

统计算法选择对统计模型检测效率的影响分析

高婉玲 洪 玫 杨秋辉 赵 鹤
(四川大学计算机学院 成都 610065)

摘 要 近年来,统计模型检测技术已经得到了广泛的应用,不同的统计算法对统计模型检测的性能有所影响。主要对比不同统计算法对统计模型检测的时间开销影响,从而分析算法的适用环境。选择的统计算法包括切诺夫算法、序贯算法、智能概率估计算法、智能假设检验算法及蒙特卡罗算法。采用无线局域网协议验证和哲学家就餐问题的状态可达性验证为实例进行分析,使用 PLASMA 模型检测工具进行验证。实验结果表明,不同的统计算法在不同的环境中对模型检测的效率有不同的影响。序贯算法适用于状态可达性性质的验证,时间性能最优;智能假设检验算法与蒙特卡罗算法适合验证复杂模型。这一结论有助于在模型检测时对统计算法的选择,从而提高模型检测的效率。

关键词 统计模型检测,统计算法,无线局域网协议,哲学家问题,PLASMA

中图分类号 TP311 **文献标识码** A

Efficiency Analysis of Different Statistical Algorithms on Statistical Model Checking

GAO Wan-ling HONG Mei YANG Qiu-hui ZHAO He

(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract Recently, statistical model checking technology has been widely used, and different statistical algorithms have different effects on the performance of the statistical model checking. This paper mainly compared the running time of different statistical algorithms, thus analyzed the applicable environment of the algorithms. The statistical algorithms include Chernoff algorithm, sequential algorithm, smart aim-listed probability estimation algorithm, smart content testing algorithm and Monte Carlo algorithm. Models are the Wireless LAN (WLAN) and the Dining Philosophers problem, using PLASMA model checking tool for validation. The result shows that different statistical algorithms have different influences on the efficiency of model checking when the environment is different. Sequential algorithm is fit for verifying the reachability of state, and the time performance is the best. Smart content testing algorithm and Monte Carlo algorithm are fit for verifying complex models. This conclusion can help the selection of statistical algorithms in model checking, in order to improve the efficiency of model checking.

Keywords Statistical model checking, Statistical algorithm, WLAN, Dining philosophers, PLASMA

1 引言

随着计算机的快速发展,计算系统和网络系统广泛应用于各个领域。因此,系统的安全性、可靠性及资源利用率成为一个重要问题。然而,系统复杂性的迅速增加将导致许多挑战。近年来,统计模型检测(Statistical Model Checking, SMC)技术已被提出^[1-4]。统计模型检测技术主要模拟待验证系统的执行路径,通过伯努利实验,判断每次模拟路径是否满足给定的系统属性约束;其次使用假设检验的方法对系统模拟路径的样本空间进行统计分析,评估系统是否满足属性。因此,SMC 技术可以规避目标系统的复杂的内部逻辑来检测系统的可靠性。目前,SMC 技术引起了学术界和工业界的广泛关注,已成功应用于实时调度、智能家居等领域^[5-6]。

本文通过实验来分析在统计模型检测中统计算法选择对模型检测效率的影响。以无线局域网协议验证和哲学家问题的验证为例,选择蒙特卡罗算法^[7]、切诺夫算法^[8-9]、序贯算法^[10]、智能概率估计算法和智能假设检验算法^[11],对统计模型检测效率的影响进行分析,分析各种算法在效率(验证时

间)方面的优劣。实验结果表明,序贯算法逐次抽样的思想更适合于验证模型的状态可达性,其时间性能优于切诺夫算法;智能假设检验算法的时间性能优于智能概率估计算法,更适合验证复杂模型的状态可达性;由于模型复杂性对蒙特卡罗算法的收敛速度影响较小,因此蒙特卡罗算法适用于处理复杂问题。

本文第2节简单介绍了所采用的统计模型检测算法;第3节展示了用于验证算法性能的实验对象、无线局域网协议和经典的哲学家就餐问题;第4节是实验设计;第5节是实验结果及分析;最后总结全文。

2 算法介绍

统计模型检测技术主要包括假设检验与概率估计两大类^[12]。具体算法包含随机模拟的蒙特卡罗算法、离散独立变量的经典切诺夫算法和逐次抽样的序贯算法等,这些算法已比较成熟,并成功用于随机系统的分析中;除此之外,抽样样本不固定的智能假设检验与智能概率估计算法可以模拟系统的最佳执行路径,并验证模型的特性。以上算法被集成在模

