软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(5):1128−1143 [doi: 10.13328/j.cnki.jos.005216]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种面向 CPS 的自适应统计模型检测方法*

杜德慧[1,2,3], 咎 慧[3], 姜凯强[3], 程 贝[3]

[1](教育部可信软件国际合作联合实验室(华东师范大学),上海 200062)

[2](可信软件国际联合研究中心(华东师范大学),上海 200062)

[3](上海市高可信重点实验室(华东师范大学),上海 200062)

通讯作者: 杜德慧, E-mail: dhdu@sei.ecnu.edu.cn

摘 要: 随着计算机与物理环境的交互日益密切,信息-物理融合系统(cyber-physical system,简称 CPS)在健康医疗、航空电子、智能建筑等领域具有广泛的应用前景,CPS 的正确性、可靠性分析已引起人们的广泛关注.统计模型检测(statistical model checking,简称 SMC)技术能够对 CPS 进行有效验证,并为系统的性能提供定量评估.然而,随着系统规模的日益扩大,如何提高统计模型检测技术验证 CPS 的效率,是目前所面临的主要困难之一.针对此问题,首先对现有 SMC 技术进行实验分析,总结各种 SMC 技术的受限适用范围和性能缺陷,并针对贝叶斯区间估计算法(Bayesian interval estimate,简称 BIE)在实际概率接近 0.5 时需要大量路径才能完成验证的缺陷,提出一种基于抽象和学习的统计模型检测方法 AL-SMC.该方法采用主成分分析、前缀树约减等技术对仿真路径进行学习和抽象,以减少样本空间;然后,提出了一个面向 CPS 的自适应 SMC 算法框架,可根据不同的概率区间自动选择 AL-SMC 算法或者 BIE 算法,有效应对不同情况下的验证问题;最后,结合经典案例进行实验分析,实验结果表明,自适应 SMC 算法框架能够在一定误差范围内有效提高 CPS 统计模型检测的效率,为 CPS 的分析验证提供了一种有效的途径.

关键词: 信息-物理融合系统;统计模型检测;抽象;学习;自适应性

中图法分类号: TP311

中文引用格式: 杜德慧,咎慧,姜凯强,程贝.一种面向 CPS 的自适应统计模型检测方法.软件学报,2017,28(5):1128−1143. http://www.jos.org.cn/1000-9825/5216.htm

英文引用格式: Du DH, Zan H, Jiang KQ, Cheng B. Self-Adaptive statistical model checking approach for CPS. Ruan Jian Xue Bao/Journal of Software, 2017,28(5):1128−1143 (in Chinese). http://www.jos.org.cn/1000-9825/5216.htm

Self-Adaptive Statistical Model Checking Approach for CPS

DU De-Hui[1,2,3], ZAN Hui[3], JIANG Kai-Qiang[3], CHENG Bei[3]

[1](MOE International Joint Laboratory of Trustworthy Software (East China Normal University), Shanghai 200062, China)

[2](International Joint Research Center of Trustworthy Software (East China Normal University), Shanghai 200062, China)

[3](Shanghai Key Laboratory of Trustworthy Computing (East China Normal University), Shanghai 200062, China)

Abstract: Cyber-Physical systems (CPSs) are advanced embedded systems engaging more interaction between computer and physical environment. CPSs are widely used in the field of healthcare equipment, avionics, and smart building. Meanwhile, the correctness and reliability analysis of CPSs has attracted more and more attentions. Statistical model checking (SMC) is an effective technology for verifying CPSs, which facilitates the quantitative evaluation for system performance. However, it is still a challenge to improve the performance of SMC with the expansion of systems. To address this issue, this study explores several SMC algorithms and concludes that

* 基金项目: 国家自然科学基金(61472140, 61170084); 上海市自然科学基金(14ZR1412500)

Foundation item: National Natural Science Foundation of China (61472140, 61170084); Natural Science Foundation of Shanghai (14ZR1412500)

收稿时间: 2016-07-18; 修改时间: 2016-09-25; 采用时间: 2016-12-07; jos 在线出版时间: 2017-01-20

CNKI 网络优先出版: 2017-01-20 16:06:42, http://www.cnki.net/kcms/detail/11.2560.TP.20170120.1606.019.html

Bayesian interval estimate is the most practical and efficient algorithm. However, large scale of traces are needed when the actual probability is around 0.5 during the evaluation. To overcome this difficulty, an algorithm, AL-SMC is proposed based on abstraction and learning techniques to reduce the size of sampling space. AL-SMC adopts some sophisticated techniques such as property-based projection, extraction and construction of prefix frequency tree. In addition, to improve the efficiency of SMC further, a framework of self-adaptive SMC algorithm, which uses the proper algorithm by probability prediction adaptively, is presented. Finally, the self-adaptive SMC approach is implemented with three benchmarks. The experimental results show that the proposed approach can improve the performance within an acceptable error range.

Key words: cyber-physical system; statistical model checking; abstract; learning; self-adaptive

信息-物理融合系统(cyber-physical system,简称 CPS)[1]是一类综合计算、网络和物理环境的多维复杂系统,在健康医疗、智能交通、航空电子、智能建筑等领域有着广泛的应用前景.与传统系统不同,CPS 是一个交叉领域学科,除计算机外,还融合了机械、环境、土木、电子、生物、化学、航空等诸多工程领域的模型和方法.CPS 是一种运行在开放环境中的复杂异构系统,存在大量不确定性因素,如运行环境、信号误差、人的行为等.CPS 具有以下两个特点:1) CPS 的应用大都安全攸关或功耗要求严苛,在保证系统功能正确的前提下,还需要满足非功能性需求约束,如吞吐量、能耗、时间等;2) CPS 是异构的混成系统,融合了连续的物理过程和离散的系统行为,且其运行的开放环境具有不确定性、随机性.因此,如何使用现有的模型检测技术分析验证 CPS 的正确性及量化评估系统性能指标,是当前面临的挑战.

传统的模型检测技术通过对系统的状态空间进行全遍历,能够自动验证、分析系统行为的安全性、可靠性,但是它存在状态空间爆炸的问题(即随着系统规模的增加,系统的状态空间呈指数级增长).随着系统规模的扩大,传统的模型检测算法[2−4]很难验证、分析复杂系统行为的正确性、可靠性,也无法进行定量评估分析.因此,使用传统的模型检测技术分析验证 CPS 的可靠性、定量评估系统的性能面临着巨大困难.统计模型检测技术(statistical model checking,简称 SMC)[5]能够有效评估系统模型满足目标约束的概率区间,可用于定量分析系统模型的性能指标,能够有效缓解状态空间爆炸问题.统计模型检测技术主要使用统计方法分析复杂系统的模拟执行路径,优点是效率高、执行速度快,该项技术的关键点在于:首先生成系统的模拟路径,并对其进行伯努利实验,判断模拟路径是否满足给定的系统约束;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,评估系统满足约束的概率区间.对于大型、复杂的系统而言,生成系统模拟路径要更快、更容易,而不需要将系统模型转换为验证工具的输入语言.因此,统计模型检测技术为大型、复杂的 CPS 的正确性、可靠性分析和性能评估提供了广阔的研究前景.目前,SMC 技术已引起学术界和工业界的广泛关注,成功应用于分析实时调度[6]、生物系统[7]、能耗感知的智能家居(energy-aware building)[8,9]等领域.

虽然 SMC 无需遍历状态空间,但当结果的精度要求较高或者检测小概率事件时,仍需生成大量系统执行路径才能使统计算法收敛.随着系统规模的增加,路径的生成和验证十分耗时.因此,如何有效减少 SMC 收敛所需的路径数量,提供一个高效实用的 SMC 验证方案,是对 CPS 进行有效验证的重要问题.针对该问题,我们的工作贡献主要包括以下 3 个方面.首先,通过对已有 SMC 算法的原理分析和实验比较,得出贝叶斯区间估计算法是实用性和效率最好的算法.但其存在实际概率趋于 0.5 时所需路径数量剧增的问题,为弥补贝叶斯区间估计算法的不足,我们提出了基于抽象和学习的统计模型检测方法,旨在减少统计分析所需的路径数量,提高 SMC 的效率.其次,我们提出了一种自适应的 SMC 算法框架,可根据系统满足目标约束的概率估值,动态地在 BIE 和基于抽象和学习的统计模型检测算法之间选择合适的算法.该算法框架为 CPS 的验证问题提供了一种效率较高的验证途径.最后,我们通过几个典型的 CPS 案例对单纯的 BIE 算法和自适应 SMC 算法进行对比,实验数据表明,使用自适应 SMC 算法框架能够有效提高 SMC 的效率,并将误差控制在一定范围内.

