

摘要

本文的工作重点是研究半 Markov 控制过程中的并行优化算法。首先给出一种半 Markov 控制过程性能势的估计算法,相对于基于实现矩阵的估计方法具有速度快,对计算机硬件要求不高等特点。其后研究了半 Markov 控制过程基于单样本轨道的仿真优化算法,并从实际应用的角度出发发展了相应的并行算法,并提供了计算实例并分析比较其结果。

最后,我们给出了通讯中两个实际网络的优化与导数估计的应用实例.一是半 Markov 决策过程方法在航空军械综合系统维护问题中的应用;二是 M/PH/1 排队系统的性能灵敏度估计与仿真及其在 ATM 交换机性能分析中的应用。

本文的创新之处在于:

(1): 根据 markov 性能势理论,将它推广到半 markov 模型,得出了半 markov 模型的性能优化公式,并给出了相应的优化算法。

(2): 利用并行估计,给出了一套并行算法,大大提高了处理速度。

关键字: 半 markov 过程,性能势,实现矩阵,优化算法,并行仿真

Abstract

Parallel simulation algorithm of Semi-Markov control processes is studied in this thesis. A simulation algorithm of the performance potentials is presented, with the advantage of faster speed, lower demands of computer hardware than the algorithm based on realization matrix. Then we provide a simulation optimization algorithm of Semi-Markov control processes based on single sample path, and a corresponding parallel algorithm was developed. Some numerical examples are given to illustrate the application of the algorithms.

Finally, two real examples in communication were given. One example is that semi-Markov decision process method in the maintenance of the aviation ordnance synthetic system, the other is sensitivity estimate & simulations of performance in M/PH/1 queueing systems and the applications in ATM switch.

The innovation of this paper is:

(1): According to the Markov performance potentials theory, extend it to the semi-Markov model, get the performance optimization equation, and the optimization algorithms is given.

(2): based on the Parallel simulation, a Parallel algorithms is given, and the speed is much higher.

Keywords: semi-Markov control processes, performance potential, realization matrix, optimization algorithm, Parallel simulation

第一章 绪论

在这一章中简单介绍了离散事件动态系统，其后说明了本文的工作所涉及到的相关概念:半 Markov 控制过程和性能势、仿真及并行计算，并简单介绍了本文的结构，作为后续文章的基础。

1.1 离散事件动态系统

离散事件动态系统是系统和控制领域的一门新的分支，它在许多实际应用中有着广泛的应用前景，因此自提出以来得到飞速的发展。

1.1.1 基本概念

随着信息通讯、计算机和机器人等高新技术的发展和应用，在通信网络、柔性制造、交通管理、军事指挥等领域相继出现了大量反映现代科学技术发展方向的人造系统，这类系统通常能被描述为离散事件动态系统(DEDS)。经过近几十年的快速发展，离散事件动态系统已经成为控制和系统领域的一个前沿方向，并在当今一大批高科技领域获得了广泛应用，是系统和控制领域的一门新的分支，它是由哈佛大学何毓琦(Y C.Ho)教授等人于八十年代前后创立的[1]，名称也是由何毓琦教授等人提出的，以区别于此前已得到广泛研究的连续变量动态系统(CVDS)。在 DEDS 中，对系统行为进程起决定作用的是一批离散事件，而不是连续变量，所遵循的是一些复杂的人为规则，而不是物理学定律或广义物理学定律。正是基于对这类人造系统行为和性能研究的需要，推动着 DEDS 理论的形成和发展[2-12]

构成 DEDS 的基本要素是离散事件，DEDS 行为随时间的演化过程就是离散事件复杂交互影响的结果。粗略地说，离散事件是指 DEDS 中发生在离散时刻的事件，所谓事件则是使 DEDS 状态发生变动的一个行动或事情。

和离散时间连续变量动态系统不同，DEDS 中离散事件的发生时刻通常是异步的。离散事件的发生时刻，取决于这一时刻前系统行为的演化过程。柔性生产线中的“工件到达机床”和“工件加工完成”，通信网络中的“信号到达网络”和“信息传递结束”，就是离散事件的一些典型例子。在 DEDS 中，一个离散事件的发生，在驱动系统状态产生跃变的同时，还会按照系统的运行规则在系统其他部分触发新的离散事件，从而形成在离散事件驱动下系统状态的演化过程。柔性生产线中工件沿着加工机床的持续加工过程，通信网络中信息沿传输设施的持续传递和变换过程，就是离散事件复杂交互影响下所形成的 DEDS 行为演化过程的一些例子。

从上面可以看出，DEDS 中的离散事件具有三个基本特征:其一，离散事件是导致 DEDS 状态发生跃变和触发新离散事件的唯一因素，也即离散事件是驱动系统状态演化的基本因素。其二，离散事件的发生时刻是异步的和非约定的，即发生时刻由且只由系统的演化过程所决定。其三，离散事件是研究 DEDS 的主体，对 DEDS 的分析归结为确定离散事件交互影响所导致的系统状态的演化过程，对 DEDS 的控制归结为禁止不期望事件的发生或使事件按期望的时序发生。

在实际系统中有许多可以用 DEDS 来描述。例如一间只有一个理发师的理发店，在正常工作的时间内，如果没有顾客到达理发店，则理发师空闲;如果有顾客到达理发店，则理

发师为顾客进行理发服务。如果顾客到达理发店时，理发师正在为其他顾客服务，则新来的顾客在一旁排队等待。显然，每个顾客到达理发店的时间是随机的，而理发师为每个顾客进行理发服务的时间也是随机的，从而每个队列中的顾客等待的时间也是随机的，这是一个典型的离散事件系统的例子。

在现代高技术领域中出现的大量人造系统大多数也可以用 DEDES 来描述。例如通讯网络中的呼叫接入系统，其中一个用户呼叫的到来和结束就是两个离散的事件，而某一时刻所有不同的服务节点中接入的不同形式的用户数目就构成了一个状态。如果考虑缓冲器，则还需要计入不同缓冲器中的不同的用户数目。

如图 1.1 所示的柔性生产线或称柔性制造系统，简称为 FMS (flexible manufacturing systems)，也是最具典型性和重要性的一个例子。在计算机控制单元的控制下，不同的待加工工件通过自动物料传送系统输送到相应加工中心的缓冲区，由加工中心对工件进行多类型的加工，从刀具的控制到加工工艺的确定都由计算机控制程序决定和控制，且程序的调用和更改非常方便灵活，因此具有很好的柔性。加工完毕的工件通过自动物料传送系统输送到自动仓库。

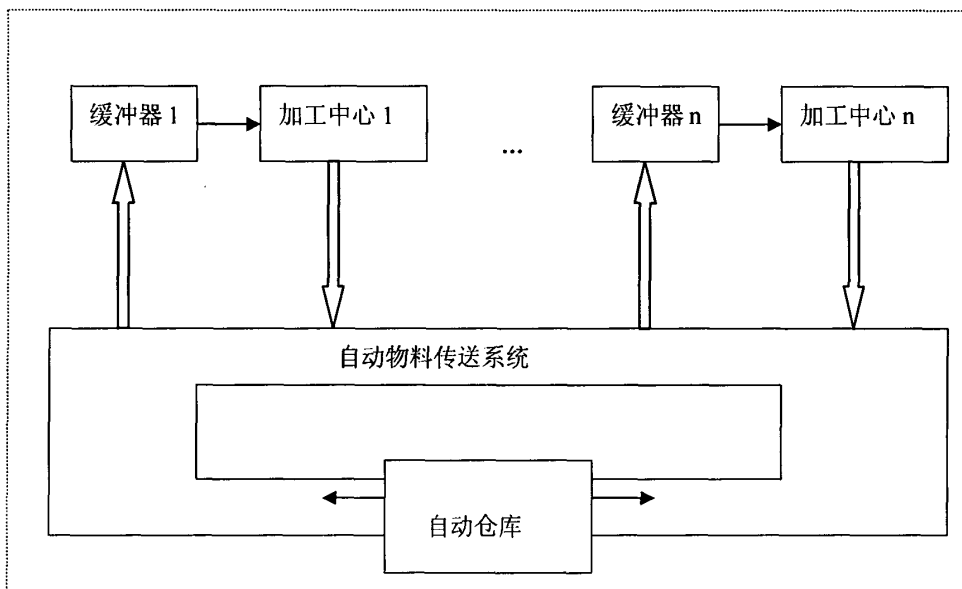


图 1.1 柔性制造系统示意图

这样的柔性制造系统是一个典型的离散事件动态系统。在 FMS 中，各个加工中心对各类工件的加工活动构成系统的状态，由工件和加工中心组成系统的资源，资源的投入或释放构成为系统的离散事件。表征系统加工活动的状态的跃变，由待加工工件的到达(投入)和机床的完成加工(释放)等事件所驱动。显然，状态演化过程中，状态跃变时刻将呈现出异步性，而系统演化则由离散事件错综复杂的相互作用所决定。当 FMS 的基本参数(如加工中心对各种工件的作业时间、自动物料传送系统的运行速度、工件的加工特性等)为固定不变时，FMS 中出现的事件可认为是确定性的，整个 FMS 可表征为一个确定性 DEDES。如果 FMS 的基本参数(如机床作业时间、自动物料传送系统的运行速度等)受内部和外部因素影响而具有不确定性时，FMS 中的事件在本质上是随机性的，FMS 相应地需要采用随机 DEDES 来描述。

基于 FMS 的 DEDES 模型，可用来确定对待加工工件的排序，分析加工过程的加工节奏，避免 FMS 出现阻塞现象，优化配置各个缓冲区的容量，以及优化系统的生产率等。

1.1.2 DEDS 的研究方法

DEDS 的研究, 根据所用模型和采用工具的不同, 基本上可以分为三个不同的方向, 即逻辑层次, 时间层次, 及随机统计层次。采用不同的方式来描述 DEDS 中的事件和状态这两个基本要素, 并通过它们运用不同的手段来研究同一个过程, 从而决定了这三种模型及其研究方法。在对 DEDS 的研究中各有其侧重点, 因而它们也各有其优缺点。

在逻辑层次模型下, 采用的系统模型主要有形式语言, 有限自动机, Markov 链, Petri 网[13-18]等。在该层次模型下, 一般用一些离散的符号来表征系统的状态, 并且不要求状态空间(事件的集合)有任何的拓扑结构。运用人们规定的某种机制来驱动事件的产生和灭亡, 从而导致系统状态的转移。因此, 从逻辑层次研究 DEDS, 就是研究事件和状态按照逻辑时间序列运行, 而不涉及系统的真实物理时间。故其采用的研究手段, 大多是一些逻辑性较强的工具, 例如有限自动机, Petri 网等, 在逻辑层次下, 对系统的控制及控制作用下的系统行为有较深刻的研究, 但由于其未考虑系统的物理时间, 故难以用于连续时间 DEDS 的动态分析。

在时间层次模型下, 采用的主要建模方法包括, 极大极小代数, 有限递归过程等, 而基于极大极小代数或更一般的双子代数的研究, 由于其特殊的运算规则, 导致系统的动态方程非常类似于通常连续变量线性系统。在这一层次下, 不仅考虑事件和状态的逻辑顺序关系, 而且也涉及系统真实的物理时间, 从而可以用于系统的动态分析。特别是, 基于线性化的动态方程, 可以定量的研究系统的稳定性及结构特征等系统行为。但其主要缺陷是适用的范围较窄[19], 并且通常连续变量线性系统中一系列行之有效的分析, 控制和优化技术难以应用到这样的线性系统模型中。

在随机统计层次模型下, 采用的系统模型主要是排队网络和随机过程。一般来说, DEDS 中某个参变量的微小变化, 都会导致系统以十分不同的方式运行, 因此需要考虑系统中许多参变量的随机变化, 从而在性能分析中需要采用统计平均的方法。从统计性能层次研究 DEDS, 也是 DEDS 研究的最初形式, 何毓琦(Y. C. Ho)教授等人提出 DEDS 这个概念时就是从该层次出发的。在这一层次下, 采用的系统模型主要为 Markov 过程、半 Markov 过程、广义半 Markov 过程和各种类型的排队网络等, 其中有代表性的研究方法是摄动分析(PA), 无穷小摄动分析(IPA), Markov 决策(控制)过程等。

总体而言, 现在对离散事件动态系统建模的研究, 还远不是成熟的和完善的。不管是从形式的统一性, 还是从数学表达的简明性和计算分析的可行性, 都远不如连续变量动态系统的建模那样完美。更广义地说, 由于 DEDS 属于人造系统的范畴, 系统机制中可能会同时并存多种交互作用, 如事件之间的交互作用、人与系统的交互作用、系统与环境的交互作用等, 系统各种关系中也可能会同时含有多种表达形式, 如定 t 、定性和知识的形式等。因此, 一个复杂的离散事件动态系统建模, 最终可能需要同时借助于运筹学、系统与控制理论、人工智能与自然语言理解等多学科的方法的结合。一般认为, 如图 1.2 所示那样, DEDS 的模型定位于这三个学科之间的交叉部分。

1.2 半 Markov 控制过程和性能势

Markov 控制过程, 又称为 Markov 决策过程、Markov 决策规划、Markov 动态规划和受控 Markov 过程等, 是研究类随机序贯决策问题的理论, 即在系列相继的或连续的时刻(称

之为决策时刻)点做出决策,在各个决策时刻点,决策者根据观察到的状态从可用的若干个决策中选择一个;将决策付诸实施后,系统将获得与所处状态和所采取决策有关的一项报酬(或费用等)并影响系统在下一决策时刻点所处的状态。系统在下一决策时刻点处的状态是随机的。在这一新的决策时刻点上,决策者要观察系统所处的新的状态(即收集新的信息)并采取新的决策,如此一步一步进行下去。在每一决策时刻采取的决策不仅影响当前决策周期的运行和性能(状态的逗留时间,获得的报酬或代价等),而且会直接或间接影响下决策时刻系统的运行和性能,并以此影响将来。决策的目的是选择一种控制决策方案,使系统的运行在某种意义上(称为性能准则)达到最优,例如在期望平均报酬(或期望平均代价)准则下,使系统的期望平均总报酬(或期望平均代价)最大(或最小)。这样,在每一个决策时刻所选择的控制决策,不但要使当前决策周期获得的报酬(代价)尽可能大(小),而且要使系统在将来的后续周期所获得的报酬(代价)尽可能大(小)。这需要决策者从全局的观点做出权衡,因而这也是 MCP 优化控制问题的难点所在.MCP 实际上是随机型离散事件动态系统的唯一的动态控制方法,与离散事件动态系统的逻辑控制方法也有着密切的关系。其基本模型有离散时间 Markov 控制过程(DTMCP)和连续时间 Markov 控制过程(CTMCP)。从性能上来说,MCP 又可分为折扣准则 MCP 和平均代价 MCP 等。