本文受嵌入式系统软件形式化验证技术研究(2014JY0112)资助。

高婉玲(1992—),女,硕士生,主要研究方向为软件形式化验证与软件自动化测试,E-mail:gwllinyu@163.com;洪玫(1963—),女,硕士,教授,主要研究方向为软件工程,E-mail:hongmei@scu.edu.cn;杨秋辉(1970—),女,博士,副教授,主要研究方向为软件工程,E-mail:yangqiuhi@scu.edu.cn;赵鹤(1992—),女,硕士生,主要研究方向为软件形式化验证与软件自动化测试,E-mail:1085593477@qq.com。

型检测工具中,可以用于验证系统模型是否满足给定性质。

2.1 蒙特卡罗算法

蒙特卡罗算法(Monte Carlo Algorithm)^[7]于1946年由美国拉斯阿莫斯国家实验室的3位科学家John von Neumann, Stan Ulam 和 Nick Metropolis 共同提出。蒙特卡罗算法又称统计模拟算法、随机抽样技术。

它的基本思想是:对于求解的问题,首先建立一个概率模型和随机过程,使它的参数等于问题的解;然后通过对模型或随机过程的观察或抽样试验来计算所求参数的统计特征,最后给出近似解。蒙特卡罗算法的解不一定是准确的,其正确性和准确性依赖概率论,具有严格的数学基础,并且可以通过数学分析手段对实验进行控制。只要模拟的次数足够多,就可得到一个比较精确的概率,将误差和错误率降至可容忍范围。

蒙特卡罗算法的最大特点是简单快速。该算法避免了数学困难,有很强的适应性,能够较好地解决多维等高难度的数学计算问题,在统计物理、信息科学、公用事业、地质、医学及计算机科学等领域都得到了成功的应用。

2.2 切诺夫算法

切诺夫算法(Chernoff Algorithm)^[8-9]于1923年由美国数学家、统计学家及物理学家 Herman Chernoff 提出,并以其名字命名。

切诺夫算法要求变量是独立的,使用两两独立的样本点时误差概率近似直线下降;当使用完全独立的样本点时,误差概率以近似指数分布的形式下降,此概率的边界是切诺夫界。它的基本思想是:对于要求解的问题,首先给定一系列独立的观察值和概率,然后计算实际值与期望值的误差,经过切诺夫公式推导,得到观察值在给定误差和置信度条件下的平均概率的区间。

切诺夫算法的表现形式多样,可用于计算独立泊松事件与离散独立随机变量的概率区间。通过探索随机的扰动空间,它可以有效地评估应用或算法的鲁棒性水平。切尔诺夫界也被用来计算稀疏网络中的排列路由问题的严格边界,所以在稀疏网络设置平衡和包的路由中得到了很好的应用。

2.3 序贯算法

序贯算法(Sequential Algorithm)^[10]于1947年由Wald A提出,它是数理统计学的一个重要分支,一般性的序贯方法称为序贯概率比检验。序贯方法有两个要素:停止法则与判决法则。停止法则指示何时停止逐次抽样的过程,判决法则指示何时作出推断或选择。它的基本思想是:以 Gauss-Markov 模型为基础,建立处理测量数据的函数模型,给定初值,然后在经过有限次迭代后计算获得结果。

序贯分析的特点是:在研究决策问题(统计推断或选择)时,不预先固定样本量,而是逐次取样,先抽少量样本,根据其结果再决定停止抽样或继续抽样,直到样本能提供足够的信息,恰当做出决策为止。

这种算法主要用于变形观测的数据处理,也曾用于军需验收工作,此外还可用于一般的统计决策、点估计、区间估计等。

2.4 智能概率估计算法

本文采用的智能概率估计算法(Smart Probability Estimation Algorithm)^[11]类似于经典切诺夫算法,与切诺夫算法不同的是它需要设置在每一步中使用的模拟预算,即算法运行时每次迭代过程中调度执行的模拟总数,这种模拟预算大于经典切诺夫界。它计算最大和最小概率,可用于验证概率与阈值的大小关系。该算法的输出结果是最佳调度的估计概率、算法执行的步骤数量和模拟的总数,克服了固定抽样方案

中每次抽样样本数量相同的缺点。

2.5 智能假设检验算法

本文采用的智能假设检验算法(Smart Hypothesis Testing Algorithm)^[11]类似于序贯算法,与序贯算法不同的是需要设置模拟预算,即算法运行时每次迭代过程中调度执行的模拟总数。在该算法中,使用概率的阈值来直接计算模拟预算的初始分配。后面步骤与智能概率估计算法类似,即为每次调度执行已分配的模拟数量,运行时保持在最优调度的模拟数量的一半。如果有调度满足假设的条件,该算法立即终止并报告该假设是满足的。如果最优调度都不满足假设条件,该算法终止并报告结果,它同样克服了固定抽样方案中每次抽样样本数量相同的缺点。