1 现有统计模型检测算法的分析比较

统计模型检测技术建立在蒙特卡洛模拟(Monte Carlo)技术之上,它能够有效评估系统模型满足目标约束的概率区间,对系统进行验证分析.SMC 算法主要分为定性和定量两类,定性类算法包括 3 种:Single Sampling

Plan(SSP),Sequential Probability Ratio Test(SPRT)和 Bayesian Hypothesis Testing(BHT),用以验证“系统 S 满足 BLTL 约束 ϕ' 的概率是否大于或等于某个概率阈值 $\theta(\theta\in(0,1))$”这类问题,即 $S\models P_{\geqslant\theta}(\phi')$;定量类算法包括 Approximate Probabilistic Model Checking(APMC)和 Bayesian Interval Estimation(BIE),用以解决“系统 S 满足约束 ϕ' 的概率是多少”这类问题,即 $S\models P_{=?}(\phi')$.不同类型的 SMC 算法的主要区别在于统计参数的计算和置信度满足条件的判断上,由于 SSP 算法可用性不高,且与 SPRT 算法的原理类似,下面我们主要针对统计参数的计算和置信度满足的判断条件两部分,对其余 4 种 SMC 算法进行简要的原理分析和性能评估.

1.1 4种SMC算法的原理分析

定性类算法都基于假设检验.通过随机取样并检验原假设(即被检验的假设)H_0:$P\geqslant\theta$ 与备择假设 H_1:$P<\theta$(P 是 S 满足 ϕ' 的真实概率)来判断某条路径是接受 H_0 还是拒绝 H_0.这种方法并不能保证结论完全正确,只能通过参数限定来减少犯错的机率.

定量算法可以直接返回 S 满足 ϕ' 的概率 p,并将误差限定在某一范围内.APMC 和 BIE 两种算法都可被看作返回一个概率区间,即($p-\delta$,$p+\delta$),但两种算法计算获得 p 和 δ 的方式不同.

1.1.1 SPRT 算法

最初 Wald 提出的 SPRT 方法[10]基于简单假设(如 H:p=0.8),在错误率同等的条件下,最小化所需样本总数,此时,SPRT 可达到最优.然而 SMC 所验证的问题都是基于复合假设(如 H:$p\geqslant$0.8)的,因此,Younes 等人将 SPRT 扩展到复合假设检验[11],引入无差异区域(p_1,p_0),但此时的 SPRT 已无法达到最优.SPRT 需要选择两个参数 A 和 B($A>B$)来保证检验强度,并计算概率比,即

$$\frac{p_{1n}}{p_{0n}}=\prod\frac{\Pr(B_i=b_i\mid p=p_1)}{\Pr(B_i=b_i\mid p=p_0)}=\frac{p_1^x(1-p_1)^{n-x}}{p_0^x(1-p_0)^{n-x}} \tag{1}$$

当 $\frac{p_{1n}}{p_{0n}}>A$ 时接受 H_0 假设,当 $\frac{p_{1n}}{p_{0n}}<B$ 时接受 H_1 假设.当 H_0 和 H_1 其中之一被接受时,SPRT 算法终止[9].

1.1.2 BHT 算法

基于贝叶斯的假设检验方法 BHT[12]同样利用一个概率比值来衡量 H_0 和 H_1 的相对倾向程度,称为贝叶斯因子(Bayes factor),它最早由 Jeffreys[13]提出.对于原假设 H_0 和备择假设 H_1,贝叶斯因子 B 被定义为两者的似然比,它在数理统计中是一个能够同时反映灵敏度与特异度的综合指标,即

$$B=\frac{P(d_n\mid H_0)}{P(d_n\mid H_1)} \tag{2}$$

这里,d_n 表示样本路径集.在 BHT 算法中,我们通常还会预设一个常数阈值 T(要求 $T>1$),如果 $B>T$,则接受 H_0 假设;如果 $B<T$,则接受 H_1 假设.贝叶斯统计的难点在于复杂的条件概率计算,这可以利用共轭分布来简化计算.由于 SMC 所验证的参数 θ 是(0,1)上的值,所以适合使用 Beta 分布(定义在(0,1)上有两个参数 α,$\beta>0$ 的概率分布),Beta 概率密度函数表达如下:

$$g_{(\alpha,\beta)}(u)=\frac{1}{B(\alpha,\beta)}u^{\alpha-1}(1-u)^{\beta-1} \tag{3}$$

其中,$B(\alpha,\beta)$为 Beta 函数:

$$B(\alpha,\beta)=\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\mathrm{d}t \tag{4}$$

通过调整 Beta 函数,可以近似模拟许多(0,1)上的光滑单峰概率密度.特别地,当 $\alpha=\beta=1$ 时,其实是(0,1)上的均匀分布,通常被用作 BHT 计算的默认先验分布.

1.1.3 APMC 算法

最早由 Thomas 等人[14]提出的 APMC 算法通过计算 x/n(x 表示实际正样本数,n 表示总样本路径数)来获得概率 p,并通过 Chernoff-Hoeffding 界来保证其准确性.Hoeffding 在文献[12]中证明,通过统计获得的

$$p=x/n \tag{5}$$

与真实概率 P 之间满足 $\Pr(|P-p|>\varepsilon)<2e^{-\frac{N\varepsilon^2}{4}}$,变换为 APMC 所需的形式,即

$$\Pr(|P-p|\leqslant\varepsilon)\geqslant 1-\delta,\text{当}N\geqslant 4\log\left(\frac{2}{\delta}\right)\Big/\varepsilon^2 \tag{6}$$

此外,APMC 算法需要两个参数ε(近似参数),δ(置信参数)来保证结果准确性,并可以计算出所需的路径样本数量 n.

1.1.4 BIE 算法

BIE 算法[15]是一种基于贝叶斯统计理论的 SMC 算法.与 APMC 不同,BIE 算法不直接计算其值,而是寻找包含真实概率的区间,然后,取估值区间(假定为(t_0,t_1))的中间值作为概率 p,即 $p=t_0+x=t_1-x$.这里,x 是预设的半区间大小.

因此,BIE 算法的核心为区间估计.当给定一个目标阈值(区间覆盖系数 $c\in\left(\frac{1}{2},1\right)$)和一个区间$(t_0,t_1)$时,有:

$$\int_{t_0}^{t_1} f(u\mid d_n)\mathrm{d}u=c \tag{7}$$

由于区间(t_0,t_1)包含了真实概率 P,所以在满足公式(7)的条件下尽量缩小区间.然而,并非所有分布都能保证找到这样的最优区间,对于 Beta 分布这样的单峰概率密度,可取后验概率的均值作为所求区间的中心,再通过预设的δ得到区间端点.对于参数为α,β的 Beta 先验分布,后验概率的均值为 $p=(x+\alpha)/(n+\alpha+\beta)$,后验概率分布函数如下:

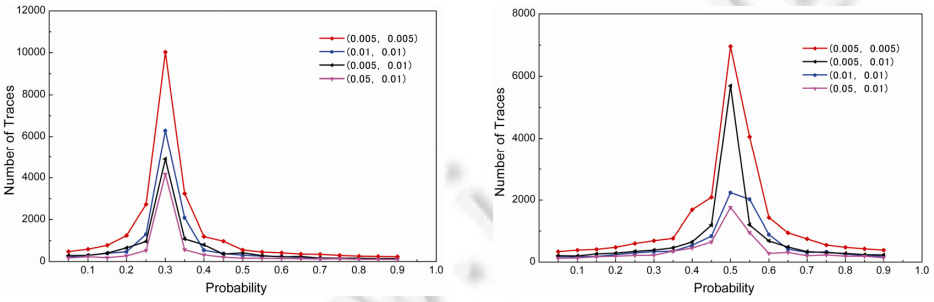
$$\int_{t_0}^{t_1} f(u\mid d_n)\mathrm{d}u=F_{x+\alpha,n-x+\beta}(t_1)-F_{x+\alpha,n-x+\beta}(t_0) \tag{8}$$

其中,d_n 表示样本路径集,$F(\cdot)$为 Beta 分布函数.

1.2 4种SMC算法的性能比较

SMC 算法的总体性能取决于仿真器和统计分析器的性能.对于大多数模型而言,路径的产生与验证耗时占 SMC 总体耗时的 90%左右[15].仿真器的性能直接决定了每条执行路径产生的效率,不同模型和工具差异较大,且与 SMC 的统计分析效率无关.我们将使用取随机数的方式来模拟不同的概率空间,将仿真器对 SMC 算法性能的影响降到最低,以准确评估 SMC 算法本身的性能。此外,我们还对比算法终止时所需的路径数量(可看作 SMC 算法对仿真器性能的依赖程度).下面将从算法所需的路径数和统计计算的耗时两个方面,对 4 种 SMC 算法进行性能比较.