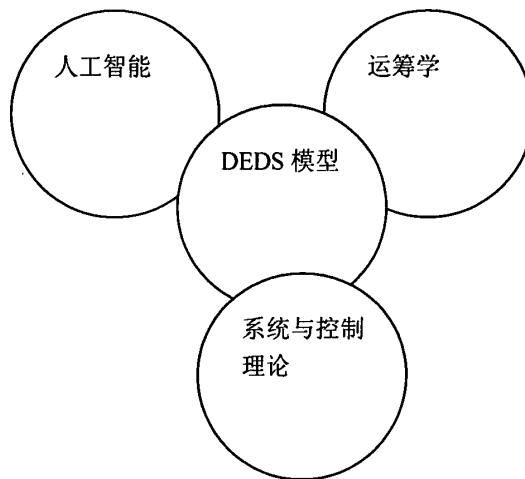


图 1.2 DEDS 与其它学科的关系

MCP 的优化方法通常有数值计算求解、直接搜索法、线性规划方法(LP)、梯度方法、以及策略迭代(PI)和数值迭代(vi)方法等。目前后三种方法是主要的优化手段。对于 MCP 的优化目前国际上已经有诸多理论和应用成果问世[31-35],国内在其优化理论和应用方面的研究,也有了诸多成果[36-40]。对有限行动集或可数行动集、紧致行动集,折扣准则下以及平均性能准则下的优化问题都有了很多研究成果。本文在 MCP 基于性能势的最优性原理和最优化方程的基础上考虑一些优化算法,如策略迭代算法以及数值迭代算法。

在实际的优化问题中,系统模型可以是已知的,也可以是未知或者不完全知道的。在系统模型已知时,可以根据模型参数直接进行优化计算。然而在状态空间很大的优化计算问题中,存在两个问题。一是直接的优化计算量常常相当大,尤其是矩阵求逆运算等,需要耗费大量的运算时间;二是在一定的策略下一部分状态出现的概率很小或者几乎不出现,

对优化问题的贡献很小,从而耗费了不必要的计算时间。为了降低优化问题的成本,我们可以考虑基于仿真的算法,通过仿真随机产生出具有代表性的状态,然后对这些状态进行计算,这样避免了每次都要对那些很少出现的状态进行计算,因而节省了计算时间。另外由于性能势的一个特点就是可以在一条单样本轨道上得到其无偏估计值,并用于进一步的优化,因此在基于性能势的优化问题中也要考虑基于仿真的优化方法。

在系统模型未知或者不完全知道时,也必须求助于仿真的方法。首先需要对系统的模型参数进行估计和改进以期得到和符合实际系统的优化结果,而在基性能势的优化算法中,不仅要对系统的模型参数进行估计,还要对性能势进行估计。

二十世纪九十年代,曹希仁(X.R.Cao)教授和陈翰馥教授提出了一个新的理论:Markov性能势理论[27],并揭示了Markov性能势、无穷小摄动分析(IPA)与Markov决策过程三者之间的联系[411]。这一成果被广泛应用到一般Markov系统和排队网络的灵敏度分析中,并取得了显著成果[27-28. 41-48];另一方面,曹希仁(X.R.Cao)运用Markov性能势理论,给出了MCP在有限行动集下基于性能势的优化理论和算法[41, 42]。在此基础上,我们进一步建立了Markov控制过程和受控排队网络在紧致行动集上基于性能势的优化理论和算法[49-53]

事实上,性能势理论为MCP的优化提供了一个统一的基本的理论框架。运用性能势理论,从Poisson方程着手,在较少的假设条件下,很容易建立起MCP基于性能势的最优性原理和最优性方程,并且容易证明其最优解的存在性定理。在此基础上可给出一些收敛的优化算法,如基于计算的梯度类算法和策略迭代以及数值迭代算法[40, 49-54]。但是这种基于理论计算的优化方法,依赖于系统的精确模型,往往需要系统的转移矩阵或无穷小矩阵的完全信息及有关矩阵的求逆运算(如由Poisson方程求解性能势)。对于一个实际系统,其模型可能未知或不全知,那种需要转移矩阵或无穷小矩阵的理论算法已经不再适用;另一方面,对那些存在“维数灾难”问题的大规模实际系统,即使模型已知,由于其状态空间很大,在进行优化计算时,有关矩阵求逆运算也需要大量计算机内存,占用相当长的计算时间,往往导致计算机负担过重而不可行。而性能势的一个重要特点是可以通过仿真或观测一个实际系统的运行,得到单个样本轨道来进行无偏估计[28],在此基础上可以发展基于单条样本轨道的仿真优化算法和有关在线优化算法。这种算法将适用于大规模实际系统在模型已知或未知情况下的优化求解。

从实际应用的角度出发,需要对基于单样本轨道的优化算法效率做进一步的优化。考察性能势的仿真算法,可以发现其具有很好的并行性,而且基于性能势的策略迭代算法和数值迭代算法都易于进行并行处理。这样在理论计算优化方法和仿真优化方法的基础上可以进一步发展一种新的高效并行理论算法和并行仿真算法[55-57],因而能大大节省计算时间,从而为大规模实际系统的优化提供一条途径。这也是本文的主要内容之一。

性能势的这些性质为实际Markov系统的性能分析和优化提供了一个强有力的工具,它在实际系统中的应用突出体现在排队网络中。实际系统中的许多排队网络都可以用Markov控制过程来描述,我们称这样的排队网络为Markov型排队网络。本文也把有关排队网络的研究作为重要内容。

目前,离散事件动态系统在随机统计层次下是主要以Markov过程为研究模型的,该类模型的性能灵敏度分析以及性能优化问题已经有了很多成果。但是由于半Markov过程的逗留时间服从一般分布,因而半Markov过程是比Markov过程更为广泛的一类系统,更接近实际系统。所以半Markov控制过程的相关问题逐渐成为近年来DEDS领域的一个研究热点问题。半Markov控制过程(SMCP)是一类受到一系列控制决策驱动的非Markov系统,其状态转移规律和控制决策所采用的行动方案相互作用决定了系统的演化,过程在每个状态的逗留时间服从一般分布。并且该分布不仅与当前状态有关,而且与将要转移到的

下一个状态有关。很明显，如果该分布是指数分布，则就是 Markov 控制过程，那么，我们可以看出，半 Markov 控制过程是一类比 Markov 控制过程更广泛的一类系统。

本文将重点研究半 Markov 控制过程的相关理论，并给出一些实际例子。

1.3 仿真及并行计算

计算机仿真方法已经成为分析复杂系统的重要手段，这一方法能够为各种类型的实际问题提供数值解。利用仿真手段可以代替某些费用昂贵的试验，并且能够更快地积累数据，以便选择合适的假设及检验预测的结论，并检验有关理论中假设的合理性。在实际仿真中运算速度常常是考虑的重点之一，因此除了理论本身的改进之外需要采取并行计算的方法减少运算时间满足实际仿真的需求。

1.3.1 仿真

上面已经说明基于性能势来解决 Markov 控制系统优化问题可以通过理论方法，计算出性能势的理论值进行优化，但这种方法涉及到矩阵求逆等运算，对于大规模系统则力不从心。而基于性能势的优化方法的一个重要优点就是可以通过观测或仿真系统的一条单样本轨道得到性能势的无偏估计，因而适用于大规模实际系统。不同于传统的基于直接计算的优化方法，在这种基于仿真的优化方法中，稳态概率和性能势都是根据系统的一条样本轨道信息来估计计算的[28, 58]。

所谓系统仿真是以系统理论、形式化理论、随机过程与统计学理论和优化理论为基础，以计算机和仿真系统软件为工具，对现实系统或未来系统进行动态实验研究的理论和方法[59-61]

从系统仿真的实施过程来看，系统仿真是通过对所研究系统的认识 and 了解，抽取其中的基本要素的关键参数，建立与现实系统相对应的仿真模型，经过模型的确认和仿真程序的验证，在仿真实验设计的基础上，对该模型进行仿真实验，以模仿系统的运行过程，观察系统状态变量随时间变化的动态规律性，并通过数据采集和统计分析，得到被仿真系统参数的统计特性，据此推断和估计系统的真实参数和性能测度，为辅助决策提供依据。

系统仿真的一般步骤如图 1.3 所示[61]

1.3.2 并行计算及其在仿真中的应用

由于并行计算在本文中有重要应用，因此我们介绍有关概念并根据相关研究的特点选择合适的硬件和软件。

1.3.2.1 并行计算的背景

简单的讲,并行计算(Parallel Computing)就是在并行计算机上所做的计算[62-64],它和常说的高性能计算(High Performance Computing)、超级计算(SuperComputing)是同义词,因为任何高性能计算和超级计算总是离不开使用并行技术。随着计算机和计算方法的飞速发展,几乎所有的学科都走向定量化和精确化,从而产生了一系列诸如计算物理、计算化学、计算生物学、计算地质学、计算气象学和计算材料科学等的计算科学,在世界上逐渐形成了一门计算性的学科分支,即计算科学与工程,简称为 CSE (Computational Science & Engineering)。在许多情况下,或者是理论模型复杂甚至理论尚未建立,或者试验费用昂贵甚至试验无法进行,此时计算就成为求解问题的唯一或主要手段.计算极大地增强了人们从事科学研究的能力,大大地加速了把科技转化为生产力地过程,深刻地改变着人类认识世界和改造世界的方法和途径。计算科学的理论和方法,作为新的研究手段和新的设计与制造技术的理论基础,正在推动着当代科学与技术向纵深发展。

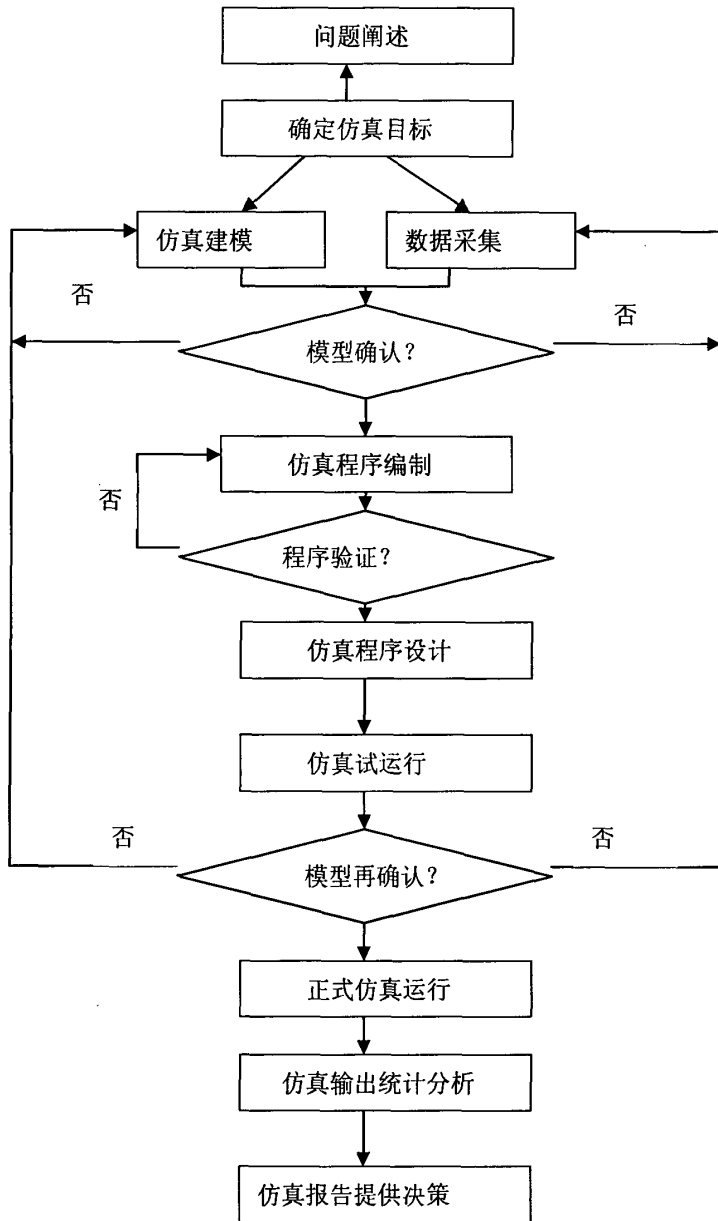


图 1.3 系统仿真的步骤

科学和工程计算中的重大挑战性课题，要求能提供 1 Tflops 计算能力、1TB 主存容量和 1 TB/s 的 I/O 带宽，这就是所谓的 3T 性能目标。而在 HPC 计划提出的当时，性能最好的计算机与 3T 指标相比，速度太慢，而存储容量太小，带宽太窄。面对这些要求，只有采用并行计算的技术，才能够在要求的时间内解决相同的问题，或在相同的时间内解决更多更复杂的问题，以较低的投入完成串行计算的任务。另外对于今后越来越高的计算速度要求，不能依赖物理设备的进化，光速是不可逾越的速度极限，设备和材料也不可能做得无限小，处理器本身的速度不能够无限提高，只有通过并行才能够不断提高速度，满足越来越高的性能要求。

由于并行计算是本文工作的重点，因此下面详细介绍一下有关的概念。

1.3.2.2 并行计算机的分类

为了利用计算机实现相关算法,使所用的高级语言能够被有效的编译并且能够用硬件来实现,必须有一种计算模型作为硬件和软件之间的桥梁以充分设计分析算法。在串行计算时,冯·诺依曼机就是一个理想的串行计算模型,在此模型上硬件设计者可以设计多种多样的冯·诺依曼机而无需虑及将要在其上执行的软件;而软件工程师能够编写各种可在此模型上有效执行的程序而不需考虑所使用的硬件。但在并行计算时尚未有一个类似于冯,诺依曼机地真正通用的并行计算模型,因此对于不同的并行计算机硬件结构,在其上的软件编制工作都有相当大的不同。在我们的并行计算工作中也必须根据所采用的并行计算机硬件结构特点,分析如何发展尽量通用和高效的算法,这也是本文工作的重点之一。

