法的平均时间(不包括执行行动的时间)。

表 1　故障注入后的测试结果(每个故障的平均值)

算法	收益	残余时间/s	算法时间/ms
遗传算法	244.29	137.71	57.98
贪心算法	89.53	223.15	0.09

通过观察以上实验结果,可以得出如下结论:

(1)遗传算法在收益方面明显优于贪心算法。每个故障的平均收益取决于所选行动的整个序列,目前最优的并不代表长远最优的。故在一个时间段内,贪心算法并不能使平均收益最优。遗传算法是寻求组合优化问题的最佳工具之一,利用遗传算法获得的平均收益比贪心算法提高了 173%。与贪心算法相比,遗传算法还有如下优越性:较短的残余时间,很高的可靠性。

(2)贪心算法的时间比较短,但遗传算法已经能够足够快地对一个实时的信息系统做出相应。

遗传算法是实际可行的,作为一个低代价、高效率的解决方案,可以广泛地被用来提高小型信息系统的生存性。

6　结论

本文将遗传算法引入到基于 POMDP 框架的生存控制器中,使其能够更好地做出决策。遗传算法通过它特殊的行为和的操作能够有效地找出优化的解决方案。实验结果表明,在一个可以接受的时间之内,遗传算法确实切实可行。

接下来的工作是:(1)为了提高算法的效率,将要深入地研究每一个参数对算法执行的影响;(2)进一步研究如何将提出的算法应用到大规模的信息系统中。

参考文献:

[1] Knight J,Sullivan K.Information survivability control systems[C]// Proceedings of the 21st International Conference on Software En-

(下转 168 页)

定义 2 平均召回率(AR) $=\frac{\sum P}{n}$ (20)

若 n 为测试样本总数,称为宏平均召回率(MAAR);若 n 为兼类数相同的样本数,称为微平均召回率(MIAR)。

定义 3 平均 F1 值(AF) $=\frac{\sum F1}{n}$ (21)

若 n 为测试样本总数,称为宏平均 F1 值(MAAF);若 n 为兼类数相同的样本数,称为微平均 F1 值(MIAF)。

实验环境为 CPU Pentium 1.6 G,内存 512 M,操作系统 Windows XP。使用的核函数为径向基函数(Radial Basis Function, RBF) $K(x, y)=e^{-\gamma\|x-y\|^2}$,其中 γ=0.01,系统参数 v=0.6。算法实现参考了 Chang 和 Lin 所开发的 libsvm[9],并在此基础上进行了相应的修改。

表 2 和表 3 给出了对测试样本的分类结果。由表 2 和表 3 可以看出,MHSSVM 在保证单类文本分类精度的同时,实现了对兼类文本的分类,并且具有较好的准确率、召回率和 F1 值。该算法不仅适应庞大的兼类识别问题,并且具有较强的扩展能力。

表 2 MHSSVM 的微平均准确率、微平均召回率和微平均 F1 值

兼类数(样本数目)	1(212)	2(26)	3(2)
MIAP	71.94%	76.28%	66.67%
MIAR	75.47%	50.00%	66.67%
MIAF	73.11%	58.59%	66.67%

表 3 MHSSVM 的宏平均准确率、宏平均召回率和宏平均 F1 值

MAAP	MAAR	MAAF
72.36%	72.64%	71.49%

5 结论

本文针对兼类文本的分类,提出了一种基于超球支持向量机的兼类文本分类算法。对具有同一兼类的文本,利用超球支持向量机在特征空间求得一个能够包围该类尽可能多映射的最小超球,使各类文本之间通过超球隔开,达到分类效果。每个超球的训练,只针对一类文本,因此,计算复杂度低,训练速度快。对待分类文本,计算它到各超球球心的距离,根据距离判定该样本所属的类别,分类简单快捷。实验结果证明,该算法不仅具有较快的训练速度,而且具有较高的分类速度和分类精度,是一种较为实用的兼类文本自动分类方法。

参考文献:

[1] Vapnik V.The nature of statistical learning theory[M].New York: Springer, 1995.

[2] Joachims T.Text categorization with support vector machines: learning with many relevant feature[C]//Proceedings of 10th European Conference on Machine Learning, ECML-98.Berlin: Springer, 1998: 137-142.

[3] Bennett K P.Combining support vector and mathematical programming methods for classification[M].Advances in Kernel Methods: Support Vector Learning.Cambridge, MA: MIT Press, 1999: 307-326.

[4] Krebel U G.Pairwise classification and support vector machines[M]. Advances in Kernel Methods: Support Vector Learning.Cambridge, MA: MIT Press, 1999: 255-268.

[5] Platt J C, Cristianini N, Shawe-Taylor J.Large margin DAGs for multiclass classification[M].Advances in Neural Information Processing Systems.Cambridge, MA: MIT Press, 2000: 547-553.

[6] 朱美琳,杨佩.基于支持向量机的多分类增量学习算法[J].计算机工程, 2006, 32(17): 77-79.

[7] 张翔,肖小玲,徐光祐.基于样本之间紧密度的模糊支持向量机方法[J].软件学报, 2006, 17(5): 951-958.

[8] 唐发明,王仲东,陈绵云.支持向量机多类分类算法研究[J].控制与决策, 2005, 20(7): 746-749.

[9] Chang C C, Lin C J.LIBSVM: a library for support vector machines [J/OL].Journal of Machine Learning Research, 2005, 6: 1889-1918. [2007-04].http://www.csie.ntu.tw/~cjlin/libsvm.

(上接 165 页)

gineering.Los Alamitos, CA: IEEE Computer Society Press, 1999: 49-71.

[2] Kaustubh R, Matti A, William H, et al.Automatic recovery using bounded partially observable markov decision processes [C]// Proceedings of the 36th International Conference on Dependable Systems and Networks, 2006.

[3] Cassandra A, Nodine M, Bondale S, et al.Using POMDP-based state estimation to enhance agent system survivability[C]//Proceedings of 2005 IEEE 2nd Symposium, 2005: 11-20.

[4] Anthony R, Lcslic P, Michael L.Acting optimally in partially observable stochastic domains[C]//Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, 1994.

[5] Cassandra A, Littman M, Zhan G, et al.Incremental pruning: a simple, fast, exact method for partially observable markov decision process[C]//Proceedings of the 3rd Conference on Uncertainty in Artificial Intelligence, San Mateo, 1997: 54-61.

[6] Aaron H, Karl K, Marshall B.A framework to control emergent survivability of multi agent systems[C]//Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, 2004: 28-35.

[7] Holland H.Adaptation in natural and artificial systems[M].Ann Arbor, Michigan: University of Michigan Press, 1975.

[8] Goldberg D, David E.Genetic algorithms in search, optimization and machine learning[M].Massachusetts: Addison-Wesley Publishing Company, 1989.

[9] Goldberg D E, Lingle R.Alleles, loci, and the traveling salesman problem[C]//Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, 1985: 154-159.

[10] Davis L.Job shop scheduling with genetic algorithms[C]//Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications.USA: Lawrence Erlbaum Associates, 1985: 136-140.

[11] Oliver L M, smith D J, Holland J R C.A study of permutation crossover operators on the traveling salesman problem[C]//Proceedings of the 2nd International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, 1987: 224-230.

[12] Chen Y, Huang H, Jana R, et al.iMobile EE-an enterprise mobile service platform[J].ACM Journal on Wireless Networks, 2003, 9(4): 283-297.