3 实验对象选择

实验对象是无线局域网协议和哲学家就餐问题,它们是计算机领域的经典问题,验证性质是在特定条件下系统模型状态的可达性。

3.1 无线局域网协议验证

无线局域网协议(Wireless Local Area Networks Protocol)^[13]是指利用无线通信技术在一定的局部范围内建立的网络,它以无线多址信道作为传输媒介,能够使用户真正实现随时、随地、随意地进行宽带网络接入。WLAN 已在军队、石化、医护管理、工厂车间、库存控制、展览和会议、金融服务、旅游服务、移动办公系统等行业中得到了应用,是无线通信和互联网技术相结合的发展方向。WLAN 的最大优点是实现了网络互连的可移动性,它能大幅提高用户访问信息的及时性和有效性,还可克服线缆限制引起的不便性,但是性能受障碍物的影响较大。

本文采用了 IEEE802.11 无线局域网协议的概率时间自动机模型。该模型代表了协议的介质访问控制方案中的双向握手机制。它包括两个站点间的碰撞冲突,即同时发送消息,利用随机指数回退的过程。模型的时间约束类似于跳频扩频技术(FHSS)物理层,以50为倍数(即1时间单位等于50 μ s)来降低复杂性。采用整数语义的模型包括并行操作的3个组件,分为是管道(chan)和发送器(sender1 和 sender2),重要参数为最大冲突数量(COL,数值为2)。验证性质描述如表1所列。

表1 WLAN 模型验证性质

性质	公式描述	文字描述
collisions	$F \leq \# 100 \text{ col} = \text{COL}$	从当前状态开始,在100步之内,冲突数量最终会达到最大冲突数量
mean	$X \leq \# 1000 \text{ s1}$	从当前状态开始,s1最终会出现在下一状态

3.2 哲学家就餐问题验证

哲学家就餐(Dining Philosophers)问题^[14]是计算机科学中的一个经典问题,托尼·霍尔重新表述计算机科学家艾兹格·迪科斯彻提出的5台计算机试图访问5份共享的磁带驱动器的同步问题,称之为哲学家问题。其主要用于演示并行计算中多线程同步时产生的问题,也可以用来解释死锁和资源耗尽。

在此例中,若干哲学家坐在一张圆桌上,桌上有与哲学家相同数量的碗和筷子,在每人两边各放一只筷子。哲学家平时处于思考状态,在饥饿时便拿起左手和右手边的筷子,只有在他拿到两支筷子时才能进餐,进餐完毕后,把筷子放回原位继续思考。所以哲学家进餐的条件是:

1)拿到两只筷子时哲学家才开始吃饭。

2)如果筷子已在他人手上,则该哲学家必须等他人吃完之后才能拿到筷子。

3)哲学家在自己未拿到两只筷子前不放下自己手中已拿到的筷子。

本文采用允许哲学家进行随机概率性选择的模型,主要参数为哲学家的个数(Philo,数值分别是 3,30,300)及资源个数(与哲学家个数对应相等),饥饿状态(hungry)和就餐状态(eat)。验证性质描述如表 2 所列。

表 2 哲学家就餐问题验证性质

性质	公式描述	文字描述
liveness	$F \leq \# 1000 ("hungry") \& (X \leq \# 1 F \leq \# 1000 "eat")$	最终会有一个哲学家饥饿,并且在此之后一个哲学家可以就餐
eat_2	$F \leq \# 31 p2 > 8$	从当前状态始,在 31 步之内,标号为 2 的哲学家开始就餐

4 实验设计

本文采用无线局域网协议模型和哲学家就餐模型以及 PLASMA 模型检测工具,分析比较切诺夫算法、序贯算法、智能概率估计算法、智能假设检验算法和蒙特卡罗算法的运行时间,从而分析算法在一定范围内的适用性。本文所采用的实验模型和验证性质都来源于 PLASMA 工具网站^[15]。

实验环境:Win10 版本的 64 位操作系统,AMD Athon(tm)X4 740 Quad Core 处理器,内存为 4GB。

实验工具:模型检测工具 PLASMA Lab^[15]。PLASMA 工具由 java 语言编写,提供了图形化界面,允许用户编辑模型和性质以运行模拟实验。工具主要分为模拟器、检查器和算法设置 3 个部分,其中模拟器主要用于执行模型并生成路径,检查器用于检测路径中性质的正确性,算法设置决定了实验条件是否满足算法的可信度和精度,最后计算出运行时间。