在考虑并行算法之前首先需要熟悉并行计算机的各种分类及其特点[62, 64]。根据一个并行计算机能够同时执行的指令与处理的数据的多少,可以把并行计算机分为 SIM1) (single-instruction multiple-data, 单指令多数据并行计算机)和 MIMD (multiple-instruction multiple-data, 多指令多数据并行计算机)两种

按同时执行的程序和数据的不同,人们又提出了 SPMD (single-programmultiple-data, 单程序多数据并行计算机)和 MPMD (multiple-programmultiple-data, 多程序多数据并行计算机)的概念。这种划分方式所依据的执行单位不是指令而是程序,显然其划分粒度(划分对象的大小程度)要大得多。

根据并行计算机结构模型的不同,一般可以分为六类:单指令多数据流机 SIMD (Single-Instruction Multiple-Data)·并行向量处理机 PVP (Parallel VectorProcessor)、对称多处理机 SMP (Symmetric Multiprocessor)、大规模并行处理机 MPP (Massively Parallel Processor), I 作站机群 COW (Cluster of Workstation)和分布式共享存储 DSM (Distributed Shared Memory)多处理机。

从物理划分上,共享内存和分布式内存是两种基本的并行计算机存储方式。除此之外,分布式共享内存也成为一种越来越重要的并行计算机存储方式。具体可分为下面几类:均匀存储访问 UMA (Uniform Memory Access),非均匀存储访问 NUMA (Nonuniform Memory Access)、全高速缓存存储访问 COMA (Cache-Only Memory Access)、高速缓存一致性非均匀存储访问 CC-NUMA (Coherent-Cache Nonuniform Memory Access)和非远程存储访问 NORMA (No-Remote Memory Access)。

对于共享内存的并行计算机,各个处理单元通过对共享内存的访问来交换信息,协调各处理器对并行任务的处理。这种共享内存的编程实现起来相对简单,但共享内存往往成为性能特别是扩展性的重要瓶颈。

对于分布式内存的并行计算机,各个处理单元都拥有自己独立的局部存储器。由于不存在公共可用的存储单元,因此各个处理器之间通过消息传递来交换信息,以协调和控制各个处理器的执行。这也是本文采用的消息传递并行编程模型所面对的并行计算机的存储方式,不难看出,通信对分布式内存并行计算机的性能有重要的影响,复杂的消息传递语句的编写成为在这种并行计算机上进行并行程序设计的难点所在,但是,对于这种类型的并行计算机,出于它有很好的扩展性和很高的性能,因此它的应用非常广泛。

本文的并行工作的硬件基础是国家高性能中心(合肥)曙光 2000 并行计算机[65-66]曙光 2000 基于分布式存储机群系统和消息传递体系结构,是通用的可扩展超级服务器系统。系统的节点数目可以从 4 个扩展到 128 个,节点 CPU 采用 Motorola 公司的 Power PC604e 微处理器,节点之间通过 10/100Mbps 高速以太网和 100MB/s 的“蛀洞”(Wormhole)路由芯片组成的二维网络互连。在 32 个计算节点(单 CPU)缺省配置下,整机峰值速度达到

200GFLOPS(即每秒 200 亿次浮点运算), 内存总容量达到 8GB, 磁盘总容量达到 144GB. 曙光 2000 的节点运行完整的 IBM AIX 操作系统, 可以运行上千种 AIX 应用程序和数据库、网络服务器等商用软件, 支持 C, Fortran, Java 等程序设计语言, 支持 ESSL, BLAS, Scalapack 等高效数学和工程库。系统软件提供 BCL(Basic CommunicationLibrary)底层通讯机制、定制的 PVM 和 MPI 并行程序设计环境、集成化并行程序设计环境、并行调试器、基于 Web 的友好界面、自动并行化工具、COSMOS 可扩展文件系统、对高可用性的支持、作业管理、资源管理、系统管理、AutoPAR 自动并行化工具、服务器聚集软件, 以及 DSM, HPF, ParaVT 等其它工具。广泛采用 JAVA 和面向对象方法, 在设计上注重可用性要求。

1.3.2.3 并行算法模型及并行语言的选择

一个物理问题并行求解的最终目的是将该问题映射到并行机上通过不同层次上的抽象映射来实现的。如图 1.4 所示。

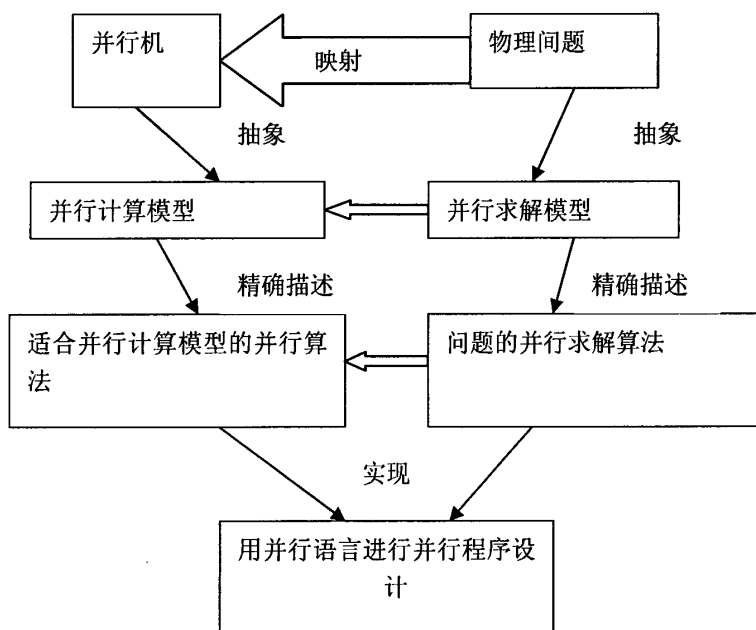


图 1.4 问题的并行求解过程

忽略并行机的非本质的细节特征, 可以得到该并行机的并行计算模型。在这一模型上, 可以设计各种适合该模型的并行算法, 这些算法精确描述了该并行模型能够实现的功能, 而这些算法是通过用特定的并行语言设计并行程序后得以实现。对于现实世界的物理问题, 为了能够高效地并行求解, 必须建立它的并行求解模型, 一个串行的求解模型是很难在并行机上取得满意的并行效果的。有了并行求解模型, 就可以针对该模型设计高效的并行算法, 对该问题的求解进行精确描述和定量分析, 对各种算法进行性能上的比较, 最后通过并行程序设计, 实现问题和并行机的结合。

并行程序设计需要将问题的并行求解算法转化为特定的适合并行计算模型的并行算法。为了达到这一目的, 首先是问题的并行求解算法必须能够将问题内在的并行特征充分体现出来, 否则并行求解算法将无法利用这些并行特征, 从而使问题的高效并行求解成为

不可能;其次是并行求解模型要和并行计算模型尽量吻合,这样,就为问题在并行机上的高效解决提供了前提。

目前两种最重要的并行编程模型是数据并行和消息传递[64]。数据并行编程模型的编程级别比较高、编程相对简单,但它仅适用于数据并行问题;消息传递编程模型的编程级别相对较低,但消息传递编程模型可以有更广泛的应用范围。数据并行即将相同的操作同时作用于不同的数据,因此适合在 SIMD 及 SPMD 并行计算机上运行。在向量机上通过数据并行求解问题的实践也说明,数据并行可以高效地解决一大类科学与工程计算问题。

消息传递即各个并行执行的部分之间通过传递消息来交换信息、协调步伐、控制执行,消息传递一般是面向分布式内存的,但是它也可适用于共享内存的并行机。消息传递为编程者提供灵活的控制手段和表达并行的方法,一些用数据并行方法很难表达的并行算法,都可以使用消息传递模型来实现。灵活性和控制手段的多样化,是消息传递并行程序能提供高的执行效率的重要原因。

消息传递模型一方面为编程者提供了灵活性,另一方面,它也将各个并行执行部分之间复杂的信息变换和协调、控制的任務交给了编程者,在一定程度上增加了编程者的负担,这也是消息传递编程模型编程级别低的主要原因:虽然如此,消息传递的基本通信模式是简略和清楚的,学习和掌握这些部分并不困难,因此目前大量的并行程序设计仍然是消息传递并行编程模式。本文的并行工作考虑到为了尽可能扩大程序的适用面,因此采用消息传递模型。

本文考虑到并行语言的使用广泛性和并行程序设计的通用性,另外由于 C 语言在工程和科学计算中的广泛应用,因此选择 MPI (Message Passing Interface)并行标准的 C 语言版本实现并行算法 [64]

MPI 是一个库,而不是一门语言。可以把 C+MPI 看作是一种在原来串行语言(C 语言)基础之上扩展后得到的并行语言。MPI 库可以被 C/C++调用,从语法上说,它遵守所有对库函数过程的调用规则,与一般的函数过程没有什么区别。MPI 是一种标准或规范的代表,而不特指某一个对它的具体实现。迄今为止,所有的并行计算机制造商都提供对 MPI 的支持,可以在网上免费得到 MPI 在不同并行计算机上的实现,一个正确的 MPI 程序,可以不加修改地在所有的并行机上运行。MPI 是一种消息传递编程模型,并成为这种编程模型的代表和事实上的标准。MPI 虽然很庞大,但是它的最终目的是服务于进程间通信这一目标的。在 MPI 上很容易移植其他的并行代码,而且编程者不需要去努力掌握许多其他的全新概念,就可以学习编写 MPI 程序。正是由于 MPI 的这些特点,促使本文的并行算法都采用 MPI 来实现。

1.4 文章结构安排

本文主要研究具有数状态空间的半 Markov 控制过程在无穷水平平均代价准则下的性能灵敏度分析和优化问题。给出了以等价 Markov 过程和嵌入 Markov 链的性能势表示的灵敏度公式,建立了优化理论,给出了最优性方程并证明了解的存在性定理;研究了优化算法,给出了基于单个样本轨道的仿真优化算法及其并行算法。并给出了大量的数值例子,来说明相应算法的应用。文章的结构安排如下:

第二章中建立了半 Markov 过程的性能势理论,并在性能势的基础上讨论了稳态性能势灵敏度分析问题。首先利用等价无穷小矩阵,定义了半 Markov 过程的 Poisson 方程,将该 Poisson 方程的解定义为半 Markov 过程的性能势;然后利用等价 Markov 过程和嵌入 Markov 链的方法,讨论了半 Markov 过程的性能灵敏度分析问题,导出了基于等价 Markov

过程和嵌入 Markov 链的性能势的灵敏度公式。并对其进行了串行和并行的仿真。

第三章讨论了一类可数状态空间的半 Markov 控制过程的性能优化问题, 研究半 Markov 控制过程基于直接计算的理论优化方法, 讨论了半 Markov 控制过程基于样本轨道仿真的优化方法。

第四章, 我们给出了通讯中两个实际网络的优化与导数估计的应用实例. 一是 Markov 决策过程方法在呼叫接入控制中的应用; 二是 M/PH/1 排队系统的性能灵敏度估计与仿真及其在 ATM 交换机性能分析中的应用。

最后对本文工作做了一个总结, 指出了进一步的研究方向。

事实上, 许多实际人造系统, 例如柔性制造系统中自动车的移动策略, 物流系统中的最小代价路径问题, 大规模计算机和通信网络中的多媒体信流的接收和发送, 系统的最优维护代价和安全可靠性等问题, 最终均可归结为一个半 Markov 控制过程问题。本论文的研究能够为大量实际复杂人造系统的性能优化提供一个直接的、行之有效的方法, 这对进一步改进系统的设计, 提高系统的运行效率和管理水平具有极其重要的意义。

第二章 半 Markov 性能势及其仿真

2.1 半 Markov 性能势理论

在实际系统中,我们还常常会遇到非 Markov 型排队网络的情况,与 Markov 过程相比,半 Markov 过程或广义半 Markov 过程具有更广泛的应用范围。

2.1.1 半 Markov 过程

定义 1 若状态空间 $\Phi = \{1, 2, \dots, K\}$ 上的随机过程 $N = \{N_t; t \geq 0\}$ 满足下列条件:

(1) 对任意的 $i, j \in \Phi$, 若已知过程处于状态 i , 则它下一次转移到状态 j 的概率是 p_{ij} ,

并设 $p_{ii} = 0$.