图 1(a)和图 1(b)分别展示了真实概率 P 为 0.3 和 0.5 时,执行 100 次 SPRT 算法所需的平均路径数量.由图 1 可知:随着置信度(α,β)的提高,所需路径数会增加;并且概率阈值θ越接近真实概率,所需路径越多.



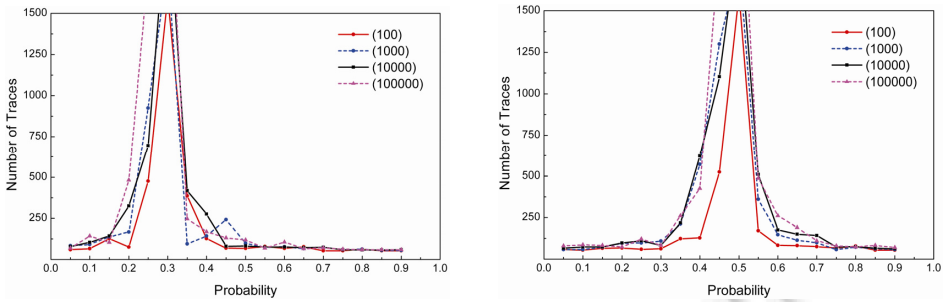
(a) 验证的真实概率 P 为 0.3　　(b) 验证的真实概率 P 为 0.5

Fig.1 Traces needed for SPRT algorithm at different confidence and probability threshold θ

图 1 不同置信度和概率阈值θ下,SPRT 算法所需的路径数量

图 2 展示了真实概率 P 为 0.3 和 0.5 时,执行 100 次 BHT 算法所需的平均路径数量.规律与 SPRT 类似:随着贝叶斯因子的可接受阈值 T 的增大,所需样本数量呈指数增长.实验中,当 BHT 算法的θ约等于真实概率时,

所需路径数接近正无穷.



(a) 验证的真实概率 P 为 0.3　　(b) 验证的真实概率 P 为 0.5

Fig.2　Traces needed for BHT algorithm at different confidence and probability threshold θ

图 2　不同置信度和概率阈值θ下,BHT 算法所需的路径数量

图 3 对比了 SPRT 与 BHT 在同一置信度下所需的路径数量(SPRT 的δ=0.01,BHT 的 T=10000).当θ远离真实概率时,BHT 算法所需路径数少于 SPRT 算法;但当θ接近真实概率时,BHT 所需路径数急剧增大,可能远大于 SPRT 所需路径数.

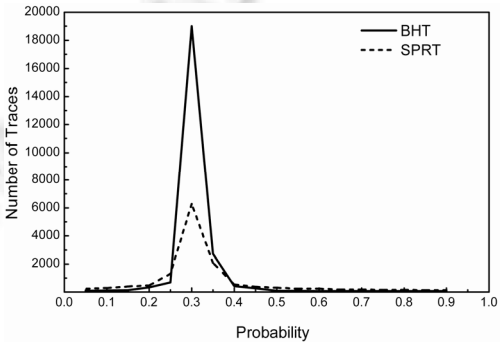
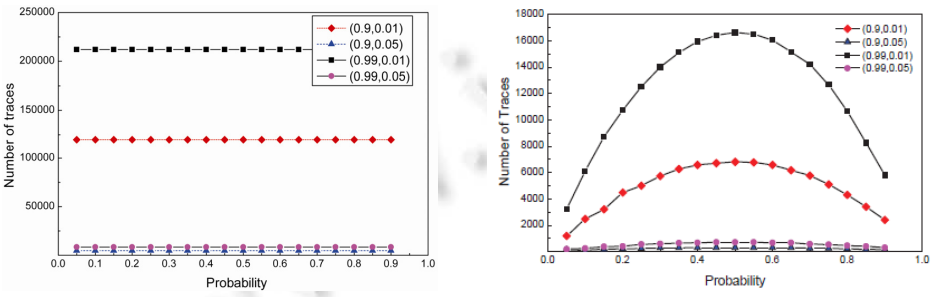


Fig.3　Traces needed for SPRT and BHT algorithm with the same confidence (δ=0.01,T=10000)

图 3　同一置信度下 SPRT 与 BHT 算法所需路径数量的对比(δ=0.01,T=10000)

图 4(a)展示了执行 100 次 APMC 算法所需的平均路径数量.如图 4 所示,APMC 在检验不同概率值时,所需路径数量相等;但随精度要求提高,所需路径数量也急剧上升.



(a) APMC　　(b) BIE

Fig.4　Traces needed for APMC and BIE algorithm at different probabilities under different accuracy

图 4　不同精度下,APMC 和 BIE 算法检验不同概率值所需的路径数量

图 4(b)展示了执行 100 次 BIE 算法所需的平均路径数量.由图可知,BIE 在真实概率 P 为 0.5 时所需路径最多,而接近 0 或 1 时所需路径数最少.对比发现,BIE 所需路径数在精度较高时明显小于 APMC 所需路径数(当δ=0.01,BIE 中的 c=0.9,即 APMC 中的 $1-\varepsilon$=0.9 时,BIE 所需最大路径数为 17 000,而 APMC 所需路径数为 212 000).

图 5 则更直观地比较了 APMC 与 BIE 在同一精度下所需的路径数量.

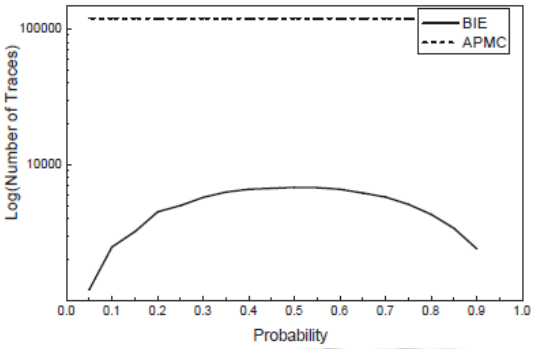


Fig.5 Traces needed for APMC and BIE algorithm with the same accuracy (δ=0.01,c=1−ε=0.9)

图 5 同精度下 APMC 与 BIE 算法所需路径数对比(δ=0.01,c=1−ε=0.9)

接下来分析比较 4 种 SMC 算法的耗时情况.表 1 为 4 种 SMC 算法在 1 000 次迭代下统计计算部分的耗时对比.其中,BHT 和 BIE 在不同积分步长下耗时差异显著,可见,积分是贝叶斯类 SMC 算法的主要耗时部分,具体应用中,我们应根据需要选择合适的积分步长;而 SPRT 与 APMC 算法的耗时都小于 1ms.尽管 BHT 与 BIE 的统计过程耗时严重,但仍远小于仿真路径的生成和验证所需的时间.例如,BIE 在积分步长为 10 000 时,单次迭代耗时仅 0.02s,而仿真软件生成一条路径的耗时普遍在 0.3s~3s,可见,实际应用中,BIE 的积分过程并不会成为性能瓶颈.由以上实验数据分析可知,BIE 与 BHT 所需路径数量较少,因此实际应用中的总体性能将优于 SPRT 与 APMC.

Table 1 Time consuming of different SMC algorithms' computation part (at 1 000 iterations)

表 1 SMC 算法统计计算部分的耗时对比(1 000 次迭代)

SMC 算法	积分步长:1 000	积分步长:10 000	积分步长:100 000
BHT	208ms	1883ms	18.2s
BIE	27.1s	241.6s	40min
APMC	<<1ms		
SPRT	<<1ms		

综上,我们可以得出以下结论:

(1) 定性类算法(SPRT 和 BHT)得出验证结果所需的路径数要小于定量类算法(APMC 和 BIE),但定量类算法能够评估出系统满足目标约束的概率区间,可以有效地对系统的性能进行定量评估,这在实际中更有用.

(2) SPRT 和 BHT 算法在验证真实概率接近概率阈值的约束时性能最差,其中,BHT 所需路径数量将急剧增大,接近无穷,几乎无法完成验证;而 BIE 在检验真实概率 P 接近 0.5 的约束时性能最差,因此,使用中应尽量避免这两类情况出现.

(3) APMC 算法的统计分析效率高于 BIE 算法,而 BIE 算法所需路径数少于 APMC 算法.对于大多数模型,路径的生成和验证都为主要耗时部分,所以 BIE 算法在实际应用中性能更好.