实验对象:无线局域网协议模型、collisions 性质和 mean 性质;3 个、30 个、300 个哲学家就餐问题的模型及其相应的 liveness 和 eat_2 性质。所有实验的输入是待验证模型和性质,输出为对应条件下的算法执行时间。

实验参数设置如下:

1)切诺夫算法,实现在 PLASMA 中的“Chernoff”算法选项中。参数设置为:误差 1%、置信度 99%(即 $\epsilon=0.01$, $\delta=0.01$)及误差 5%、置信度 95%(即 $\epsilon=0.05$, $\delta=0.05$)。

2)序贯算法,实现在 PLASMA 中的“Sequential”算法选项中。参数设置为:误差 1%、置信度 99%(即 $\alpha=0.01$, $\delta=0.01$, $\beta=0.01$, $\text{proba}=0.5$),及误差 5%、置信度 95%(即 $\alpha=0.05$, $\delta=0.05$, $\beta=0.01$, $\text{proba}=0.5$)。

3)智能概率估计算法,实现在 PLASMA 中的“Chernoff ND”算法选项中。除了增加模拟预算 $\text{budget}=100000$,运行时选择计算最小概率(Minimum)和最大概率(Maximum)外,其他参数的设置与切诺夫算法相同。

4)智能假设检测算法,实现在 PLASMA 的“Sequential ND”算法选项中。除了增加模拟预算 $\text{budget}=100000$,运行时选择计算最小概率(Minimum)和最大概率(Maximum)外,其他参数的设置与序贯算法相同。

5)以上参数设置可保证算法在误差和置信度上相同,从而比较算法的运行时间,但每种算法在运行时会产生不同的模拟调度数量。所以蒙特卡罗算法的参数设置(total samples)为其他算法在每次运行中产生的模拟次数,即可与其他算法的运行时间进行对比。

实验步骤:

1)打开 PLASMA 工具,导入待验证模型和性质;

2)在工具界面中选择要执行的算法(Chernoff, Sequential, Chernoff ND, Sequential ND 和 MonteCarlo,其中 Chernoff ND 与 Sequential ND 有计算最小概率 minimum 或最大概率 maximum 选项),并设置相应的实验参数;

3)点击运行算法,等待实验结束,记录运行时间;

4)实验重复 30 次操作,计算平均值;

5)导入下一个待验证模型和性质,继续以上操作。

5 实验结果及其分析

导入待验证模型 WLAN 中的 collisions 性质,各个算法的执行时间如表 3 所列。

表 3 collisions 性质验证时间/s

collisions	Chernoff	Sequential	ChernoffND min	Sequential ND min	Chernoff ND max	Sequential ND max
参数(0.01)	56.0846	10.2905	465.64	10.297	2822.535	1422.237
参数(0.05)	20.0535	10.5222	464.907	10.78	2025.488	1022.611

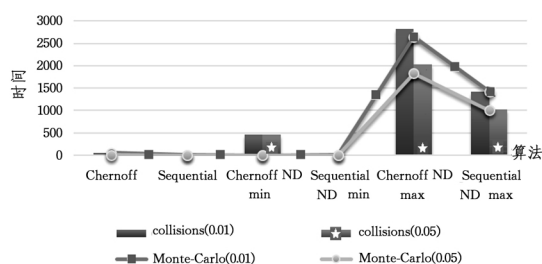


图 1 collisions 性质验证时间

图 1 中左边柱的参数设置为误差 1%、置信度 99%,右边柱的参数设置为误差 5%、置信度 95%。高折线与低折线是其他算法在误差为 1%与 5%时产生的模拟数量下的 Monte-Carlo 算法的运行时间。数据显示,Chernoff ND 算法与 Chernoff 算法的时间相差很大,主要原因是算法的原理不同,

一种算法在后一种算法的基础上在运行期间进行了多次迭代,Sequential ND 算法与 Sequential 算法与之相同。

Chernoff 算法的运行时间在参数为 0.01 时比 Sequential 算法的运行时间长 50 多秒,而由于误差增加及置信度降低,导致模拟数量减小,所以在参数为 0.05 时,Chernoff 算法的时间长有所下降。在计算最小概率时,Chernoff ND 的运行时间比 Sequential 算法的运行时间长 460 多秒。在计算最大概率时,Chernoff ND 的运行时间比 Sequential 算法的运行时间长 1000 秒左右。MonteCarlo 算法的运行时间比 Chernoff ND max 与 Sequential ND max 的少 200 秒左右,而与其他算法差别不大。所以,序贯算法及蒙特卡罗算法可用于验证此类性质,且在计算满足条件的最大概率时会花费较多时间。