(2) 若已知过程处于状态 i 和下一次将转移到状态 j , 则它在状态 i 的逗留时间的分布为 F_{ij} 。

则称 $N = \{N_t; t \geq 0\}$ 为一个半 Markov 过程(SMP)。

定义 2 设 $N = \{N_t; t \geq 0\}$ 为状态空间 $\Phi = \{1, 2, \dots, K\}$ 上的随机过程, $0 = t_0 < t_1 < t_2 \dots$ 是该过程发生状态转移的时刻, 则 $X_n = N_{t_n+0}$ 是过程在时刻 t_n 发生的第 n 次转移后所处的状态。若对任意的整数 $n \geq 0$, 任意的 $i_0, \dots, i_{n-1}, i, j \in \Phi$ 和任意的实数 $t \geq 0$, 均有

$$\begin{aligned} & P(X_{n+1} = j, t_{n+1} - t_n \leq t | X_k = i_k, k = 0, 1, \dots, n-1, X_n = i; t_1, \dots, t_n) \\ & = P(X_{n+1} = j, t_{n+1} - t_n \leq t | X_n = i) = Q_{ij}(t) \end{aligned} \quad (2.1)$$

则称随机过程 $\{(X_n, t_n); n \geq 0\}$ 为 Markov 更新过程 (MRP); 转移概率族 $\{Q_{ij}(t), i, j \in \Phi\}$ 或矩阵 $[Q_{ij}(t)]$ 称为空间 Φ 上的半 Markov 核。

显然, 当 N 为 SMP 时, 对应的 $\{(X_n, t_n); n \geq 0\}$ 是一个 MRP。反之, 若 $\{(X_n, t_n); n \geq 0\}$ 为一个 MRP, 则由

$$N_t = X_n, \quad t_n \leq t < t_{n+1} \quad (2.2)$$

定义的随机过程 $N = \{N_t; t \geq 0\}$ 为一 SMP。称 $X = \{X_n; n \geq 0\}$ 为半 Markov 过程的嵌入 Markov 链。另外，以 V_{ii} 表示过程从状态 i 出发，首次回到状态 i 所经历的时间，即 V_{ii} 是更新过程 $\{X_n^i; n \geq 0\}$ 的更新间距。

记

$$p_{ij}(t) = P(N_t = j | N_0 = i) \tag{2.3}$$

$$h_i(t) = 1 - \sum_{j=1}^K Q_{ij}(t) \tag{2.4}$$

下面的引理 1 至引理 3，可参看文献[76]。

引理 1 对任意的 $i, j \in \Phi$ 和 $t \geq 0$,

$$p_{ij}(t) = \delta_{ij} h_i(t) + \sum_{k \in \Phi} \int_0^t p_{kj}(t-x) dQ_{ik}(x) \tag{2.5}$$

这里

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

若记 $p(t) = [p_{ij}(t)]$, $Q(t) = [Q_{ij}(t)]$, $H(t) = \text{diag}(h_1(t), h_2(t), \dots)$, 并设 $Q(t)$ 在 $[0, +\infty)$ 上连续可微，则 (2.5) 式可以用矩阵形式表为

$$p(t) = H(t) + \int_0^t Q'(x) p(t-x) dx \tag{2.6}$$

引理 2 若半 Markov 过程 $N = \{N_t; t \geq 0\}$ 是不可约的，对 $i \in \Phi$, V_{ii} 是非格子变量且 $EV_{ii} < \infty$ ，则极限

$$\pi_i = \lim_{t \rightarrow \infty} p_{ji}(t) \tag{2.7}$$

存在，且与初始状态 j 无关。 π_i 称为过程 N 在状态 i 的稳态概率或极限概率，用

$\pi = (\pi_1, \pi_2, \dots, \pi_K)$ 表示 N 的稳态概率向量。

另外，过程 N 的稳态概率可进一步用下面的方法来计算。

引理 3 在引理 2 的条件下，对 $i \in \Phi$ ，其稳态概率为

$$\pi_i = \frac{p_i ET_i}{\sum_{j=1}^K p_j ET_j} \quad i = 1, 2, \dots, K$$

其中 (p_1, p_2, \dots, p_K) 为其嵌入 Markov 链的稳态概率, 而 $ET_i = \sum_{j=1}^K p_{ij} \int_0^{\infty} t dF_{ij}(t)$, $i \in \Phi$ 为状态 i 的平均逗留时间。

由引理 3, 我们有:

定理 1 若半 Markov 过程 $N = \{N_t; t \geq 0\}$ 满足引理 2 的条件, 则对 $i \in \Phi$, 极限 $\pi_i = \lim_{t \rightarrow \infty} p_{ji}(t)$ 是方程组

$$\begin{cases} \pi A = 0, \\ \pi e = 1 \end{cases} \quad (2.8)$$

的唯一解, 其中 $A = \psi^{-1}(P - I)$, $\psi = \text{diag}(ET_1, ET_2, \dots, ET_K)$, $P = [P_{ij}]_{K \times K}$, I 为单位阵, e 为 K 维列向量, 它的每个分量均为 1.

证明: 因为嵌入链的稳态概率 (p_1, p_2, \dots, p_K) 满足 $(p_1, p_2, \dots, p_K)(P - I) = 0$, 由引理 3, $\pi A = \pi \psi^{-1}(P - I) = (p_1 ET_1, p_2 ET_2, \dots, p_K ET_K) \psi^{-1}(P - I) / \sum_{j=1}^K p_j ET_j$

$$\begin{aligned} &= (p_1, p_2, \dots, p_K) \text{diag}(ET_1, ET_2, \dots, ET_K) \psi^{-1}(P - I) / \sum_{j=1}^K p_j ET_j \\ &= (p_1, p_2, \dots, p_K) \psi \psi^{-1}(P - I) / \sum_{j=1}^K p_j ET_j = (p_1, p_2, \dots, p_K)(P - I) / \sum_{j=1}^K p_j ET_j = 0, \end{aligned}$$

显然 $\pi e = 1$, 另外, 在该定理的条件下, $\text{rank}(A, e) = K$, 证毕。

2.1.2 半 Markov 过程性能势

考虑上一节所讨论的半 Markov 过程 $N = \{N_t; t \geq 0\}$, 状态空间为 $\Phi = \{1, 2, \dots, K\}$, 其半 Markov 核为 $Q(t) = [Q_{ij}(t)]_{K \times K}$ 。过程 N 存在唯一的稳态概率分布 π_i , $i \in \Phi$ 。设 $f: \Phi \rightarrow R = (-\infty, +\infty)$ 为一个性能函数, 可表示为列向量 $f = (f(1), f(2), \dots, f(K))^T$ 。定义稳态性能 η_f 为 f 关于稳态概率 π 的期望值, 即

$$\eta_f = E_{\pi}[f] = \sum_{i=1}^K \pi_i f(i) = \pi f \quad (2.9)$$

下面我们给出实现因子与实现矩阵的概念。我们用 $\{N_t^{(i)}; t \geq 0\} = \{N_t; N_0 = i; t \geq 0\}$,

$\{N_t^{(j)}; t \geq 0\} = \{N_t : N_0 = j; t \geq 0\}$ 表示分别起始于两个不同初始状态 i 和 j 的过程 N_t 的两个样本, 设上述两条起始于不同初始状态的样本轨道首次达到相同状态的时间为 $T^{(i,j)}$, 显然 $T^{(i,j)}$ 是一随机变量, 进一步我们要求 $T^{(i,j)}$ 的均值有限. 状态 i, j 对应的实现因子 $d_{i,j}$ 指的是: 如果当 $t = 0$ 时, 过程从状态 i 被摄动到 j , 并且在 $t = T^{(i,j)}$ 时, 摄动轨道又回到了原来的状态, 即摄动被实现, 则 $d_{i,j}$ 度量了在此过程中网络性能测度的改变量. 即我们有

定义 3 对任意给定的状态 $i, j \in \Phi$,

$$d_{i,j} = E\left\{\int_0^{T^{(i,j)}} [f(N_t^{(j)}) - f(N_t^{(i)})] dt\right\}. \quad (2.10)$$

称 $d_{i,j}$ 为实现因子. 而所有可能的状态对的实现因子构成的矩阵 D 称为实现矩阵,

$$D = [d_{i,j}]_{K \times K}, \text{ 显然 } D^T = -D.$$

与参考文献[74]的方法一样, 可以证明下面的结果, 它们在基于仿真的梯度估计中具有重要的作用.

定理 2 对任意给定的状态 $i, j \in \Phi$, 我们有

$$d_{i,j} = \lim_{T \rightarrow \infty} \{E[\int_0^T f(N_t^{(j)}) dt] - E[\int_0^T f(N_t^{(i)}) dt]\}. \quad (2.11)$$

上述定理表明, 网络性能测度的改变量等价于(在统计意义上)一摄动周期上的改变量. 下面的定理提供了一个更简单的基于一条样本轨道的仿真来估计 $d_{i,j}$ 的方法. 我们定义从状态 j 到状态 i 的首达时间如下:

$$S^{(j)}(i) = \inf\{t : t \geq 0, N_t^{(j)} = i\}.$$

定理 3 对任意给定的状态 $i, j \in \Phi$, 有

$$d_{i,j} = E\left\{\int_0^{S^{(j)}(i)} f(N_t^{(j)}) dt\right\} - E[S^{(j)}(i)]J(v) \quad (2.12)$$

$$= E\left\{\int_0^{S^{(j)}(i)} [f(N_t^{(j)}) - J(v)] dt\right\}. \quad (2.13)$$

由实现因子的定义及定理 2 的结论, 我们有对任意 $i, j, k \in \Phi$, $d_{i,j} = d_{i,k} + d_{k,j}$. 实现因子的这一性质类似于物理学中势能的性质, 由此我们给出下述有关性能势的定义.

定义 4 如果对每一个状态 $i \in \Phi$, 存在一个量 g_i , 满足

$$d_{i,j} = g_j - g_i.$$

则称 g_i 为网络过程在状态 i 的性能势, 且称 $g = (g_1, g_2, \dots, g_K)^T$ 为性能势向量(简称性能势).

显然, g 满足 $D = eg^T - ge^T$ 。注意到对任意常数 c ,

$$e(g + ce)^T - (g + ce)e^T = eg^T - ge^T = D,$$

故 $g + ce$ 仍是一性能势向量。下面, 我们给出性能势的表达式。

定理 4 半 Markov 过程的性能势具有表达式

$$g = D^+ \pi^T. \quad (2.14)$$

2.2 仿真的相关内容

下面我们介绍一下仿真涉及到的随机变量的产生、样本轨道等相关内容。

2.2.1 仿真的基本概念

本小节主要讨论服从一定分布的随机变量的产生问题[69]

在几乎所有的仿真过程中都需要用服从各种分布的随机变量来描述系统中存在的随机性和偶然性问题, 使仿真中的事件看来象是随机发生的。随机变量的产生基础是随机数, 随机数可以通过对 $[0, 1]$ 区间上均匀分布抽样生成。一般情况下利用计算机产生伪随机数, 进而生成满足一定分布的随机变量。

计算机产生的伪随机数一般服从 $[0, 1]$ 上的均匀分布, 然而在仿真研究过程中, 其他一些理论分布如正态分布、指数分布、泊松分布和 Γ 分布等的使用要比均匀分布多的多。因此需要一种产生随机数的方法能够模仿从任意分布中的抽样, 所有这样的技术都是从产生一个或多个均匀分布的伪随机数开始的。通过应用某种转换方法即可从这些均匀分布的伪随机数中产生非均匀分布的随机变量。

产生随机变量的方法有许多种, 对于给定分布的随机变量, 可根据其特点选择其中的一种或几种方法。系统仿真中最常用的随机变量产生方法有反变换法、函数变换法、组合法和拒绝法。其中反变换法是最常用最简单的方法, 本文的研究也采用这种方法。当分布密度函数 $V(x)$ 可积分得出累积分布函数 $F(x)$ 或 $F(x)$ 是一个经验分布时可以使用该方法。下面就讨论这种方法。

假定我们需要从某个分布上产生一个伪随机数, 这个分布的累积分布函数为 $F(x)$, $F(x)$ 具有累积分布函数的所有性质, 则从以下两步可以得到所需要的伪随机数: 利用已知的随机数产生方法产生一个服从标准均匀分布的随机数。若记 r 为所产生的均匀分布下的随机数, 则所需的非均匀分布下的随机变量为:

$$x_0 = F^{-1}(r)$$

随机变量累积及分布函数 $F(x)$ 的取值范围为 $[0, 1]$ 。现以在 $[0, 1]$ 上均匀分布的独立随机数作为 $F(x)$ 的取值规律, 则落在 Δx 内的样本个数的概率就是 ΔF , 从而随机变量 x 在区间 ΔX 内出现的概率密度函数的平均值为 $\Delta F / \Delta x$, 当 $\Delta x \rightarrow 0$ 时, 其概率密度函数就等于 dF/dx , 符合给定的密度分布函数, 满足正确性要求。

如下例: 设某分布的累积分布函数由下式给出:

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < 1/4 \\ (3x+1)/7, & 1/4 \leq x < 2 \\ 1, & x \geq 2 \end{cases}$$

且产生的均匀分布随机数为 0.1021, 0.2162 和 0.7621, 先将它们转换为上述分布下的随机变量。求出 $F(x)$ 的逆函数即得

$$x = F^{-1}(a) = \begin{cases} a, & 0 \leq a < 1/4 \\ (7a-1)/3, & 1/4 \leq a < 1 \end{cases}$$

于是有

$$x_1 = F^{-1}(0.1021) = 0.1021$$

$$x_2 = F^{-1}(0.2162) = 0.2162$$

$$x_3 = F^{-1}(0.7621) = 1.4449$$

即为由 $F(x)$ 确定的分布下的随机变量。

特别地, 若分布的累积分布函数由下式给出:

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

而 ξ 为所产生的 $(0,1)$ 上的均匀分布的随机数, $t = F^{-1}(\xi) - \frac{1}{\lambda} \ln(1-\xi)$ 为

$F(x)$ 确定的分布下的随机变量, 即 t 满足指数分布。

2.2.2 Monte-Carlo 仿真和样本轨道

Monte-Carlo 仿真对我们研究基于样本轨道的仿真优化算法具有重要作用[70]。设 x 是一个随机变量(向量), 具有均值 m 。Monte-Carlo 仿真的基本思想就是通过仿真产生随机变量(向量) x 的 N 个样本 x_1, x_2, \dots, x_N , 则 x 的均值 m 可以通过下列样本均值来估计

$$M_N = \frac{1}{N} \sum_{k=1}^N x_k, \text{ 令 } M_1 = x_1, \text{ 则有}$$

$$M_{N+1} = M_N + \frac{1}{N+1} (x_{N+1} - M_N)$$

若 N 是给定常数, 且 N 个样本 x_1, x_2, \dots, x_N 是独立同分布(i.i.d)的, 且共同均值为 m , 共

同方差为 σ^2 , 则有

$$E[M_N] = m, \text{Var}(M_N) = \sigma^2/N$$

若 x_1, x_2, \dots, x_N 是独立同分布(ild)的, 且共同均值为 m , 共同方差为 σ^2 , 但这些样本并

不是独立的而是相关的，则

$$E[M_N] = m, \text{Var}(M_N) \neq \sigma^2/N$$

若样本数 N 也是随机变量，且 x_1, x_2, \dots, x_N 关于 N 的条件均值仍为 m ，则有

$$E[M_N] = m。若 x_1, x_2, \dots, x_N 关于 N 的条件均值不再是 m ，则一般有 $E[M_N] \neq m$$$

即样本均值 M_N 是 m 的有偏估计。若在条件 N 下， x_1, x_2, \dots, x_N 独立同分布，且共

$$\text{同方差仍为 } \sigma^2, \text{ 则有 } \text{Var}(M_N) = \sigma^2 E[1/N]$$

样本轨道是系统在某个(些)控制作用下，根据转移概率产生的一系列相继转移的状态及相关信息，分为有限长样本轨道和无限长样本轨道。

样本轨道一般记做 X_1, X_2, \dots, X_N 或 $(X_0, T_0), (X_1, T_1), \dots, (X_N, T_N), \dots$ 其中 T_N 为系统在状态 X_N 的逗留时间。在模型已知的情况下，我们既可以根据模型参数用计算机来仿真模拟一个实际系统，又可以直接观测一个实际系统，得到样本轨道。在模型未知的情况下，只能通过观测实际系统的运行或通过观测实际系统的一个实物模型的运行来得到样本轨道。