2 基于抽象和学习的 SMC 算法

通过对现有的 4 种 SMC 算法的原理分析和实验数据的总结可知,在实际应用中,BIE 算法可定量确定目标

约束的概率区间且效率较高.但在实际概率接近 0.5 时,BIE 算法所需路径数量急剧增加,由于路径的生成和验证为 SMC 算法的主要耗时部分,这将严重影响算法的性能.为了在整个概率区间内都能有效地使用 SMC 算法对 CPS 进行约束验证,我们提出了一种基于抽象和学习的 SMC 算法(AL-SMC),以解决 BIE 算法在真实概率 P 接近 0.5 时所需路径数目剧增的问题.

2.1 基于抽象和学习的统计模型检测

为了避免直接计算最终概率,可采用分解模型的方式,使得每一个子模型计算的概率远离 0.5.基于此思想,我们提出了一种基于抽象和学习的 SMC 算法.

2.1.1 基本思想

通过对 SMC 算法分析路径过程的研究可知,每条路径在某一时间点上即可判断满足或不满足ϕ'.而对于不同模型,该点在时间轴上的分布情况不同.假设对于 n 个样本路径(时间长度为 t),最终在时间$[0,t]$上满足ϕ'的概率为 p,且存在 $m\geqslant 1$ 段子路径(时间段分别为 $[t_1^i,t_2^i]$ 且 $[t_1^i,t_2^i]\in(0,t)$,$1\leqslant i\leqslant m$),其满足ϕ'的概率分别为 $p_1,...,p_m$,则有 $p=\sum_{i=1}^{m}p_i$. 如果 p=0.5,则 $p_1,...,p_m$<0.5,且划分出的 $p_1,...,p_m$ 越平均,效果越好.若分别检验 m 段子路径的满足概率,理论上所需的总路径数将会减少,即 $Trs(p')=\max(Trs(p_i))<Trs(p)$,$1\leqslant i\leqslant m$($Trs(\cdot)$代表所需路径数).但 CPS 生成的原始路径都十分复杂,状态数量极大,AL-SMC 的核心在于逐步抽象、学习路径,尽量简化路径以划分出概率更平均的路径段.从本质上讲,路径分段等价于子模型划分,且所有子路径段满足ϕ'的概率值近似构成了模型满足ϕ'的概率密度.

AL-SMC 算法的核心思想如图 6 所示.

- 首先,取少量的样本路径(样本数量可根据不同的精度需求自定)作为路径训练集,并将其作为基于抽象和学习的路径预处理器的输入.
- 接着执行预处理器.它主要包括 4 个功能模块:基于约束的投影、基于主成分分析的特征降维、关键点抽取以及前缀频率树的构建与约减.
- 经过预处理器处理后得到的 PFT 和增量原始路径一起作为统计模型检测器的输入,借助多个普通的 SMC(如 BIE)算法同时进行统计分析,再求和得到最终概率.

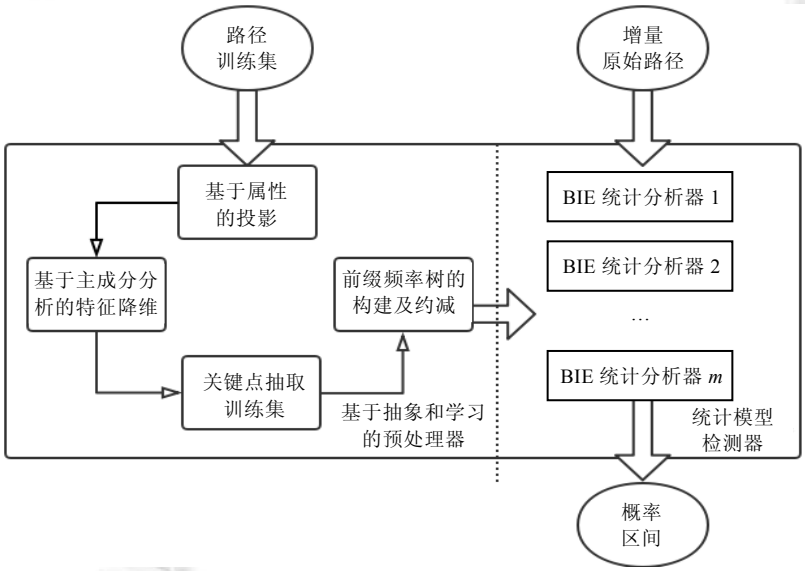


Fig.6 Framework of the optimized SMC algorithm

图 6 优化的统计模型检测算法框架

下面,我们将简单介绍 AL-SMC 算法中两个阶段的核心步骤,更详尽的算法过程参见文献[16].

2.1.2 抽象阶段

抽象阶段利用不同的抽象技术对原始路径δ^0进行简化,其详细代码描述如图 7 所示.

算法 1. Trace 的简化和抽象.
输入:BLTL 属性公式ϕ',原始 Trace δ^0,Trace 提取阈值 p.
输出:简化和抽象后的新 Traceδ^3,原始特征矩阵 EM.
1: δ^1:=*traceDiscretization*(δ^0) //原始 Trace 离散化
2: **for all** $s\in\delta^1$ **do**
3: s:=s+*sChk*+*tChk* //对离散 Trace 的每一个状态增加状态和 Trace 的验证标记
4: **end for**
5: δ^2, *EM*:=*pcaDimensionReduction*(δ^1) //使用 PC A 算法对进行降维,并得到特征矩阵
6: **for all** $s\in\delta^2$ **do**
7: **if** *tChk*=1 **then**
8: $\delta^3\cup s$ //对于 tcheck 值为 1 的状态,直接加入新的 Trace 中
9: **end if**
10: **end for**
11: : $\delta^3=\delta^3\cup$*traceExtraction*(δ^2,p) //将出现频率较大的 Trace 加入新的 Trace 中
12: **return** δ^3, *EM*

Fig.7 Abstraction and simplification of Trace

图 7 路径的简化和抽象

如算法 1 中所描述,路径的简化和抽象主要分为 3 个关键步骤,即算法中的 *traceDiscretization*(·)函数(对应于图 6 中的基于属性的投影模块)、*pcaDimensionReduction*(·)函数(对应于图 6 中的主成分分析模块)和 *traceExtraction*(·)函数(对应于图 6 中的前缀频率树的构建和降维模块).下面我们简要介绍这 3 个函数的基本思想.

1) *traceDiscretization*(·)函数

在 CPS 中,存在大量离散和连续变量,设变量集合为 $SV=L\cup X$;一条原始路径δ^0是 k 维向量 v 的集合(k=#(SV),即 SV 的元素个数).由于离散变量改变的状态点对应于系统的关键状态点,基于此,分两步对原始路径进行处理:

(1) 基于离散变量分段.找到路径中所有离散变量有变化的时间点 t_i,对路径进行第 1 次分段,并增加两个维度 *sChk*;*tChk*:*sChk* 表示子段路径是否满足ϕ',*tChk* 表示整段路径目前是否满足ϕ'.

(2) 面向目标约束对连续变量投影.当检验某一确定约束公式(如ϕ')时,只考虑某些连续变量在路径的哪一点使约束满足,因此将约束中不包含的连续变量删去,其余连续变量投影到上面新增的两维上去.投影方法类似于检查给定路径段上ϕ'是否满足,*sChk* 只检验子路径满足情况,*tChk* 从第 1 个 *sChk* 变为 1 的时刻开始,之后都为 1.

2) *pcaDimensionReduction*(·)函数

经过 *traceDiscretization*(·)函数处理后,路径中只留下离散变量.由于对某一具体的公式ϕ',影响系统满足该约束的核心变量并不多,可利用变量之间的相关性来进一步减少变量.我们借助机器学习中常用的一种降维算法——主成分分析(principal components analysis,简称 PCA)[17],继续对路径进行投影抽象.

首先将特征的概念引入 CPS 模型中,并利用方差贡献来建立目标评价模型,计算每个成分(即变量)的综合得分,然后排序.在经过标准化、计算协方差矩阵、计算特征值和特征向量、求主成分和计算综合得分等步骤后,即可求得问题的主成分变量,从而减少路径中的变量总数.

3) *traceExtraction*(·)函数

经过抽象之后,状态已经相对较少,但并非所有状态量都对计算最终概率有用,还需将对路径中的状态点进行筛选.关键点提取方法的具体步骤是:

(1) 关键点必包含每条路径中判断其满足ϕ'的点(即 *tChk*=1 的点),这是计算满足概率的直接相关点.

(2) 经 PCA 降维后的变量不再是二值,而是多值,因此,我们选择训练样本集下频率总和有较大覆盖的取值(如频率和≥0.8)即可.

2.1.3 学习阶段

经过前 3 个步骤的抽象简化后,路径的复杂度大为降低.接下来是学习阶段,通过构造用于统计模型检测分析的前缀频率树来划分出概率更平均的路径段,前缀频率树的定义如下.