导入待验证模型 WLAN 中的 mean 性质,各个算法的执行时间如表 4 所列。

表 4 mean 性质验证时间/s

mean	Chernoff	Sequential	Chernoff ND min	Sequential ND min	Chernoff ND max	Sequential ND max
参数(0.01)	468.9167	20.2337	26698.747	15226.54	—	20.63
参数(0.05)	130.6454	10.4235	20343.171	10318.973	—	10.422

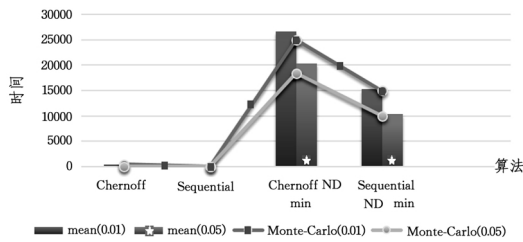


图 2 mean 性质验证时间

图 2 中的指标名称含义同 collisions 性质。Chernoff 算法的运行时间在参数为 0.01 时比 Sequential 算法的运行时

间长 330 多秒,而由于误差增加及置信度降低,导致模拟数量减小,因此在参数为 0.05 时,Chernoff 算法的时间大幅下降。在计算最小概率时,Chernoff ND 的运行时间比 Sequential 算法的运行时间长 10000 秒左右。在计算最大概率时,Chernoff ND 的运行时间已大于软件所能承受的范围。MonteCarlo 算法的运行时间比 Chernoff ND min 与 Sequential ND min 少 1000 秒左右,而与其他算法差别不大。所以,切诺夫算法不适合验证此类性质。

导入待验证模型 Dining Philosophers 中的 liveness 性质,各个算法的执行时间如表 5 所列。

表 5 liveness 性质验证时间/s

liveness(0.01)	Chernoff	Sequential	Chernoff ND min	Sequential ND min	Chernoff ND max	Sequential ND max
p3(0.01)	11.7829	10.283	79.8771	45.5	81.4204	10.16
p30(0.01)	24.9829	10.936	873.548	497.64	845.877	10.94
p300(0.01)	821.533	13.6629	51128.27	24272.125	44936.335	24.15
liveness(0.05)	Chernoff	Sequential	Chernoff ND min	Sequential ND min	Chernoff ND max	Sequential ND max
p3(0.05)	10.7217	10.172	58.437	31.453	58.953	10.16
p30(0.05)	11.2124	10.345	657.517	336.766	652.798	10.31
p300(0.05)	22.9766	10.849	37520.712	19027.817	37286.821	10.906

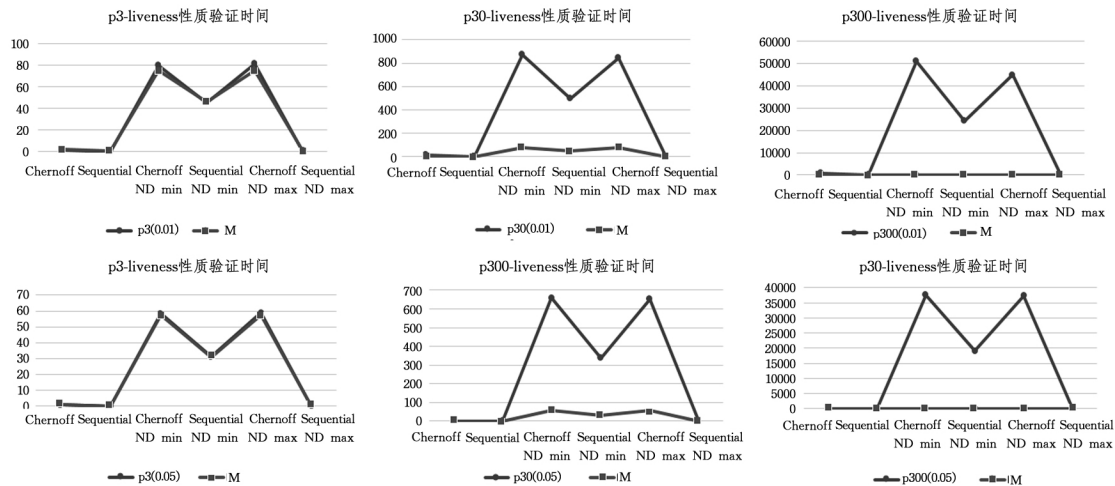


图 3 各参数条件下 liveness 性质验证时间

表 5 和图 3 中的指标名称含义同 collisions 性质,图表中 $p_x(0.01)$ 或 $p_x(0.05)$ 代表哲学家个数为 x , 参数设置条件为误差 1% 或误差 5%。由表可看出,Chernoff 算法的运行时间比 Sequential 算法的运行时间长,且在参数为 0.05 时验证时间减少。在计算最小概率时,Chernoff ND 的运行时间长于 Sequential 算法的运行时间,在哲学家数量为 300 时甚至有 20000 多秒的差距。在计算最大概率时,Sequential ND 算法的时间优势明显。由同一行的不同哲学家个数模型时间图可