当得到样本轨道之后，就可以根据样本信息来估计有关数值，进行系统的性能分析和优化计算，或通过估计有关模型参数，然后利用得到的估计模型来进行相关计算。无论是在模型已知还是模型未知的情况下，当能够通过观测实际系统的运行来得到样本轨道时，一般就可以建立适合实际系统需要的在线仿真优化算法。

2.3 性能势的串行仿真

2.3.1 基于对过程一条样本轨道的仿真估计性能势

本节我们将给出基于对过程一条样本轨道的仿真估计性能势的算法。本算法的基本思想是：通过对系统实际进程的仿真，估计出系统的任意两个状态之间的平均首达时间 $\hat{E}\{S^{(j)}(i)\}$ ($j, i \in \Phi$)，以及任意两个状态之间当时间从 0 到 $S^{(j)}(i)$ 中的网络性能测度 $\hat{E}\left\{\int_0^{S^{(j)}(i,v)} f(N_t^{(j)}) dt\right\}$ ，各状态的稳态概率行向量 $\hat{\pi}$ 。由此可以得到任意两个状态之间实现因子的估计值 $\hat{d}_{i,j}$ ，以及各状态的性能势的估计值 \hat{g} 。

该算法的基本原理及收敛性等结论与参看文献[63]的算法是一样的，为节省篇幅，仅将对半 Markov 过程的性能势的仿真算法给出。

步 1: 给定初始状态 $i = i_0$ 。

for j=1 to K (标志矩阵 ftFlag 记录近期各状态转移是否正在发生)

ftFlag(i_0, j) = 1

end for

步 2: 取一服从 $[0, 1)$ 均匀分布的随机数 ζ , 如 ζ 满足:

$$\sum_{l=1}^{i-1} p_{i,l} \leq \zeta < \sum_{l=1}^j p_{i,l}$$

其中 $\sum_{l=1}^K p_{i,l} = 1$, $p_{i,l}$ 是状态 i 到状态 l 的一步状态转移概率, 则过程的下一个状态为 j 。

步 3: 取一服从 $[0, 1)$ 均匀分布的随机数 ξ , 根据逗留时间的分布函数 $F_{ij}(t)$, 计算当前状态 i 的逗留时间 t_i ^[8]。

$$t_i = F_{ij}^{-1}(\xi) \quad (2.15)$$

对于一般分布函数 $F_{ij}(t)$, 其反函数的表达式常常无法求出, 本文中的算例就是这种情形。这时, 可以采用一定的算法 (如区间对分法等) 寻找一时间 t_i , 使之在一定的精度条件下满足 $F_{ij}(t_i) = \xi$ 。

步 4: 对本步状态转移 $i \rightarrow j$ 进行记录:

stateTime(i) = stateTime(i) + t_i (记录系统在各状态停留时间)

cost = $t_i \times f_i$ (f_i 是系统代价函数向量 f 的第 i 个分量)

for $il=1$ to K

for $jl=1$ to K (ftTemp, ftCostTemp 是两个临时参数矩阵)

ftTemp(il, jl) = ftTemp(il, jl) + $t_i \times$ ftFlag(il, jl)

ftCostTemp(il, jl) = ftCostTemp(il, jl) + cost \times ftFlag(il, jl)

end for

end for

for $il=1$ to K

ftNum(il, j) = ftNum(il, j) + ftFlag(il, j)

ftSum(il, j) = ftSum(il, j) + ftTemp(il, j)

ftCostSum(il, j) = ftCostSum(il, j) + ftCostSum(il, j)

ftTemp(il, j) = 0; ftCostTemp(il, j) = 0; ftFlag(il, j) = 0

end for

for $jl=1$ to K (标志矩阵 ftFlag 记录近期各状态转移是否正在发生)

ftFlag(j, jl) = 1

end for

步 5: 返回步 2 进行下一轮状态转移仿真, 直到循环了指定的仿真次数 N_s 。

步 6: 求稳态概率估值 $\hat{\pi}$ 和性能势估值 \hat{g} :

```

allTime =  $\sum_{l=1}^K$  stateTime(l)    ( 仿真总时间 )
for i=1 to K
     $\hat{\pi}(i) = \text{stateTime}(i) / \text{allTime}$ 
end for
 $\hat{J} = \hat{\pi}f$ 
for i=1 to K
    for j=1 to K
         $\hat{D}(i,j) = \frac{\text{ftCostSum}(i,j)}{\text{ftNum}(i,j)} - \frac{\text{ftSum}(i,j)}{\text{ftNum}(i,j)} \hat{j}$ 
    end for
end for
 $\hat{g} = \hat{D}^r \hat{\pi}^r$ 
    
```

对这种基于实现矩阵估计性能势的算法进行的分析和实际仿真中都可以发现上述算法中运算量主要集中在第四步,为计算实现矩阵,每一步仿真循环中都涉及到大量的矩阵运算,消耗大量仿真时间,在状态空间维数很大的情况下尤其严重。在实际中的应用也因此受到限制。因此尽量减小运算量或缩短仿真时间的重要性也突出起来,这也是本论文要考虑的主要内容之一。本章第四节中我们将利用并行计算的方法估计实现矩阵,进一步得到性能势的估计值,从而缩短仿真时间。下面我们首先考察另一种算法,该算法避开了实现矩阵的估计从而减少大量的矩阵运算,只记录相关的必要信息最终得到性能势的估计值。然而这种算法是以精度的略微降低为代价的。仿真的流程具体如下:(图 2.1)

2.3.2 转化为等价的 Markov 过程

前面我们给出了基于半 Markov 过程一条样本轨道估计其性能势的算法,从本质上来说该算法仅取决于过程的两个基本特性:其一是各状态间的一步转移概率 p_{ij} , 第二是各状态

的平均逗留时间 $ET_i = \sum_{j=1}^K p_{ij} \int_0^{\infty} t dF_{ij}(t)$, $i, j \in \Phi$ 。如果我们能够构造一个满足这两条基

本特性的 Markov 过程,则该 Markov 过程的性能势与原半 Markov 过程的性能势应该是完全一致的。

显然,若一个 Markov 过程状态 i 逗留时间的分布函数为 $1 - \exp(-\mu_i t)$, 其中,
 $\mu_i = 1 / ET_i$, $i = 1, 2, \dots, K$, 各状态间的一步转移概率为 p_{ij} , $i, j \in \Phi$, 则该 Markov 过程

满足上述要求。而该 Markov 过程的性能势可由下面的公式求得：

$$g = -A^* f. \quad (2.16)$$

其中， $A^{\#} = (A + e\pi)^{-1} - e\pi$ 是该 Markov 过程无穷小生成矩阵 A 的群逆，而 $A = \Lambda(P - I)$ ， $\Lambda = \text{diag}(\mu_1, \mu_2, \dots, \mu_K)$ ， $P = [P_{ij}]_{K \times K}$ ， I 为单位阵， e 的意义同前， π 是该 Markov 过程的稳态概率行向量（定理 1 保证了它也是原半 Markov 过程的稳态概率行向量）。

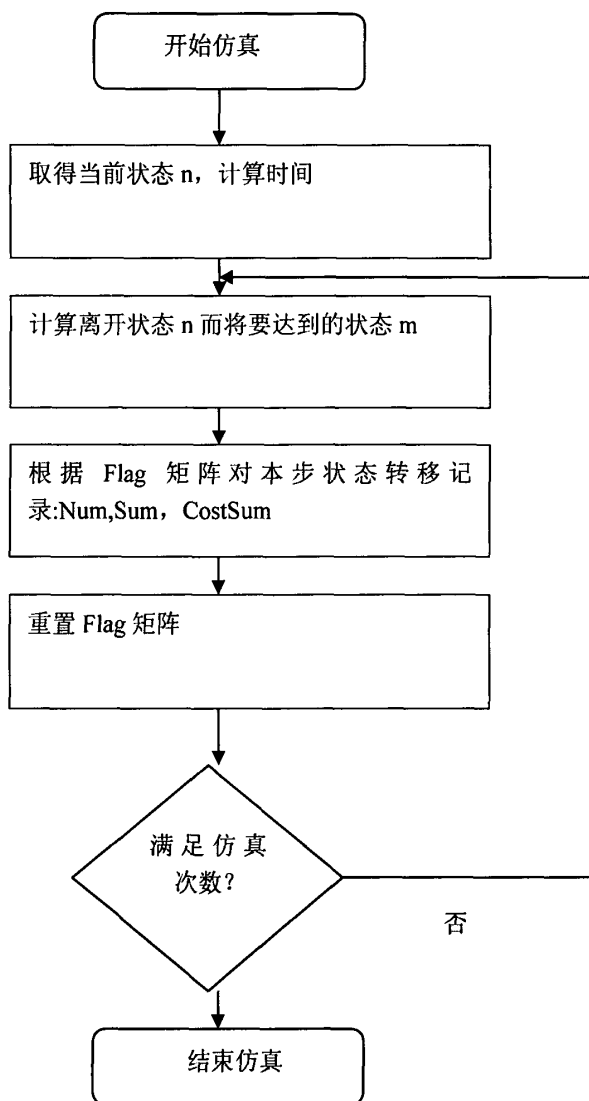


图 2.1 仿真流程图

例 考虑一个具有有限状态空间的半 Markov 过程, $K=10$, 状态间的一步转移概率矩阵 $P = [P_{ij}]_{K \times K}$ 随机地给出 (见附录), 满足 $P_{ii} = 0, i = 1, 2, \dots, K, Pe = e$ 。性能函数列向量 $f(i) = i, i = 1, 2, \dots, K$ 。从状态 i 到状态 j 的时间服从参数为 j 的 i 阶爱尔朗分布, 即

$$F_{ij}(t) = \int_0^t \frac{j^i}{(i-1)!} x^{i-1} e^{-jx} dx \quad i, j = 1, 2, \dots, K。$$

表 2.1 给出了该半 Markov 过程基于仿真的性能势估值, 根据性能势的物理意义

$$g = \lim_{T \rightarrow \infty} \int_0^T \{P(t)f - \pi P(t)fe\} dt$$

计算出的理论值, 以及按 (2.16) 式计算出的等价 Markov 过程的性能势, 从中可以看出仿真值确实随仿真次数的增加而趋向于理论值, 并且等价的 Markov 过程性能势与原半 Markov 过程的性能势是完全一致的。

性能势 10000 次仿真的估计值	-4.9395	-2.8173	-4.6663	-4.5726	-5.5788
性能势 10 万次仿真的估计值	-3.6756	-2.6713	-0.6737	1.5321	6.0501
性能势 100 万次仿真的估计值	-5.2540	-2.8231	-6.0216	-5.1123	-5.6579
性能势 100 万次仿真的估计值	-3.6907	-2.4660	-0.7244	1.4947	6.1191
性能势 100 万次仿真的估计值	-5.0853	-3.0589	-5.8229	-5.1187	-5.7827
性能势 100 万次仿真的估计值	-3.7444	-2.4467	-0.6338	1.5331	6.1455
性能势的理论值	-5.1292	-3.0864	-5.7900	-5.1211	-5.7973
性能势的理论值	-3.7151	-2.4681	-0.6424	1.4781	6.1765
等价 Markov 过程的性能势	-5.1292	-3.0864	-5.7900	-5.1211	-5.7973
等价 Markov 过程的性能势	-3.7151	-2.4681	-0.6424	1.4781	6.1765

表 2.1 例中半 Markov 过程性能势各种算法的结果比较

2.4 性能势的并行仿真

在 DEDS 研究中, 计算机仿真占有重要的地位。由于 DEDS 系统本身的复杂性和随机不确定性, 单纯用理论分析的方法很难得到满意的结果和答案, 例如无穷小扰动分析方法, 就是兼容了仿真分析和理论分析的长处, 提出只要对系统进行一次仿真, 就能够通过理论分析在标称样本轨道上构造出扰动路径, 并计算出扰动后的性能指标, 从而得到性能指标对参数的灵敏度。

由于 DEDS 本身的随机不确定性, 对其进行的计算机仿真通常非常耗时, 仿真量很大。例如模拟 ATM 网络这样的高速通信网络, 其中某些类型业务的信元丢失率要求在 10^{-9} , 数量级, 换句话说, 就是在传输 10^9 个信元才有可能发生一个丢失事件。要模拟这种稀有事件, 计算机所要运行的时间可想而知是相当长的。为了加快仿真研究速度, 使仿真研究真正具有对实际系统的及时分析用途, 采用多处理器的并向仿真就不失为一种简单而有效的方法。

由于上述原因, 自从出现了大规模并行计算机以来, 国外学者在应用并行机进行 DEDS 仿真方面已经做了大量工作。早期的研究工作致力于对一条 DEDS 复杂的大样本轨道进行分布化和并行化仿真[67], 如 Righte 和 Walrand (1989), Fimoto (1990), 后来学者们又开始研

究怎样在并行计算机上同时仿真多条 DEDS 样本轨道, Vakili (1991, 1992)就是其中最早的学者之一。在这些研究中, 研究对象一般是具有不同参数值或操作规律的大量系统。Vakili 提出的并行仿真方法叫做“标准时钟”(Standard Clock)法。它的基本思想是用一个时钟机构来同时驱动所有系统的仿真, 即用时间同步策略来并行仿真。

另外一种 DEDS 并行仿真方法叫做“事件同步”(Event Synchronization)法, 通过使用 ES 方法, 只有当所有处理器完成它们当前事件的仿真之后各个处理器才能开始仿真下一个事件, 并且在各个处理器中的仿真事件数目是一样的。其基本思想在摄动分析领域中用来同时产生多条样本轨道(Ho 和 Li,1988;Cassandras 和 Strickland, 1989)。