定义(前缀频率树(prefix frequency tree,简称 PFT)). 一棵前缀频率树 T 是一个元组 $T=(D,R,d_0)$,其中,

- D 代表树结点的集合.每个结点可表示为 $d=(id,f,n)$,其中,id 是这个状态向量唯一编号(用 0/1 串表示),f 记录在此结点终止的路径数,n 记录所有经过此结点的路径数,所以任意结点都满足 $f\leqslant n$.
- R 代表结点之间的关系,可表示为(d,d');除根结点外,每个结点对于 R 来说都有且仅有 1 个前驱结点,但可以有 0 个或多个后继结点.
- $d_0\in D$ 为树的根结点,有且仅有 1 个,对于 d_0 来说没有前驱结点.

根据文献[18]中的随机文法推断方法(用于从句子回推文法自动机),可将系统的离散路径看作一种语言的句子来构造前缀频率树.我们规定不满足目标约束的样本路径为空句,即在 PFT 的根结点终止;满足目标约束的样本路径在某一非根结点终止.如此即可构建出一棵 PFT,它实际上是原系统的一个高度离散化的抽象模型,并具有如下性质:

- $n(d_{leaf})=f(d_{leaf})$,即所有叶子结点的 f 与 n 相等;
- $n(d_i)-f(d_i)=\sum_{d_t\in child(d_i)}n(d_t)$,即任一非叶子结点的 n 与 f 之差等于其所有孩子结点上的 n 之和;
- $n(d_0)-f(d_0)=\sum_{i>0}f(d_i)$,即根结点的 n 与 f 之差等于所有非根结点的 f 之和.

尽管抽象阶段已对路径做了简化处理,但直接利用路径δ^3构造出的 PFT 仍然十分庞大,因此,可通过前缀频率树的横向、纵向约减来有效减小 PFT 的规模,便于之后的分析.如图 8 所示,在构造初始 PFT 之后,调用 *reduce*1_*recur*(·)函数,通过检验、判断每个结点的孩子结点的 n 和 f 值来进行横向合并;然后,根据每个结点是否有分支和 f 值的情况,进行结点的纵向合并,即 *reduce*2_*recur*(·)函数,完成进一步约减.

算法 2. 前缀频率树的构建及约减.
输入:关键点提取后的 Trace 集合Σ^3,PFT 约减参数 r.
输出:构建和约减后的前缀频率树 T.

```
1:  d0:=rootofT                          //为的根结点
2:  dtemp:=d0                            //构建过程从根结点开始
3:  NodeCollection:=∅                    //node 集合初始化为空
4:  for all δ_i^3 ∈ Σ^3 do
5:  d_temp := getNode(δ_i^3)
6:  n(d_temp),f(d_temp):=addNodeNF(d_temp)   //对每一个结点增加 n,f 标记
7:  NodeCollection∪d_temp                //将得到的 node 加入 node 集合中
8:  end for
9:  d0:=PFTConstruction(NodeCollection)  //通过 node 集合构建 PFT
10: d0:=reduce1_recur(d0,r)              //PFT 第 1 阶段约减
11: d0:=reduce2_recur(d0,r)              //PFT 第 2 阶段约减
12: return T with d0
```

Fig.8 Construction and reductions of PFT

图 8 前缀频率树的构建及约减算法

经过两步约减后,PFT 各终止结点上的路径样本数更加平均,即划分出了概率更平均的路径段.若将终止结点数/总结点数定义为前缀频率树的结点有效率,则使用约减算法能够提高结点的有效率.

2.1.4 最终概率的统计分析

约减后的 PFT 即 T',可看作对原系统在概率空间上进行了划分的抽象模型.假设 T'含有 m 个非根终止结点(f>0),考虑到在大样本量的情况下会有未被 T'接受的正样本,我们设置 m+1 个 BIE 分析器,并发执行 BIE 统计模型检测算法,对在不同结点终止的路径分别进行统计.当所有 BIE 分析器都终止时,整个算法终止.最后,对每个 BIE 分析器重新执行一次计算,并分别进行如下处理:

1) 若 $p_i>\delta$,即 pi 超过半区间大小,则表明 p_i 值较准确,不修正;

2) 若 $p_i \leqslant \delta$,则表明 p_i 不准确,执行修正 $p_i=x_i/N$.

最终概率为 $p=\sum_{i=1}^{m+1} p_i$.此过程可以描述为算法 3,如图 9 所示.

算法 3. 最终概率的多 BIE 分析.
输入:约减后的前缀频率树 T',BLTL 属性ϕ',半区间大小δ,终止结点数 m.
输出:最终概率.

```
1:  I:={(x_1,γ_1,end_1),...,(x_{m+1},γ_{m+1},end_{m+1})}   //BIE 分析器集合,分别代表第 i 个分析器的
                                                            正样本 Trace 数、统计参数及是否终止
2:  N:=0                                                    //记录 Trace 样本总数
3:  while ∃i∈I and end_i=false do
4:  δ:=generateSampleTrace()                                //仿真器生成一条 Trace
5:  δ':=preprocess(δ)                                       //经过预处理器,得到抽象 Trace
6:  if δ'⊨φ' then
7:      i:=findEndNode(T',δ')                               //沿树的某条路径找到对应的终止结点 i
8:      p_i,γ_i:=computeStatisticalParameter_i(I)           //执行第 i 个终止结点对应的 BIE 算法
9:  end if
10: if checkEndCondition_i(γ_i) then
11:     end_i=true                                          //此时,第 i 个分析器终止
12: end if
13: end while
14: return p:=∑_{i=1}^{m+1} p_i                             //计算并返回最终概率
```

Fig.9 Multi-BIEs calculation of the final probability

图 9 最终概率的多 BIE 统计算法

2.2 算法分析

AL-SMC 算法从原始路径(σ^0)抽象简化至最终的前缀频率树(T')构建、约减的过程,对于目标约束ϕ'而言,ϕ'是否满足以及何时满足的信息并无丢失,只是变换了观察视角.即:对于ϕ'的概率空间来说,系统模型间存在等价关系:

$$P_M(\delta^0)=P_M(\delta^1)=P_M(T') \tag{9}$$

因此,最终概率评估的误差取决于 BIE 统计分析的过程.

AL-SMC 算法中各阶段的时间复杂度与空间复杂度见表 2.由表 2 可知:1) 基于 PCA 的特征降维的复杂度取决于 PCA 算法本身所需的特征数;2) 两种递归算法在 PFT 复杂时可能会出现递归栈溢出,可通过调整参数来有效地控制 PFT 的规模来解决这一问题;3) 最终概率的多 BIE 分析算法所需迭代次数因实际情况而定,而单次迭代的时间主要取决于积分步长.

Table 2 Time and space complexity of AL-SMC algorithm

表 2 AL-SMC 算法的时间与空间复杂度

阶段算法		时间复杂度	空间复杂度	说明
抽象阶段	基于约束的投影	$O(mn)$	$O(l)$	m 表示路径长度,n 表示训练样本数
	基于 PCA 的特征降维	$O(\min(k^3,n^3))$	$O(k^2)$	k 表示特征维数,n 表示训练样本数
	关键点抽取	$O(l)$	$O(l)$	降维后特征数极少,算法接近常数时间
学习阶段	前缀频率树的构建	$O(mn)$	$O(l)$	m 表示路径长度,n 表示训练样本数
	前缀频率树的约减 I	$O(d\log d)$	$O(\log d)$	递归算法,d 表示树结点数
	前缀频率树的约减 II	$O(d\log d)$	$O(\log d)$	递归算法,d 表示树结点数
最终概率的多 BIE 分析(单次迭代过程)		$O(\log d+i)$	$O(l)$	d 表示树结点数,i 表示积分的复杂度

3 自适应的统计模型检测框架

AL-SMC 算法可以有效缓解 BIE 算法在实际概率值接近 0.5 时遇到的性能问题,因此,结合 BIE 和 AL-SMC 算法,可以对[0,1]区间上的任意概率进行有效验证.在此基础上,本文提出了一种动态的自适应 SMC 算法框架,

对于一个 CPS 实例,可根据概率值的预估,灵活选择 SMC 算法,其算法流程图如图 10 所示.