看出,在设置同样参数的情况下,当哲学家增多时,各个算法的运行时间曲线(包含最高峰值的曲线,类似于 M 形状)相似,但当哲学家个数增多时,模型增大,运行时间明显增加。由同列的图可看出,当哲学家个数相同,即模型大小相同时,误差越小,算法运行时间越长。在 3 个哲学家模型中,蒙特卡罗算法的运行时间与其他算法相近,且运行时间没有太大变化。

导入待验证模型 Dining Philosophers 中的 eat_2 性质,各个算法的执行时间如表 6 所列。

表 6 eat_2 性质验证时间/s

eat_2(0.01)	Chernoff	Sequential	Chernoff ND min	Sequential ND min	Chernoff ND max	Sequential ND max
p3(0.01)	13.4406	10.515	258.141	22.78	158.25	10.47
p30(0.01)	23.4017	10.813	82.375	10.94	798.86	49.72
p300(0.01)	102.8847	10.573	627.480	10.61	3311.70	391.23
eat_2(0.05)	Chernoff	Sequential	Chernoff ND min	Sequential ND min	Chernoff ND max	Sequential ND max
p3(0.05)	10.8046	10.217	31.781	11.64	116.625	10.31
p30(0.05)	11.2078	10.261	78.531	10.31	613.970	51.41
p300(0.05)	30.4358	10.190	619.189	10.39	2792.896	410.42

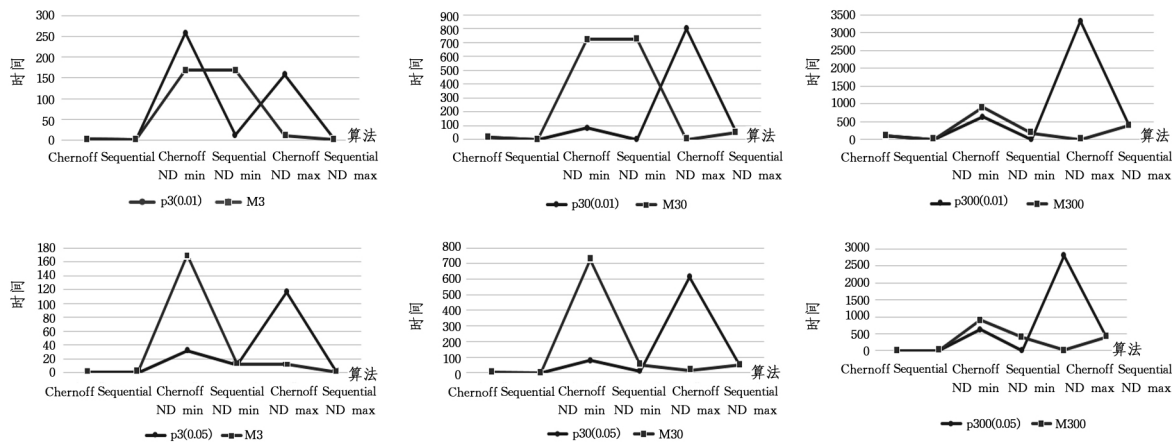
图 4 各参数条件下 eat₂ 性质验证时间

表 6 和图 4 中的指标名称含义同 liveness 性质。由表 6 可看出, Sequential 算法的运行时间比 Chernoff 短, Chernoff ND 与 Sequential ND 算法受误差及置信度的影响较小。在计算最小概率时, Chernoff ND 的运行时间长于 Sequential 算法的运行时间。在计算最大概率时, Sequential ND 算法的时间优势明显。由同一行的哲学家个数为 3, 30, 300 的模型的时间图可看出, 在同样的参数设置下, 当哲学家增多时, 各个算法的运行时间曲线(包含最高峰值的曲线)的峰值出现在 Chernoff ND min 与 Chernoff ND max, 表示此算法时间花费较高; 同时, 代表时间的纵坐标有明显增加, 表示算法运行时间会随着模型的增大而增加。由同列的图可看出, 在 3 个哲学家模型中, 误差越小, 算法运行时间越长, 在其他模型中差别不大。而在 300 个哲学家的模型中, 由于模拟数量大, 蒙特卡罗算法表现出优势。