由何毓琦(YC.Ho)教授提出的“有序优化”(Ordinal Optimization)理论已经成为 DEDS 摄动分析领域中的研究热门[71], 这种理论采用相同的随机序列来同时产生多条样本轨道的方法, 正好适合于用 ES 并行仿真方法来进行仿真, 从而带动了在该领域中并行仿真的研究。

在这一节中, 我们将在半 Markov 性能势串行估计算法的基础上给出相应的并行算法, 其中基于实现因子的仿真方法是考虑的重点。因为在性能势的直接估计算法中每一步状态记录涉及到的运算量都很少, 并且如果并行化则各个处理节点之间的通讯开销太多, 得不偿失。如果期望的样本轨道足够长的话可以把样本轨道平均分配给数个并行计算节点, 每个处理节点按照串行算法计算, 得到的结果作算术平均作为最后的估计值。

在并行算法的设计当中, 一般有两种设计方法:一是在相应的串行算法基础上, 通过将总算法划分成许多可以同时进行计算的子任务来进行并行化设计;另一种是抛开串行算法, 以一种全新的并行计算思路构造并程序

对于多数并行算法设计, 第一种设计思想更直接可行。本文即采用这种方法, 通过对本章第三节提出的性能势的串行仿真算法进行并行化来得到最终的并行仿真算法。为了研究并行仿真算法, 必须首先对相应的串行仿真算法有充分的了解和仔细的分析。清楚了解串行算法中各个部分的计算特性, 计算量的比例轻重, 算法中各个部分相互关联的情况等等。为此我们对串行仿真算法进行大量的数值实验以研究算法中各个部分的计算量。对于一个状态空间维数为 21 的系统, 典型的实验结果如下表所示:

样本轨道长	状态转移仿真		参数矩阵运算		性能势、稳态概率计算		总时间
10, 000	0.38	6.3%	5.45	90.0%	0.22	3.7%	6.05
100, 000	3.35	5.7%	54.87	93.9%	0.22	0.4%	58.44
1, 000, 000	33.28	5.8%	541.13	94.2%	0.22	0.04%	574.63

注:表中每部分两栏分别为运算时间和其占总时间的百分比。数据来源:文献[67]

表 2.2 固定问题规模状态空间维数=21

另外我们对固定样本轨道长度为 100,000, 不同状态空间维数的情况进行考查, 典型的实验结果如下表所示:

状态空间维数	状态转移仿真		参数矩阵运算		性能势、稳态概率计算		总时间
4	2.47	29.4%	5.87%	69.9%	0.06	0.7%	8.4
21	3.35	5.7%	54.87	93.9%	0.22	0.4%	58.44
45	4.34	1.7%	253.7	98.0%	0.93	0.3%	258.97

注:表中每部分两栏分别为运算时间和其占总时间的百分比。数据来源:文献[67]

表 2.3 固定样本长度=100.000

从这两个表可以清楚看出,对于中等或者大规模问题的仿真,有 90%以上的运算量集中在参数矩阵运算当中.因此算法中这一步骤并行化程度的高低将直接决定整个并行仿真算法的并行效率.下面我们将集中于这一部分的并行化研究.

观察本章第三节中基于实现矩阵的串行估计算法,第四步的参数矩阵计算是一类特殊的矩阵计算,这种计算只有矩阵和数的乘积和两个同维矩阵对应元素的乘积,即一个矩阵的第 i 行第 1 列元素与另一个矩阵的第 i 行第 1 列元素相乘.另外在一般的矩阵并行计算当中,矩阵元素多数情况下是大于处理器个数的,这样就需要考虑矩阵元素在各个处理器上的划分问题.在并行计算中划分问题需要从两点出发:一是要尽量做到各个处理器达到负载均衡,另一点是要尽量减少并行计算时各个处理器之间的通讯开销.

根据串行算法的特点我们可以按照下面的方式在各个处理器上划分并行子任务,以提供最大的并行效率.首先对参数矩阵运算中涉及到的六个参数矩阵采用相同的划分方式,让这六个矩阵的第 i 行第 1 列元素都被划分到同一个处理器中.对于该算法,通常的矩阵划分方法如行划分、列划分、块划分等都不能完全达到各个处理器的负载均衡.为此我们采用“螺旋式”划分方法[67],其基本思路是:因为矩阵运算中同时有行计算和列计算,划分时就应该兼顾,于是采用对角线的对称行列的方式,再加上螺旋循环来平均划分数目的方法,从而产生了螺旋式划分法.需要说明的是,这里的“螺旋式”划分方法是一种通用的并行计算划分方法,一般来说如果在并行矩阵计算中同时有矩阵的按行计算和按列计算,“螺旋式”划分方法就很有可能达到比行划分和列划分更好的负载均衡.这样的划分方式对于该算法的并行效率非常重要.由于微处理器的飞速发展,并行计算机的处理器运算速度都相当快,但是各个处理器之间交换数据的通讯速度相对来说却要缓慢的多.这样在一般的并行计算中,数据通讯占了并行计算运行时间的相当大的比重,并且常常成为提高并行计算效率的瓶颈.因此如何减少处理器之间的通讯开销就成为设计并行算法时要重点考虑的方面之一.另外对于不同结构的并行计算机一般都要设计不同的并行算法以根据并行机的结构特点,最大程度的减少通讯开销.而这里提出的并行算法却没有矩阵运算的通讯开销,也不用考虑并行机结构,可以适用于各种结构的并行机,因此该并行仿真算法具有很好的并行效率,并且算法的适用面很广,能够应用于各种结构的并行计算机.

当前 MIMD (multiple-instruction multiple-data, 多指令多数据并行计算机)在并行计算领域扮演着越来越重要的角色,由于个人微机(PC)的迅猛发展,由 PC 机群组成的 MIMD 松散型的并行计算环境将在并行计算领域占据主导地位.这种 MIMD 型并行计算机具有成本低廉,构建方便,处理器升级迅速,容易扩充以满足不同程度的运算需要等优点.但是它的主要缺点是通讯带宽较小,通讯速度慢,常常成为并行计算中的瓶颈.在这里我们的并行算法主体中没有通讯开销,使得该算法特别适合这种 MIMD 型并行计算机,具有广泛的应用前景和实用价值.

在对矩阵运算部分使用了“螺旋式”划分法后,实际仿真时还需要采用“公共随机数”(Common Random Number)策略[67],即在所有的处理器上使用相同的随机数发生器的“种子”,这样就会产生相同的随机数序列,即“公共随机数”,在每一个处理器上都使用这个随机序列仿真样本轨道,在每个处理器上产生相同的样本轨道,从而避免了在各个处理器之间传送样本轨道的数据.试验结果分析表明,“公共随机数”策略是该并行仿真算法具有高并行度最重要的因素[56, 67]

性能势基于实现矩阵的并行估计算法如下:

第一步:初始化,设置状态空间,概率转移矩阵等,在所有处理器中使用相同的“种子”初始化随机数发生器,以实现“公共随机数”策略.

第二步:每个处理器都在相同的样本轨道上获取当前状态 n_i , 计算状态 i 的服务时间.

第三步:每个处理器都在相同的样本轨道上确定下一步转移将要到达的状态.

第四步:对本次状态转移进行记录,所有处理器并行计算参数矩阵(使用螺旋式划分法)。

第五步:返回第二步进行下一轮状态转移仿真,直到循环了指定的仿真次数。

第六步:计算稳态概率估值、实现矩阵和性能势估值。

问题规模	处理器数	仿真时间(s)	加速比	并行效率
k=10	1	17.0		
	2	11.8	1.44	72.0%
	5	7.6	2.24	44.8%
	10	6.2	2.74	27.4%
k=45	1	261.1		
	5	54.9	4.76	95.2%
	15	20.8	12.55	83.7%
k=495	1	32546		
	5	6687	4.87	97.3%
	9	3755	8.67	96.3%

表2.4 并行仿真算法实验结果

第三章 半 Markov 控制过程基于性能势的优化算法

本章研究了几种 Markov 控制过程基于性能势的优化算法, 分别给出相应的基于单样本轨道仿真的优化算法、并行优化算法和并行仿真优化算法。并提供了一些数值例子比较几种算法的结果, 对于并行算法着重比较的是其运行时间并得到了满意的结果。[70, 80]

3.1 半 Markov 控制过程基于性能势的优化理论

3.1.1 半 Markov 过程的性能灵敏度分析

考虑一个正常返的, 非周期的, 不可约的半 Markov 过程 $Y = \{Y_t; t \geq 0\}$, 其状态空间为 Φ , 半 Markov 核为 $Q(t) = [Q_{ij}(t)]$ 。由引理 1 可知, Y 存在唯一的稳态概率分布 $p(i)$, $i \in \Phi$ 。用 $p = (p(1), p(2), \dots)$ 表示 Y 的稳态概率向量, 记 $\{X_n; n \geq 0\}$ 为 Y 的嵌入 Markov 链, 则其也存在唯一的稳态概率分布 $\pi(i)$, $i \in \Phi$, 且满足 $\pi e = 1$, $\pi(P - I) = 0$, 这里 $\pi = (\pi(1), \pi(2), \dots)$, $P = [p_{ij}]$ 为嵌入 Markov 链的一步转移概率矩阵。

同 Markov 过程完全一样, 置 $f: \Phi \rightarrow R$ 为性能函数, 并设 $\sum_{i \in \Phi} p(i)|f(i)|$ 收敛, 定义稳态性能 η_f 为 f 关于稳态概率 p 的期望值, 即

$$\eta_f = E_p[f] = \sum_{i \in \Phi} p(i)f(i) = pf \quad (3.1)$$

下面我们将半 Markov 过程的稳态性能灵敏度分析转化为与之等价的 Markov 过程的灵敏度分析。记 S_{ij} 为过程从状态 i 转移到 j 并在状态 i 的平均逗留时间, 即

$$S_{ij} = \int_0^{\infty} t dF_{ij}(t) \quad (3.2)$$

又 S_i 为过程在状态 i 的平均逗留时间, 即

$$S_i = \int_0^{\infty} t dQ_i(t) = \sum_{j \in \Phi} p_{ij} S_{ij} \quad (3.3)$$

这里 F_{ij} , $Q_i(t)$, p_{ij} 的定义见第 2 节。置 $\Lambda = \text{diag}(S_1^{-1}, S_2^{-1}, \dots)$, $A = \Lambda(P - I)$, 则显然, $Ae = 0$ 。由文献[80]可知,

$$p(i) = \frac{\pi(i)S_i}{\sum_{j \in \Phi} \pi(j)S_j} \quad (3.4)$$

或用矩阵表示为

$$p\Lambda = \frac{1}{\sum_{j \in \Phi} \pi(j)S_j} \pi I \quad (3.5)$$

故我们有, $pA = p\Lambda(P - I) = \frac{1}{\sum_{j \in \Phi} \pi(j)S_j} \pi I(P - I) = 0$ 。因此, $A = \Lambda(P - I)$ 可以作为一个

Markov 过程的无穷小矩阵。

现在考虑一个正常返的, 不可约的 Markov 过程 $X = \{X_t; t \geq 0\}$, 具有状态空间 Φ , 无穷小矩阵为 $A = \Lambda(P - I)$, 则由上面的讨论可知, 这个 Markov 过程 X 具有唯一的稳态概率,

且这个稳态概率就是原半 Markov 过程 Y 的稳态概率, 因此对同样的性能函数 f , Markov 过程 X 与原半 Markov 过程 Y 有相同的稳态性能, 从而将半 Markov 过程 Y 的稳态性能灵敏度分析转化为与之等价的 Markov 过程 X 的稳态性能灵敏度分析。

如果我们设半 Markov 过程 Y 的半 Markov 核 $Q(t)$ 为定义在区间 $J \subset R$ 上的某个参数 θ 的可微函数, 性能函数 f 也与参数 θ 有关, 且是 θ 的可微函数。可以得到稳态性能 η_f 关于参数 θ 的导数公式为

$$\frac{\partial \eta_f}{\partial \theta} = p \frac{\partial \Lambda}{\partial \theta} (P - I)x^{(f)} + p\Lambda \frac{\partial (P - I)}{\partial \theta} x^{(f)} + p \frac{\partial f}{\partial \theta} \quad (3.6)$$

这里 $x^{(f)}$ 为 Markov 过程 X 关于性能函数 f 的势能, 即 $x^{(f)}$ 满足 Poisson 方程

$$\Lambda(P - I)x = -f + \eta_f e \quad (3.7)$$

3.1.2 半 Markov 决策过程在平均代价准则下的最优性方程

我们将文献[76]中关于连续时间 Markov 控制过程 (CTMCP) 的有关结果推广到无限状态空间的情况。考虑一个连续时间 Markov 过程 $X = \{X_t; t \geq 0\}$, 具有状态空间 Φ 和紧致行动集 $D = D(1) \times D(2) \times \dots$, $D(i)$ 是状态 i 的容许行动集。一个平稳策略是状态空间 Φ 到行动集的一个映射, 记为 $v = (v(1), v(2), \dots)$, 这里 $v(i) = d_i \in D(i)$, $i \in \Phi$ 。令 Ω_s 是全体平稳策略集。假设在任意策略 $v \in \Omega_s$ 的驱动下, X 是不可约的, 正常返的, 其转移概率矩阵为 $P^v(t) = [p_{ij}(t, v(i))]$, 无穷小矩阵为 $A^v = [a_{ij}(v(i))]$ 。设 $\{X_n; n = 0, 1, 2, \dots\}$ 是它的嵌入

Markov 链, 其一步转移概率矩阵 $P^v = [p_{ij}(v(i))]$, 则

$$A^v = \text{diag}(\lambda(1, v(1)), \lambda(2, v(2)), \dots)(P^v - I) \quad (3.8)$$

这里 $\lambda(i, v(i)) > 0$, 为过程 X 在策略 v 下状态 i 的转移率。记在策略 v 下 X 的稳态概率为 $p^v = (p(1, v(1)), p(2, v(2)), \dots)$, 则有

$$p^v e = 1, \quad A^v e = 0, \quad p^v A^v = 0 \quad (3.9)$$

令 f 为依赖于策略 v 的性能函数, 且记 $f^v = (f(1, v(1)), f(2, v(2)), \dots)^T$ 。我们有下列假设:

假设 1. 对任意 $i, j \in \Phi$, $p_{ij}(t, v(i))$ 是定义在 $D(i)$ 上的连续函数。

假设 2. 对任意 $i \in \Phi$, $f(i, v(i))$ 是定义在 $D(i)$ 上的连续函数, 且 f^v 一致有界。

假设 3. 存在常数 λ , 满足 $\sup_{i \in \Phi, v \in \Omega_s} \{\lambda(i, v(i))\} = \lambda < +\infty$ 。

我们称 (X, Φ, D, A^v, f^v) 为约束在平稳策略集 Ω_s 上的一个 CTMCP。 X 关于无限时间水平平均代价准则为

$$\eta^v = \lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_0^T f(X_t, v(X_t)) dt \right\}, \quad v \in \Omega_s \quad (3.10)$$

因为 X 是遍历的, 故有 $\eta^v = \sum_{i \in \Phi} p(i, v(i)) f(i, v(i)) = p^v f^v$ 。在 CTMCP 问题中, 优化的目的是选择一控制策略, 使方程 (3.10) 式的右边达到最小。

对任意 $v \in \Omega_s, \alpha \geq 0$, 我们定义 CTMCP 的折扣 Poisson 方程为

$$(\alpha I - A^v + \lambda e \pi^v) g = f^v \quad (3.11)$$

令 $\tilde{P}^v = A^v / \lambda + I$, $\beta = \lambda / (\lambda + \alpha)$, 则 $0 < \beta \leq 1$, 且 \tilde{P}^v 是一随机矩阵。由文献 [] 可知, 矩阵 $(I - \beta \tilde{P}^v + \beta e \pi^v)$ 是非奇异的, 故矩阵 $(\alpha I - A^v + \lambda e \pi^v)$ 也是非奇异的, 存在唯一解 $g_\alpha^v = (\alpha I - A^v + \lambda e \pi^v)^{-1} f^v$ 。令 $\alpha = 0$, 则有

$$g^v = (-A^v + \lambda e \pi^v)^{-1} f^v \quad (3.12)$$

我们称 (3.12) 式为 CTMCP 的平均代价 Poisson 方程, $g^v = (g^v(1), g^v(2), \dots)^T$ 为性能势向量。

可得到下列引理和定理。

引理 5 在假设 3 下, 对任意 $v', v \in \Omega_s$, 有

$$\eta^{v'} - \eta^v = \pi^v [(f^{v'} + A^{v'} g^{v'}) - (f^v + A^v g^v)] \quad (3.13)$$

定理 1 在假设 1-3 下, $v^* \in \Omega_s$ 是 Markov 控制过程 (X, Φ, D, A^v, f^v) 平均代价最优平稳策略的充分必要条件为

$$f^{v'} + A^{v'} g^{v'} \leq f^v + A^v g^v, \quad \forall v \in \Omega_s \quad (3.14)$$

可得 $(-A^{v'} + \lambda e \pi^{v'}) g^{v'} = f^{v'}$, 即 $\lambda e \pi^{v'} g^{v'} = f^{v'} + A^{v'} g^{v'}$ 。 $\lambda \pi^{v'} g^{v'} = \pi^{v'} f^{v'}$, 因此

$e\eta^{v^*} = f^{v^*} + A^{v^*}g^{v^*}$ 。故有：

定理 2 在假设 1-3 下, $v^* \in \Omega_s$ 是 Markov 控制过程 (X, Φ, D, A^v, f^v) 平均代价最优平稳策略的充分必要条件是满足方程

$$e\eta^{v^*} = \min_{v \in \Omega_s} \{f^v + A^v g^{v^*}\} \text{ 或 } 0 = \min_{v \in \Omega_s} \{f^v + A^v g^{v^*} - e\eta^{v^*}\} \quad (3.15)$$

(3.15) 式称为 CTMCP 在平均代价准则下基于性能势的最优性方程。

现在我们来导出半 Markov 决策过程 (SMDP) 在平均代价准则下, 基于等价 Markov 过程性能势的最优性方程。考虑半 Markov 过程 $Y = \{Y_t; t \geq 0\}$, 具有状态空间 Φ 和紧致行动集 $D = D(1) \times D(2) \times \dots$, $D(i)$ 是状态 i 的容许行动集。一个平稳策略是状态空间 Φ 到行动集的一个映射, 记为 $v = (v(1), v(2), \dots)$, 这里 $v(i) = d_i \in D(i)$, $i \in \Phi$ 。令 Ω_s 是全体平稳策略集。假设在任意策略 $v \in \Omega_s$ 的驱动下, Y 是不可约的, 非周期的, 正常返的。 Y 的半 Markov 核为 $Q^v(t) = [Q_{ij}(t, v(i))]$, 嵌入 Markov 链的一步转移概率矩阵为 $P^v = [p_{ij}(v(i))]$ 。

在策略 v 下, Y 的稳态概率为 $p^v = (p(1, v(1)), p(2, v(2)), \dots)$, f 为依赖于策略 v 的性能函数, 且记 $f^v = (f(1, v(1)), f(2, v(2)), \dots)^T$ 。同样有下列假设:

假设 4. 对任意 $i, j \in \Phi$, $Q_{ij}(t, v(i))$ 是定义在 $D(i)$ 上的连续函数。

假设 5. 对任意 $i \in \Phi$, $f(i, v(i))$ 是定义在 $D(i)$ 上的连续函数, 且 f^v 一致有界。

假设 6. 存在常数 $\mu > 0$, 满足 $\sup_{i \in \Phi, v \in \Omega_s} \{S_i^{-1}\} = \mu < +\infty$ 。

我们称 $(Y, \Phi, D, Q^v(t), f^v)$ 为约束在平稳策略集 Ω_s 上的一个 SMDP。 Y 关于无限时间水平平均代价准则为

$$\eta^v = \lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_0^T f(Y_t, v(Y_t)) dt \right\}, \quad v \in \Omega_s \quad (3.16)$$

因为 Y 是遍历的, 故有 $\eta^v = \sum_{i \in \Phi} p(i, v(i)) f(i, v(i)) = p^v f^v$ 。

考虑一个正常返的, 不可约的 Markov 过程 $X = \{X_t; t \geq 0\}$, 具有状态空间 Φ , 无穷小矩阵为 $A^v = \Lambda^v(P^v - I)$, 则由前面的讨论可知, Markov 决策过程 (X, Φ, D, A^v, f^v) 与原半 Markov 决策过程 $(Y, \Phi, D, Q^v(t), f^v)$, 在无限水平平均代价准则下是等价的。故我们有:

定理 3 在假设 4-6 下, $v^* \in \Omega_s$ 是半 Markov 决策过程 $(Y, \Phi, D, Q^v(t), f^v)$ 平均代价最优平稳策略的充分必要条件是满足方程

$$e\eta^{v^*} = \min_{v \in \Omega_s} \{f^v + \Lambda^v(P^v - I)g^{v^*}\} \text{ 或 } 0 = \min_{v \in \Omega_s} \{f^v + \Lambda^v(P^v - I)g^{v^*} - e\eta^{v^*}\} \quad (3.17)$$

其中, $g^v = [-\Lambda^v(P^v - I) + \mu e \pi^v]^{-1} f^v$ 为 Markov 过程 X 的性能势。(3.17) 式就是 SMDP 在平均代价准则下, 基于等价 Markov 过程性能势的最优性方程。

3.2 半 Markov 控制过程基于性能势的优化算法

在系统模型已知时, 数值迭代算法无需矩阵求逆运算, 可以根据模型参数直接进行优化计算, 因此不必考虑数值迭代仿真算法, 当然对于状态空间很大的 SMCP 我们也可以通过仿真随机产生出具有代表性的状态, 然后对这些状态进行数值迭代, 这样避免了每次都对那些很少出现的状态进行计算, 因而节省了计算时间。对于梯度算法和策略迭代算法, 模型已知时的主要问题是性能势的仿真, 这一部分我们在第二章中已经详细讨论过。

在系统模型未知或者不完全知道时, 也必须求助于仿真的方法, 数值迭代算法在迭代过程中需要同时对系统的模型参数进行估计和改进, 而在梯度算法和策略迭代算法中, 不仅要同时对系统的模型参数进行估计, 还要对性能势进行估计。例如在受控排队网络中, 模型参数取决于路径概率, 而路径概率很容易由一条样本轨道估计得到。因为模型已知, 这样的优化问题可以用基于计算的方法来求解。

3.2.1 策略迭代算法及其收敛性证明

由上一节, 给出如下的策略迭代算法:

步 1: 给定 α , 令 $k = 0, \varepsilon > 0$, 选择初始策略 v_k ;

步 2: 计算 $A_\alpha^{v_k}$, $p_\alpha^{v_k}$ 和 $g_\alpha^{v_k}$;

步 3: 选择 v_{k+1} , 对每一个状态 $i \in \Phi$, 满足

$$v_{k+1}(i) \in \arg \min_{v(i) \in D(i)} \left\{ f(i, v(i)) + \sum_{j \in \Phi} A_\alpha^{v(i)}(i, j) g_\alpha^{v_k}(j) \right\}; \quad (3.18)$$

步 4: 如果 $M(f^{v_{k+1}} + A_\alpha^{v_{k+1}} g_\alpha^{v_k} - (f^{v_k} + A_\alpha^{v_k} g_\alpha^{v_k})) < \alpha \varepsilon$, 则算法终止; 否则, 置 $k := k + 1$,

转步 2.

此处定义 $M(h) = \max_i \{h(i)\}$. 在假设 1 和 2 成立的条件下, 可知至少存在一个满足方程

(3.18) 的策略 v_{k+1} 使得:

$$f^{v_{k+1}} + A_\alpha^{v_{k+1}} g_\alpha^{v_k} \leq f^{v_k} + A_\alpha^{v_k} g_\alpha^{v_k} \quad \forall v \in \Omega_s. \quad (3.19)$$

那么一定有

$$f^{v_{k+1}} + A_\alpha^{v_{k+1}} g_\alpha^{v_k} \leq f^{v_k} + A_\alpha^{v_k} g_\alpha^{v_k} \quad (3.20)$$

根据文献[76]中的引理这里 $(\alpha I - A_\alpha^{v_{k+1}})^{-1}$ 中的每个分量都大于零, 可以得到

$$\eta_{\alpha}^{v_k} - \eta_{\alpha}^{v_{k+1}} = (\alpha I - A_{\alpha}^{v_{k+1}})^{-1} [(f^{v_k} + A_{\alpha}^{v_k} g_{\alpha}^{v_k}) - (f^{v_{k+1}} + A_{\alpha}^{v_{k+1}} g_{\alpha}^{v_k})] \geq 0,$$

即

$$\eta_{\alpha}^{v_{k+1}} \leq \eta_{\alpha}^{v_k}.$$

这就表明 v_{k+1} 是一个改进策略.

下面证明策略迭代算法的收敛性:

定理 1 如果前述假设 1 和 2 成立, 那么有:

- (1) $\lim_{k \rightarrow \infty} M(f^{v_{k+1}} + A_{\alpha}^{v_{k+1}} g_{\alpha}^{v_k} - f^{v_k} - A_{\alpha}^{v_k} g_{\alpha}^{v_k}) = 0$;
- (2) 存在一个最优策略 v^* 满足 $\lim_{k \rightarrow \infty} v_k = v^*$ 且 $\lim_{k \rightarrow \infty} \eta_{\alpha}^{v_k} = \eta_{\alpha}^{v^*}$.

证明:

由上分析知 $\{v_k\}$ 为改进策略序列, 那么 $\{\eta_{\alpha}^{v_k}\}$ 是一个递减且有下界的序列, 且最优折扣代价 $\eta_{\alpha}^{v^*}$ 是它的一个下界. 因此存在一个常数 c ($c \geq \eta_{\alpha}^{v^*}$), 使得 $\lim_{k \rightarrow \infty} \eta_{\alpha}^{v_k} = c$. 因为 η_{α}^v 是紧致集 D 上的一个连续函数, 所以一定存在一个策略 $\delta = (\delta_1, \delta_2, \dots, \delta_k) \in D$ 满足 $\eta_{\alpha}^{\delta} = c$ 和 $\lim_{k \rightarrow \infty} v_k = \delta$. 那么根据文献[77]可得

$$\begin{aligned} \lim_{k \rightarrow \infty} (f^{v_{k+1}} + A_{\alpha}^{v_{k+1}} g_{\alpha}^{v_k}) &= f^{\delta} + A_{\alpha}^{\delta} g_{\alpha}^{\delta} = \alpha \eta_{\alpha}^{\delta}. \\ \alpha \eta_{\alpha}^{v^*} &= A_{\alpha}^{v^*} g_{\alpha}^{v^*} + f^{v^*}, \quad v \in \Omega, \end{aligned} \quad (3.21)$$

因而,

$$\lim_{k \rightarrow \infty} (f^{v_k} + A_{\alpha}^{v_k} g_{\alpha}^{v_k}) = \lim_{k \rightarrow \infty} \alpha \eta_{\alpha}^{v_k} = \alpha \eta_{\alpha}^{\delta},$$

所以一定存在一个整数 K , 使得对 $\forall \varepsilon > 0$, 当 $k > K$ 时有 $M(f^{v_{k+1}} + A_{\alpha}^{v_{k+1}} g_{\alpha}^{v_k} - (f^{v_k} + A_{\alpha}^{v_k} g_{\alpha}^{v_k})) < \alpha \varepsilon$ 成立. 根据文献[12]中的引理 6.3.5, 当 $k > K$ 时

$$\begin{aligned} \eta_{\alpha}^{v_k} - \eta_{\alpha}^{v^*} &= (\alpha I - A_{\alpha}^{v^*})^{-1} [(f^{v_k} + A_{\alpha}^{v_k} g_{\alpha}^{v_k}) - (f^{v^*} + A_{\alpha}^{v^*} g_{\alpha}^{v_k})] \\ &\leq (\alpha I - A_{\alpha}^{v^*})^{-1} [(f^{v_k} + A_{\alpha}^{v_k} g_{\alpha}^{v_k}) - (f^{v_{k+1}} + A_{\alpha}^{v_{k+1}} g_{\alpha}^{v_k})] \leq (\alpha I - A_{\alpha}^{v^*})^{-1} \alpha \varepsilon e = \varepsilon. \end{aligned} \quad (3.22)$$

因此 v_k 是 ε -最优平稳策略. 由 ε 的任意性, 有 $\lim_{k \rightarrow \infty} \eta_{\alpha}^{v_k} = \eta_{\alpha}^{v^*}$. 事实上, 通过极限的唯一性准则, 我们可以得到 $\eta_{\alpha}^{v^*} = \eta_{\alpha}^{\delta}$, 这里的 δ 表示最优策略的行为. 因此, 有 $\lim_{k \rightarrow \infty} v_k = v^*$ 和

$$\lim_{k \rightarrow \infty} \eta_{\alpha}^{v_k} = \eta_{\alpha}^{v^*}.$$

该定理表明前面所述的策略迭代算法将在有限步骤之内终止, 并且 ε -最优平稳策略是可得.