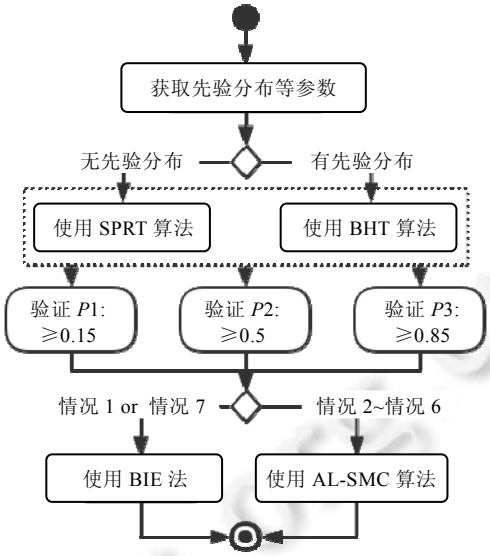


Fig.10 Framework of the self-adaptive SMC algorithm

图 10 自适应 SMC 算法框架

首先,根据有无先验分布,选择定性算法 SPRT 或 BHT,进行概率区间预估计(通常可以在 100 条~200 条路径内完成).为了快速预估计概率,我们对[0,1]概率区间进行划分,有 t_1=0.15,t_2=0.5,t_3=0.85 这 3 个点.同时检验 3 个约束公式 $P1{:}P_{\geqslant 0.15}(\varphi')$,$P2{:}P_{\geqslant 0.5}(\varphi')$,$P3{:}P_{\geqslant 0.85}(\varphi')$,通过它们的满足情况选择合适的 SMC 算法,再进行定量评估.SPRT 或 BHT 分析的误差概率取 0.1,以保证 $P1$,$P2$,$P3$ 中至少能有 2 个约束可在 250 条路径内完成,若有某一约束在 250 条路径内尚未完成,则说明真实概率在其对应的概率区间内.

图 11 为最终定量算法的选择区间.这里有两种定量类算法可选:BIE 算法与 AL-SMC 算法.当所检验概率在$[0+\delta,0.5-\eta]$和$[0.5+\eta,1-\delta]$范围时(η一般取 0.3~0.35,δ为半区间大小),选择 BIE 算法;否则,选择 AL-SMC 算法.AL-SMC 算法弥补了 BIE 算法的不足,可以更高效地检验真实概率 P 接近 0.5 左右的概率约束,但需要保证 P 不超过 0.5.

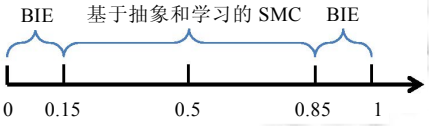


Fig.11 Selection interval of the final algorithm

图 11 最终定量算法的选择区间

下面给出详细的算法选择过程.$P1$,$P2$,$P3$ 的检验结果可分为 7 种情况.

(1) $P1$,$P2$,$P3$ 都不满足,表明 $P\in[0,0.15)$,选择 BIE;

(2) $P1$ 未完成且 $P2$,$P3$ 不满足,表明 P 接近 0.15,选择 AL-SMC;

(3) $P1$ 满足且 $P2$,$P3$ 不满足,表明 $P\in(0.15,0.5)$,选择 AL-SMC;

(4) $P2$ 未完成且 $P1$ 满足 $P3$ 不满足,表明 P 接近 0.5,选择 AL-SMC;

(5) $P1$,$P2$ 满足且 $P3$ 不满足,表明 $P\in(0.5,0.85)$,约束取反,选择 AL-SMC;

(6) $P1$,$P2$ 满足且 $P3$ 未完成,表明 P 接近 0.85,约束取反,选择 AL-SMC;

(7) $P1,P2,P3$ 都满足,表明 $P\in(0.85,1]$,选择 BIE.

最后,根据自适应 SMC 算法框架所选择的算法,完成最终概率的统计分析.本算法中虽然多次调用 SMC 算法,但可以重用已生成的路径,因此整体效率较高.其中,概率的划分和选择也基于大量实验结果,具有一定的通用性.自适应 SMC 方法的详细伪代码如图 12 所示.

算法 4. 自适应的统计模型检测算法框架.
输入:BLTL 属性公式ϕ',先验概率密度 g.
输出:最终概率.

```
Σ:=∅                                          //保存当前已生成的 Trace 集合
(P1,P2,P3):=(P≥0.15(φ'),P≥0.5(φ'),P≥0.85(φ'))  //预估所需的 3 个属性
pre:=null                                     //根据是否有先验分布,选择 SPRT 或 BHT 算法
if g≠null then
pre:=BHT(0.1)                                 //选择 BHT 算法,误差概率取 0.1
else
pre:=SPRT(0.1)                                //选择 SPRT 算法,误差概率取 0.1
end if
(r1,r2,r3):=(undone,undone,undone)            //预估结果初值
while size(Σ)<250 and r1=undone∨r2=undone∨r3=undone do
(r1,r2,r3):=pre(P1,Σ),pre(P2,Σ),pre(P3,Σ)     //依次调用 3 个定性算法,返回当前结果,
                                              //并保存当前新 Trace 到中以待重用
end while
if r1∧r2∧ r3 or ¬r1∧¬r2∧¬r3 then
p:=BIE(φ',Σ)                                  //调用 BIE 算法
else if r1∧r2 and ¬r3∨¬r3=undone then
p:=ALSMC(¬φ',Σ)                               //属性取反,调用 AL-SMC
else
    p:=ALSMC(φ',Σ)                            //调用 AL-SMC
end if
return p
```

Fig.12 Pseudocode of the self-adaptive SMC algorithm
图 12 自适应 SMC 算法框架伪代码

4 案例分析

为了对自适应统计模型检测方法进行实际应用分析,我们对智能温控系统、移动机器人路径规划和蓝牙协议进行了约束验证,并根据实验结果分析比较了自适应 SMC 算法框架和单纯的 BIE 算法的性能.待验证的约束公式见表 3(每个公式都在两种参数设定下进行验证:(1) δ=0.05,c=0.99;(2) δ=0.02,c=0.9).

Table 3 Constraint formula of three benchmarks
表 3 3 个案例的目标约束公式

案例	公式名	约束
移动机器人路径规划	ϕ_1	$P_{=?}(F^{\leqslant 100}c\geqslant 150)$
	ϕ_2	$P_{=?}(F^{\leqslant 100}c\geqslant 500)$
智能温控系统	ϕ_3	$P_{=?}(F^{\leqslant 48}energy\geqslant 210)$
	ϕ_4	$P_{=?}(F^{\leqslant 48}discomfort\geqslant 15)$
蓝牙协议	ϕ_5	$P_{=?}(F^{\leqslant 50000}energy\geqslant 290)$
	ϕ_6	$P_{=?}(F^{\leqslant 50000}energy\geqslant 2000)$

其中,ϕ_1 和 ϕ_2 表示在移动机器人路径规划问题中的能量和时间的约束检验,ϕ_3 和 ϕ_4 表示对智能温控系统中能量消耗和不舒服度的检验,ϕ_5 和 ϕ_6 表示在蓝牙协议中对能量消耗的检验.

为了形象地阐述自适应统计模型检测框架在实例中的应用,我们借助智能温控系统对公式 ϕ_4(0.05,0.99)的验证来简单解释这一过程:首先,根据先验分布,选择 BHT 算法在 200 条路径内来验证 $P1,P2,P3$,由于 $P2$ 未完成(即前述情况 4),可知真实概率 P 在 0.5 左右,故选择 AL-SMC 进行验证.经过对路径训练集的抽象和学习过程,再利用增量原始路径,使用多 BIE 统计分析器可计算出最终的概率结果.

我们对每个案例进行了 100 次重复实验,并将实验结果取最小值、最大值、平均值这 3 种情况进行分析.表 4 为不同情况下,自适应 SMC 方法和 BIE 算法所需的路径数目.由表 4 可知,采用自适应 SMC 算法框架可有效减少实验所需的路径数,即提高了性能,但是这种性能的提高并不稳定(如对公式ϕ_4,可减少 40%的路径数;而对于公式ϕ_1,只能减少 16%的路径数,从理论上来说,这取决于 PFT 的叶子结点的分布).

Table 4 Comparison of required Traces between BIE and self-adaptive SMC algorithm of three benchmarks

表 4 3 个案例中 BIE 和自适应 SMC 方法所需路径数的比较

案例	公式名和参数 (δ,c)	最少路径数 (BIE/自适应 SMC)	最大路径数 (BIE/自适应 SMC)	平均路径数 (BIE/自适应 SMC)
移动机器人路径规划	ϕ_1(0.05,0.99)	653/501	659/591	657/459
	ϕ_2(0.02,0.9)	1673/1335	1690/1496	1686/1496
	ϕ_2(0.05,0.99)	398/250	528/398	480/320
	ϕ_2(0.02,0.9)	1083/635	1305/949	1222/798
智能温控系统	ϕ_3(0.05,0.99)	578/250	647/253	625/250
	ϕ_3(0.02,0.9)	1546/342	1603/469	1546/342
	ϕ_4(0.05,0.99)	645/250	655/250	645/250
	ϕ_4(0.02,0.9)	1677/457	1685/459	1677/457
蓝牙协议	ϕ_5(0.05,0.99)	431/250	541/250	488/250
	ϕ_5(0.02,0.9)	1148/250	1344/455	1257/335
	ϕ_6(0.05,0.99)	626/250	656/289	645/250
	ϕ_6(0.02,0.9)	1598/415	1676/658	1648/514

表 5 为自适应 SMC 算法和 BIE 算法的实验结果对比,经过简单分析,我们得出,自适应 SMC 算法的概率误差被限制在一个区间内,其准确度是可接受的.对于误差分析,在文献[16]中我们做了更严谨的讨论.在实际应用过程中,可以针对具体情况,通过动态调整 r(r 为约减参数,其值表示前缀频率树的约减程度)值来控制自适应 SMC 算法的准确度.因此从实用角度来说,相对于单纯的 BIE 算法,自适应 SMC 算法的效率更高.