算法适用情况如表 7 所列。

表 7 算法适用情况

算法	数据抽样	时间性能	适用情况
切诺夫算法	给定值	长于序贯算法	受误差范围变化较大, 性能表现适中, 适用于离散独立变量, 验证规模适中模型
序贯算法	逐次取样	最短	适用于变形观测的数据处理, 适用于验证规模适中模型
智能概率估计算法	迭代抽样	最长	抽样有效概率精准, 不适用于计算复杂模型的最小概率
智能假设检验算法	迭代抽样	短于智能概率估计算法	抽样有效概率精准, 适用于验证可达性性质的最大概率
蒙特卡罗算法	随机取样	受模拟数量影响大	简单快速, 适用于验证规模较大的模型, 不适用于低精度模型

6 结论

在一定程度上表明, 切诺夫算法可以评估算法的鲁棒性水平, 但在算法性能上表现一般, 序贯算法的时间性能更具优势。序贯算法逐次抽样的思想适合于验证模型可达性性质。智能概率估计算法与智能假设检验算法采用迭代抽样, 其中智能概率估计算法的抽样次数需要大于切诺夫界的最高值, 所以它们的运行时间比切诺夫算法与序贯算法更长。智能概率估计算法在验证性质时的整体性能差于智能假设检验算法, 模型越大越复杂, 特别是在验证模型可达性性质时, 智能假设检验算法更具优势。蒙特卡罗算法不适合低精度的评估, 模型复杂性的增加不会对收敛速度产生太大的影响, 所以该算法适用于处理复杂问题。

由于只选取了统计模型检测中的部分算法及两种类型的模型及性质, 因此没有涵盖统计模型检测领域所有可用算法及可应用模型及性质。虽然采取了多次实验取平均值等措施, 但实验执行环境因素等影响还是会造成实验误差, 这是本文的局限性。统计模型检测算法的优点是效率高, 对于复杂系统而言, 生成系统模拟路径的速度更快; 但是面临的问题是在可接受的样本数量下, 难以预测和评估小概率事件发生的概率, 且随着抽样数量的增加, 大大降低了算法效率。在未来工作中, 可以考虑更多算法及模型来对统计模型检测做更全面的分析。

结束语 文中主要研究 5 种统计算法在统计模型检测工具 Plasma 中的时间性能。实验模型采用被抽象的无线局域网模型和哲学家问题模型, 验证模型中的特定性质是否满足。统计分析每个算法在不同的实验设定条件下的运行时间, 从而分析统计算法对统计模型检测的时间效率影响, 有助于在模型检测时对统计算法的选择, 以提高模型检测的效率。

参考文献

- [1] YOUNES H L S, SIMMONS R G. Statistical probabilistic model checking with a focus on time-bounded properties[J]. Information and Computation (JIC), 2006, 204(9): 1368-1409.
- [2] ZULIANI P, PLATZER A, CLARKE E M. Bayesian statistical model checking with application to Stateflow/Simulink verification[J]. Formal Methods in System Design, 2010, 43(2): 243-252.
- [3] YOUNES H L S, KWIATKOWSKA M, NORMAN G, et al. Numerical vs. statistical probabilistic model checking[J]. Software Tools for Technology Transfer (STTT), 2006, 8(3): 216-228.
- [4] HENRIQUES D, MARTINS J G, ZULIANI P, et al. Statistical Model Checking for Markov Decision Processes[C] // International Conference on Quantitative Evaluation of Systems, 2012: 84-93.
- [5] 代声馨, 洪玫, 郭兵, 等. 多处理器实时系统可调度性分析的 UP-PAAL 模型[J]. 软件学报, 2015, 26(2): 279-296.
- [6] DAVID A, LARSEN K G, LEGAY A, et al. Schedulability of Herschel revisited using statistical model checking[J]. International Journal on Software Tools for Technology Transfer, 2012, 17(2): 293-307.
- [7] HASTINGS W K. Monte Carlo sampling methods using Markov chains and their applications[J]. Biometrika, 1970, 57(1): 97-109.

(下转第 533 页)

```

{
    int i;
    cout <<“该项测试用于对核心初始化/退出次的性能作出评估”<< endl;
    for (i=0;i<100;i++)
    {
        ASSERT_TRUE(SrvCoreInit());
        ASSERT_TRUE(SrvCoreExit());
    }
}

```

2)对 IOlock 单线程获取/释放 50000000 次的性能作出评估:

```

TEST (Core,OneThreadIOlockPerformance)
{
    cout <<“该项测试用于对 IOlock 单线程获取/释放 50000000 次的性能作出评估”<< endl;
    IOlock lock;
    int i;
    SrvCoreInitIOlock(&lock);
    for (i=0;i<50000000;i++)
    {
        SrvCoreEnterIOlock(&lock);
        SrvCoreLeaveIOlock(&lock);
    }
}

```

3)用于对 IOlock 双线程各获取/释放 25000000 次的性能作出评估:

```

DWORD WINAPI TwoThreadIOlockPerformance (LPVOID lpParameter)
{
    int i;
    IOlock *plock=(IOlock *)lpParameter;
    for (i=0;i<25000000;i++)
    {
        SrvCoreEnterIOlock(plock);
        SrvCoreLeaveIOlock(plock);
    }
    return 0;
}
TEST (Core,TwoThreadIOlockPerformance)
{
    cout <<“该项测试用于对 IOlock 双线程各获取/释放 000000 次的性能作出评估”<< endl;
    IOlock *plock=new IOlock;
    int i;

```

```

HANDLE hThread;
SrvCoreInitIOlock(plock);
hThread = CreateThread (NULL, 0, TwoThreadIOlockPerformance,plock,0,NULL);
for (i=0;i<25000000;i++)
{
    SrvCoreEnterIOlock(plock);
    SrvCoreLeaveIOlock(plock);
}
WaitForSingleObject(hThread,INFINITE);
CloseHandle(hThread);
delete plock;
}

```

结束语 本套利交易系统通过“傻瓜式”交易,不但使得没有同时掌握计算机技术和丰富金融知识的普通投资者拥有了更自由开放的平台,也让程序化交易策略变得更简单易懂、方便快捷。通过引入泛化的概念,实现了全球两两标的资产之间的套利模式,为投资者们提供了更大的自由度和更好的用户体验。然而,套利的盈利能力往往在一定程度上取决于系统的速度。据调查发现,拥有世界上最先进交易系统的 Citadel 公司在不到 3 分钟的时间内竟可以下达高达 2100 万股的交易指令,类似国外的优秀高频交易团队的一笔交易的时间大多在微妙级别,而本文系统使用外盘接口 SPTrader 时仅花在大陆到香港的网络延迟就在毫秒级别。换言之,使用 SPTrader 的一笔交易所花的时间对于优秀的交易团队而言可以完成上千笔交易,导致即使出现好的交易机会也因为比别人慢而抓不住。故在下一步工作中,将围绕因外盘接口 SPTrader 在交易速度上有所局限而降低了套利交易的盈利能力这一方面进行研究与改善。

参 考 文 献

- [1] 高婷婷. 基于程序化的 ETF 套利交易系统的设计与实现[D]. 黑龙江:黑龙江大学,2013.
- [2] 高翔. 中国股指期货市场跨期套利策略的分析与实践[D]. 成都:西南财经大学,2012.
- [3] 王宏伟. 股指期货套期保值和套利策略分析[D]. 北京:中国科学院研究生院,2013.
- [4] 丁超群. 期货程序化套利平台研究与实现[D]. 上海:上海交通大学,2014.
- [5] BIALKOWSKI J. Stock Index Future Arbitrage in Emerging Markets.Polish Evidence[J]. International Review of Financial Analysis,2008,17(2):363-381.
- [6] 吕鸿杰. 基于 SOA 技术框架数据分析系统的研究与实践[D]. 成都:西南交通大学,2008.
- [7] (上接第 503 页)
- [8] BUCKLEW J. A,Sadowsky,et al. A Contribution To The Theory Of Chernoff Bounds[J]. IEEE Transactions on Information Theory,1991,39(1):75-75.
- [9] 顾雅娟,仰枫帆. 基于修正的切尔诺夫界的尾部概率的估计[C]//全国青年通信学术会议. 2010.
- [10] MALLOY M,NOWAK R. Sequential analysis in high-dimensional multiple testing and sparse recovery[J]. Mathematics, 2011,42(4):2661-2665.
- [11] http://plasma-lab.gforge.inria.fr/plasma_lab_doc/1.4.0/html/algorithms/mdp.html#smart-sampling-algorithms.
- [12] KIM Y,KIM M,KIM T H. Statistical Model Checking for Safety Critical Hybrid Systems:An Empirical Evaluation[M]. Hardware and Software:Verification and Testing. 2013:162-177.
- [13] KWIATKOWSKA M,NORMAN G,SPROTON J. Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol[M]. Process Algebra and Probabilistic Methods: Performance Modeling and Verification. Springer Berlin Heidelberg,2002:411-423.
- [14] PNUELI A,ZUCK L D. Probabilistic Verification[J]. Information & Computation,1993,103(1):1-29.
- [15] <https://project.inria.fr/plasma-lab>.