Table 5 Comparison of result between BIE and self-adaptive SMC algorithm of three benchmarks

表 5 3 个案例中 BIE 和自适应 SMC 的结果比较

案例	公式名和参数 (δ,c)	最小概率 (BIE/自适应 SMC)	最大概率数 (BIE/自适应 SMC)	平均概率 (BIE/自适应 SMC)
移动机器人路径规划	ϕ_1(0.05,0.99)	0.45802/0.43788	0.54421/0.55015	0.49819/0.49955
	ϕ_2(0.02,0.9)	0.46730/0.47491	0.53147/0.53311	0.49786/0.50209
	ϕ_2(0.05,0.99)	0.185/0.18391	0.27925/0.29511	0.24121/0.24212
	ϕ_2(0.02,0.9)	0.20184/0.20003	0.26243/0.28108	0.23877/0.23903
智能温控系统	ϕ_3(0.05,0.99)	0.32759/0.33402	0.43914/0.48486	0.39084/0.41046
	ϕ_3(0.02,0.9)	0.3553/0.34755	0.41981/0.46737	0.3909/0.40536
	ϕ_4(0.05,0.99)	0.42813/0.42612	0.54116/0.56874	0.48225/0.50363
	ϕ_4(0.02,0.9)	0.46099/0.44511	0.51391/0.54631	0.48473/0.49826
蓝牙协议	ϕ_5(0.05,0.99)	0.20554/0.14035	0.28913/0.34529	0.2471/0.25321
	ϕ_5(0.02,0.9)	0.22421/0.19866	0.28026/0.32487	0.25009/0.26466
	ϕ_6(0.05,0.99)	0.39013/0.373	0.47416/0.52743	0.43491/0.45034
	ϕ_6(0.02,0.9)	0.40148/0.38922	0.45709/0.49834	0.42684/0.44709

5 相关工作

人们利用统计知识分析系统由来已久,SMC 技术建立在蒙特卡洛模拟、假设检验等统计方法之上,通过统计分析系统仿真运行的路径来验证系统满足约束的情况.它最早被 Sen 等人提出,用来验证黑盒系统[19],即 SSP 算法的雏形,其难点在于确定算法收敛所需的路径总样本数量 N 以及接受原假设的阈值 C.Younes 等人[20]提出一种基于二叉搜索的算法来近似得到所需的 N 和 C,并指出了文献[21]中验证黑盒系统方法的一些错误.Younes 等人还在 Wald 的 SPRT 原理的基础上提出了基于对数的 SPRT 实现算法[11,20],以最小化算法所需路径的样本数量.SSP 与 SPRT 都用来解决定性验证问题,即,回答“系统 S 满足约束ϕ'的概率是否大于或等于某个概率阈值”这个问题,即 $S\models P_{\geqslant\theta}(\phi')$.与定性算法不同,定量算法可以直接返回 S 满足ϕ'的概率 p,如 APMC[14],通过计算 x/n 来

计算 p(其中,x 和 n 分别表示所需的路径正样本数量和总数量),并通过 Chernoff-Hoeffding 界来限定结果的误差范围.随后,Zuliani,Clarke 等人基于贝叶斯统计又提出了两种新 SMC 算法:BHT 和 BIE[12,15],前者基于贝叶斯假设检验解决定性验证问题,后者基于贝叶斯区间估计解决定量评估问题.

从 SSP,SPRT,APMC 到 BHT,BIE,SMC 算法所需路径数量逐渐减少,效率逐渐提高.为了进一步探索 SMC 技术,人们将数值方法与统计方法相结合[22,23],以进一步提升 SMC 效率或解决一些 SMC 难以解决的问题,如非确定性问题(non-determinism).研究 SMC 非确定性算法的还有 Henriques,其博士学位论文讨论了如何用概率方式近似解决非确定性问题[24].目前,SMC 主要验证有界(通常指时间约束,即 time-bounded)的约束,因此,如何验证传统模型检测中的无界“Until”约束,也成了 SMC 研究热点之一,He,Jennings 等人提出了一种将无界“Until”验证转化为有界“Until”的方法以解决此问题[25,26].此外,由于 SMC 针对系统仿真结果进行分析,所以也不可避免地引入了仿真领域的问题,比如小概率事件在路径中出现概率极低,使得验证过程需要产生路径的数量过多而效率低.针对该问题,Jegourel,Legay 等人基于重要性取样(importance sampling)和重要性分割(importance splitting)技术提出了面向小概率约束验证的 SMC 算法[27,28],极大地减少了验证所需的路径样本数量;解决类似问题还可以借助于机器学习技术,我们之前的研究工作[29]中提到的借助支持向量机预测事件、Kumar 在文献[30]中借助贝叶斯推断预测事件,都可以提高 SMC 的效率.

Ymer[31]和 Vesta[32]是最早实现 SMC 的验证工具.Vesta 采用了极易实现并行化的 SSP 算法的一个变种,Ymer 采用的 SPRT 算法很快也被 Younes 证明同样能够被并行化.Vesta 支持了无界“Until”的验证,但 Ymer 在实验中的效率高于 Vesta.此外,UPPAAL-SMC[33]和 Prism[34]的最新版本也支持 SMC 技术,二者都实现了定性的 SPRT 算法,同时也实现了基于置信区间(confidence interval,简称 CI)[35]的定量算法(文献[36]对不同版本的 CI 统计算法进行了对比,并根据需求的不同给出了方法选择的建议).其中,UPPAAL-SMC 支持使用 SHA 模型建模 CPS 的随机行为,并支持使用 SMC 算法定量评估系统的性能;Prism 则使用 Reactive Modules Language(RML)建模,对随机(如马尔科夫链)和非确定性(如马尔科夫决策过程)模型的验证支持较好.Plasma Lab[37]是新出现的一款纯 SMC 工具,同样支持 RML 建模,并实现了多种 SMC 算法(包括重要性取样和分割等面向小概率事件的算法).Plasma Lab 允许用户以插件集成的方式为其添加新的模型输入和验证算法,如 Simulink 的集成.

6 总 结

统计模型检测技术能够有效地分析、验证 CPS 的正确性、可靠性,并能对系统的性能进行定量评估.然而,随着系统规模的日益增大,如何有效地提高使用统计模型检测技术分析、验证 CPS 的效率,是使用模型检测技术验证、分析 CPS 所面临的主要问题之一.针对该问题,我们提出了一套有效的解决方案.主要工作贡献包括:

1) 首先,对已有 SMC 算法的原理进行了剖析,通过实验分析,详细比较已有的 SMC 算法的性能,总结出现有 SMC 算法的局限性和性能瓶颈.针对贝叶斯区间估计算法的不足,提出了基于抽象和学习的统计模型检测方法,以有效解决真实概率 P 接近 0.5 时的目标约束验证问题.
2) 其次,提出了一种自适应的 SMC 算法框架,能够根据系统满足目标约束的概率估值,动态地、灵活地选择合适的 SMC 算法,旨在从整体上提高 SMC 技术的性能,提供一整套面向 CPS 的 SMC 检测方案.
3) 具体案例分析,通过 3 个实例,将自适应的 SMC 算法与单纯的 BIE 算法的效率做对比,同时给出了谨慎的误差分析.

不足之处在于:本文的方法是将概率空间划分后,分别统计再进行累加,与直接统计分析整个概率空间相比有一定的误差.大量实验结果表明,该方法误差极小,属于可接受范围.我们下一步的工作是继续优化、改进自适应 SMC 算法,并对算法的误差分析进行精确的量化评估.

References:

[1] Lee EA. Cyber physical systems: Design challenges. In: Proc. of the 2008 11th IEEE Int'l Symp. on Object Oriented Real-Time Distributed Computing (ISORC). 2008. 363−369. [doi: 10.1109/ISORC.2008.25]

[2] Baier C, Katoen JP, Baier C, Katoen JP. Principles of Model Checking. Cambridge: MIT Press, 2008.

[3] Platzer A. Differential dynamic logic for hybrid systems. Journal of Automated Reasoning, 2008,41(2):143−189. [doi: 10.1007/s10817-008-9103-8]

[4] Platzer A, Clarke EM. Computing differential invariants of hybrid systems as fixedpoints. In: Proc. of the Computer Aided Verification. 2008. 176−189. [doi: 10.1007/978-3-540-70545-1_17]

[5] Legay A, Delahaye B, Bensalem S. Statistical model checking: An overview. In: Proc. of the Runtime Verification. 2010. 122−135. [doi: 10.1007/978-3-642-16612-9_11]

[6] Chen M, Yue D, Qin X, Fu X, Mishra P. Variation-Aware evaluation of MPSoC task allocation and scheduling strategies using statistical model checking. In: Proc. of the Design, Automation & Test in Europe Conf. & Exhibition (DATE). 2015. 199−204.

[7] Jha SK, Clarke EM, Langmead CJ, Legay A, Platzer A, Zuliani P. A bayesian approach to model checking biological systems. In: Proc. of the Computational Methods in Systems Biology. 2009. 218−234. [doi: 10.1007/978-3-642-03845-7_15]

[8] Chen MS, Gu P, Xu SY, Chen XH. Formal evaluation of scheduling strategies for smart building air-conditioning systems under uncertain environment. Ruan Jian Xue Bao/Journal of Software, 2016,27(3):655−669 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4987.htm [doi: 10.13328/j.cnki.jos.004987]

[9] Cheng B, Wang X, Liu JF, Du DH. Modana: An integrated framework for modeling and analysis of energy-Aware CPSs. In: Proc. of 2015 IEEE the 39th Annual Computer Software and Applications Conf. (COMPSAC). IEEE, 2015. 127−136. [doi: 10.1109/COMPSAC.2015.68]

[10] Wald A. Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics, 1945,16(2):117−186. [doi: 10.1214/aoms/1177731118]

[11] Younes HL, Simmons RG. Statistical probabilistic model checking with a focus on time-bounded properties. Information and Computation, 2006,204(9):1368−1409. [doi: 10.1016/j.ic.2006.05.002]

[12] Hoeffding W. Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association, 1963, 58(301):13−30. [doi: 10.1080/01621459.1963.10500830]

[13] Jeffreys H. The Theory of Probability. OUP Oxford, 1998.

[14] Hérault T, Lassaigne R, Magniette F, Peyronnet S. Approximate probabilistic model checking. In: Proc. of the Verification, Model Checking, and Abstract Interpretation. 2004. 73−84. [doi: 10.1007/978-3-540-24622-0_8]

[15] Zuliani P, Platzer A, Clarke EM. Bayesian statistical model checking with application to Stateflow/Simulink verification. Formal Methods in System Design, 2013,43(2):338−367. [doi: 10.1007/s10703-013-0195-3]

[16] Jiang KQ, Huang P, Zan H, Du DH. AL-SMC: Optimizing statistical model checking by automatic abstraction and learning. Int'l Journal of Software and Informatics, 2016,10(4):54−69. [doi: 10.21655/ijsi.1673-7288.00235]

[17] Dunteman GH. Principal Components Analysis. Sage, 1989.

[18] Carrasco RC, Oncina J. Learning stochastic regular grammars by means of a state merging method. In: Proc. of the Grammatical Inference and Applications. Springer-Verlag, 1994. 139−152. [doi: 10.1007/3-540-58473-0_144]

[19] Sen K, Viswanathan M, Agha G. Statistical model checking of black-box probabilistic systems. In: Proc. of the Computer Aided Verification. 2004. 202−215.

[20] Younes HL. Verification and Planning for Stochastic Processes with Asynchronous Events. DTIC Document, 2005.

[21] Sen K, Viswanathan M, Agha G. Statistical model checking of black-box probabilistic systems. In: Proc. of the Computer Aided Verification. 2004. 202−215. [doi: 10.1007/978-3-540-27813-9_16]

[22] Bogdoll J, Fioriti LMF, Hartmanns A, Hermanns H. Partial order methods for statistical model checking and simulation. In: Proc. of the Formal Techniques for Distributed Systems. Springer-Verlag, 2011. 59−74. [doi: 10.1007/978-3-642-21461-5_4]

[23] Pavese E, Braberman V, Uchitel S. Automated reliability estimation over partial systematic explorations. In: Proc. of 2013 the th Int'l Conf. on Software Engineering (ICSE). 2013. 602−611. [doi: 10.1109/ICSE.2013.6606606]

[24] Henriques D. Statistical model checking for markov decision processes [Ph.D. Thesis]. General Motors, 2012.

[25] He R, Jennings P, Basu S, Ghosh AP, Wu HQ. A bounded statistical approach for model checking of unbounded until properties. In: Proc. of the IEEE/ACM Int'l Conf. on Automated Software Engineering. 2010. 225−234. [doi: 10.1145/1858996.1859043]

[26] Jennings P, Ghosh AP, Basu S. A two-phase approximation for model checking probabilistic unbounded until properties of probabilistic systems. ACM Trans. on Software Engineering and Methodology (TOSEM), 2012,21(3):18. [doi: 10.1145/2211616.2211621]

[27] Jegourel C, Legay A, Sedwards S. Cross-Entropy optimisation of importance sampling parameters for statistical model checking. In: Proc. of the Computer Aided Verification. 2012. 327−342. [doi: 10.1007/978-3-642-31424-7_26]

[28] Jegourel C, Legay A, Sedwards S. Importance splitting for statistical model checking rare properties. In: Proc. of the Computer Aided Verification. 2013. 576−591. [doi: 10.1007/978-3-642-39799-8_38]

[29] Du DH, Cheng B, Liu J. Statistical model checking for rare-event in safety-critical system. Ruan Jian Xue Bao/Journal of Software, 2015,26(2):305−320 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4783.htm [doi: 10.13328/j.cnki.jos.004783]

[30] Kumar JA, Ahmadyan SN, Vasudevan S. Efficient statistical model checking of hardware circuits with multiple failure regions. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2014,33(6):945−958. [doi: 10.1109/TCAD.2014.2299957]

[31] Younes HL. Ymer: A statistical model checker. In: Proc. of the Computer Aided Verification. 2005. 429−433. [doi: 10.1007/11513988_43]

[32] Sen K, Viswanathan M, Agha GA. VESTA: A statistical model-checker and analyzer for probabilistic systems. In: Proc. of the QEST, Vol.5. 2005. 251−252. [doi: 10.1109/QEST.2005.42]

[33] Bulychev PE, David A, Larsen KG, Mikucionis M, Poulsen DB, Legay A, Wang Z. Uppaal-SMC: Statistical model checking for priced timed automata. arXiv preprint arXiv:1207.1272, 2012. [doi: 10.4204/EPTCS.85.1]

[34] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. of the Computer aided Verification. 2011. 585−591. [doi: 10.1007/978-3-642-22110-1_47]

[35] Brown LD, Cai TT, Dasgupta A. Interval estimation for a binomial proportion. In: Proc. of the Statistical Science. 2001. 101−117.

[36] Pires AM, Amado C. Interval estimators for a binomial proportion: Comparison of twenty methods. REVSTAT—Statistical Journal, 2008,6(2):165−197.

[37] Boyer B, Corre K, Legay A, Sedwards S. PLASMA-Lab: A flexible, distributable statistical model checking library. In: Proc. of the Quantitative Evaluation of Systems. Springer-Verlag, 2013. 160−164. [doi: 10.1007/978-3-642-40196-1_12]

附中文参考文献:

[8] 陈铭松,顾璠,徐思远,陈小红.不确定环境下智能大厦空调系统调度策略评估.软件学报,2016,27(3):655−669. http://www.jos.org.cn/1000-9825/4987.htm [doi: 10.13328/j.cnki.jos.004987]

[29] 杜德慧,程贝,刘静.面向安全攸关系统中小概率事件的统计模型检测.软件学报,2015,26(2):305−320. http://www.jos.org.cn/1000-9825/4783.htm [doi: 10.13328/j.cnki.jos.004783]

杜德慧(1979－),女,河南信阳人,博士,副教授,CCF 专业会员,主要研究领域为可信软件,模型验证,形式化方法.

姜凯强(1993－),男,硕士生,主要研究领域为可信软件,模型验证,形式化方法.

訾慧(1993－),女,硕士生,主要研究领域为可信软件,模型验证,形式化方法.

程贝(1988－),男,助理工程师,主要研究领域为可信软件,模型验证,形式化方法.