3.2.2 数值迭代算法及其收敛性证明

下面给出一种数值迭代算法，并证明该算法的收敛性。

首先根据引理 2 可知

$$\alpha \eta_\alpha^v = \min_{v(i) \in \Omega_\alpha} \{f^v + A_\alpha^v \eta_\alpha^v\} = \min_{v(i) \in \Omega_\alpha} \{f^v + \lambda_\alpha (\tilde{P}_\alpha^v - I) \eta_\alpha^v\},$$

整理得

$$\eta_\alpha^v = \min_{v(i) \in D(i)} \left\{ \tilde{f}^v + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^v \eta_\alpha^v \right\}.$$

这里

$$\tilde{f}^v = \frac{1}{\lambda_\alpha + \alpha} f^v, \quad \tilde{P}_\alpha^v = \frac{1}{\lambda_\alpha} A_\alpha^v + I. \quad (3.23)$$

这样，我们给出如下数值迭代算法：

步 1: 给定 α ，令 $k=0, \varepsilon > 0$ ；选择一个初始策略 v_k ，计算 $g_\alpha^{v_k}$ ，且令 $h_\alpha^k = g_\alpha^{v_k}$ ；

步 2: 选择 v_{k+1} ，对每一个状态 $i \in \Phi$ ，满足

$$v_{k+1}(i) \in \arg \min_{v(i) \in D(i)} \left\{ \tilde{f}(i, v(i)) + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \sum_{j \in \Phi} \tilde{p}_{ij}^\alpha(v(i)) h_\alpha^k(j) \right\}, \quad (3.24)$$

这里的 $\tilde{p}_{ij}^\alpha(v(i))$ 是 \tilde{P}_α^v 的第 (i, j) 个元素。

步 3: $h_\alpha^{k+1} = \tilde{f}^{v_{k+1}} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_{k+1}} h_\alpha^k$, (3.25)；

步 4: 若 $sp(h_\alpha^{k+1} - h_\alpha^k) < \frac{\alpha}{\alpha + \lambda_\alpha} \varepsilon$ ，则算法终止；否则，置 $k := k+1$ ，转步 2。

定理 2 在假设 1 和 2 成立的条件下，上述的数值迭代算法是收敛的且得到 ε 最优策略。

证明：由假设 1 和 2 知，对每一状态 $i \in \Phi$ ， $\tilde{f}(i, v(i)) + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \sum_{j \in \Phi} \tilde{p}_{ij}^\alpha(v(i)) h_\alpha^k(j)$ 在紧

致集 $D(i)$ 上连续，因此存在 v_{k+1} 满足，即对任意 k ，有

$$h_\alpha^{k+1} = \tilde{f}^{v_{k+1}} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_{k+1}} h_\alpha^k \leq \tilde{f}^{v^*} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v^*} h_\alpha^k.$$

根据此递推关系，有

$$h_\alpha^k \leq \sum_{n=0}^{k-1} P_n \tilde{f}^{v^*} + P_k h_\alpha^0, \quad (3.26)$$

这里, 定义 $P_n = \left(\frac{\lambda_\alpha}{\lambda_\alpha + \alpha}\right)^n \tilde{P}_n$, $n \geq 1$, 其中 $\tilde{P}_n = [\tilde{P}_\alpha^{v^*}]^n$, 且 $\tilde{P}_0 = [\tilde{P}_\alpha^{v^*}]^0 = I$. 可见 \tilde{P}_n 也是

一随机矩阵, 有 $\tilde{P}_n e = e$. 得

$$\alpha \eta_\alpha^{v^*} = f^{v^*} + A_\alpha^{v^*} g_\alpha^{v^*} = f^{v^*} + \lambda_\alpha (\tilde{P}_\alpha^{v^*} - I) g_\alpha^{v^*},$$

两边同时除以 $\lambda_\alpha + \alpha$, 可得

$$\tilde{f}^{v^*} = -\frac{\lambda_\alpha}{\lambda_\alpha + \alpha} (\tilde{P}_\alpha^{v^*} - I) g_\alpha^{v^*} + \frac{\alpha \eta_\alpha^{v^*}}{\lambda_\alpha + \alpha}.$$

将上式代入 (3.26) 式, 有

$$\begin{aligned} h_\alpha^k &\leq \sum_{n=0}^{k-1} P_n \left(-\frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v^*} g_\alpha^{v^*} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} g_\alpha^{v^*} + \frac{\alpha \eta_\alpha^{v^*}}{\lambda_\alpha + \alpha} \right) + P_k h_\alpha^0 \\ &\leq \frac{p_\alpha^{v^*} f^{v^*} (1 - (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k) e}{\alpha (\lambda_\alpha + \alpha)} + g_\alpha^{v^*} + \left(\frac{\lambda_\alpha}{\lambda_\alpha + \alpha}\right)^k \max_{i \in \Phi} \{(h_\alpha^0 - g_\alpha^{v^*})(i)\} e. \end{aligned} \quad (3.27)$$

对应算法的迭代过程, 定义一个有限 k 阶段策略 $\Pi_k = (v_k, v_{k-1}, \dots, v_1)$, $P_{\Pi_k}^0 = I$, 则在策略 Π_k 下, n 步转移概率矩阵 $P_{\Pi_k}^n = \tilde{P}_\alpha^{v_k} \tilde{P}_\alpha^{v_{k-1}} \dots \tilde{P}_\alpha^{v_{k-n+1}}$, $k \geq n \geq 1$, 可见 $P_{\Pi_k}^n$ 也是一随机矩阵. 迭代得

$$\begin{aligned} h_\alpha^k &= \tilde{f}^{v_k} + \tilde{P}_\alpha^{v_k} h_\alpha^{k-1} = \tilde{f}^{v_k} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_k} (\tilde{f}^{v_{k-1}} + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_{k-1}} h_\alpha^{k-2}) \\ &= \sum_{n=1}^k \left(\frac{\lambda_\alpha}{\lambda_\alpha + \alpha}\right)^{k-n} P_{\Pi_k}^{k-n} \tilde{f}^{v_n} + \left(\frac{\lambda_\alpha}{\lambda_\alpha + \alpha}\right)^k P_{\Pi_k}^k h_\alpha^0 \end{aligned} \quad (3.28)$$

根据文献[72]中的定理有

$$0 = \min_{v \in \Omega_s} \{f^v + A_\alpha^v g_\alpha^v - \alpha \eta_\alpha^v\} = \min_{v \in \Omega_s} \left\{ \tilde{f}^v + \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^v g_\alpha^v - \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} g_\alpha^v - \frac{\alpha \eta_\alpha^v}{\lambda_\alpha + \alpha} \right\}.$$

因此, 对任意的 n , 有 $\tilde{f}^{v_n} \geq \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} g_\alpha^{v_n} + \frac{\alpha \eta_\alpha^{v_n}}{\lambda_\alpha + \alpha} - \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_n} g_\alpha^{v_n}$, 并代入 (3.28) 式得

$$h_\alpha^k \geq \sum_{n=1}^k P_{\Pi_k}^{k-n} \left(\frac{\lambda_\alpha}{\lambda_\alpha + \alpha} g_\alpha^{v_n} + \frac{\alpha \eta_\alpha^{v_n}}{\lambda_\alpha + \alpha} - \frac{\lambda_\alpha}{\lambda_\alpha + \alpha} \tilde{P}_\alpha^{v_n} g_\alpha^{v_n} \right) + P_{\Pi_k}^k h_\alpha^0$$

$$\geq \frac{p_\alpha^v f^v (1 - (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k) e}{\alpha(\lambda_\alpha + \alpha)} + g_\alpha^v + (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k \min_{i \in \Phi} \{(h_\alpha^0 - g_\alpha^v)(i)\} e. \quad (3.29)$$

由 (3.27) 和 (3.29) 式得

$$\begin{aligned} & \frac{p_\alpha^v f^v (1 - (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k) e}{\alpha(\lambda_\alpha + \alpha)} + (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k \min_{i \in \Phi} \{(h_\alpha^0 - g_\alpha^v)(i)\} e \leq h_\alpha^k - g_\alpha^v \\ & \leq \frac{p_\alpha^v f^v (1 - (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k) e}{\alpha(\lambda_\alpha + \alpha)} + (\frac{\lambda_\alpha}{\lambda_\alpha + \alpha})^k \max_{i \in \Phi} \{(h_\alpha^0 - g_\alpha^v)(i)\} e. \end{aligned}$$

令 $k \rightarrow \infty$ 得

$$\lim_{k \rightarrow \infty} h_\alpha^k = g_\alpha^v + \frac{p_\alpha^v f^v e}{\alpha(\lambda_\alpha + \alpha)} = \eta_\alpha^v,$$

因而

$$\lim_{k \rightarrow \infty} (h_\alpha^{k+1} - h_\alpha^k) = 0,$$

可见

$$\lim_{k \rightarrow \infty} sp(h_\alpha^{k+1} - h_\alpha^k) = 0,$$

故对任意给定的常数 ε , 存在 N , 使得当 $k > N$ 时, 恒有 $sp(h_\alpha^{k+1} - h_\alpha^k) < \frac{\alpha}{\alpha + \lambda_\alpha} \varepsilon$ 成立,

亦即算法能够在有限步内停止. 下面证明当 $k \rightarrow \infty$ 时, 有 $\eta_\alpha^{k+1} \rightarrow \eta_\alpha^v$.

$$\begin{aligned} \eta_\alpha^{k+1} &= (\alpha I - A_\alpha^{v_{k+1}})^{-1} f^{v_{k+1}} = (\alpha I - A_\alpha^{v_{k+1}})^{-1} [f^{v_{k+1}} - (\alpha I - A_\alpha^{v_{k+1}}) h_\alpha^k] + h_\alpha^k \\ &= (\alpha I - A_\alpha^{v_{k+1}})^{-1} [f^{v_{k+1}} + \lambda_\alpha \tilde{P}_\alpha^{v_{k+1}} h_\alpha^k - (\alpha + \lambda_\alpha) h_\alpha^k] + h_\alpha^k \\ &= (\alpha + \lambda_\alpha) (\alpha I - A_\alpha^{v_{k+1}})^{-1} (h_\alpha^{k+1} - h_\alpha^k) + h_\alpha^k \leq \frac{\alpha + \lambda_\alpha}{\alpha} \max_i (h_\alpha^{k+1} - h_\alpha^k) + h_\alpha^k \end{aligned}$$

而

$$\begin{aligned} \eta_\alpha^v &= (\alpha I - A_\alpha^v)^{-1} f^v = (\alpha I - A_\alpha^v)^{-1} [f^v - (\alpha I - A_\alpha^v) h_\alpha^k] + h_\alpha^k \\ &= (\alpha I - A_\alpha^v)^{-1} [f^v + \lambda_\alpha \tilde{P}_\alpha^v h_\alpha^k - (\alpha + \lambda_\alpha) h_\alpha^k] + h_\alpha^k \\ &\geq (\alpha I - A_\alpha^v)^{-1} [f^{v_{k+1}} + \lambda_\alpha \tilde{P}_\alpha^{v_{k+1}} h_\alpha^k - (\alpha + \lambda_\alpha) h_\alpha^k] + h_\alpha^k \\ &= (\alpha + \lambda_\alpha) (\alpha I - A_\alpha^{v_{k+1}})^{-1} (h_\alpha^{k+1} - h_\alpha^k) + h_\alpha^k \end{aligned}$$

$$\geq \frac{\alpha + \lambda_\alpha}{\alpha} \min_i (h_\alpha^{k+1} - h_\alpha^k) + h_\alpha^k.$$

所以

$$\eta_\alpha^{v_{k+1}} - \eta_\alpha^{v_k} \leq \frac{\alpha + \lambda_\alpha}{\alpha} sp(h_\alpha^{k+1} - h_\alpha^k) \leq \frac{\alpha + \lambda_\alpha}{\alpha} \varepsilon$$

3.2.3 数值例子

考虑一个具有四个状态的半 Markov 过程 $Y = \{Y_t; t > 0\}$ ，其状态空间为 $\Phi = \{1, 2, 3, 4\}$ ，对每个状态 $i \in \Phi$ 的容许行动集为紧致集 $D(i) = [0.5, 25]$ ，一个平稳策略为 $v = (v(1), v(2), v(3), v(4))$ ， $v(i) \in D(i)$ ， $i = 1, 2, 3, 4$ 。假设其嵌入 Markov 链的一步转移概率为：

$$\begin{aligned} p(1,1,v(1)) &= 1 - e^{-\frac{v(1)}{2}} & , & & p(1,2,v(1)) &= \frac{5}{8} e^{-\frac{v(1)}{2}} & , & & p(1,3,v(1)) &= \frac{1}{8} e^{-\frac{v(1)}{2}} & , \\ p(1,4,v(1)) &= \frac{1}{4} e^{-\frac{v(1)}{2}} & ; & & p(2,1,v(2)) &= \frac{1}{3} e^{-\frac{v(2)}{3}} & , & & p(2,2,v(2)) &= \frac{1}{4} e^{-\frac{v(2)}{3}} & , \\ p(2,3,v(2)) &= \frac{1}{5} e^{-\frac{v(2)}{3}} & , & & p(2,4,v(2)) &= 1 - \frac{47}{60} e^{-\frac{v(2)}{3}} & & & p(3,1,v(3)) &= \frac{1}{4} e^{-\frac{v(3)}{4}} & , \\ p(3,2,v(3)) &= \frac{1}{4} e^{-\frac{v(3)}{4}} & , & & p(3,3,v(3)) &= \frac{1}{3} e^{-\frac{v(3)}{4}} & , & & p(3,4,v(3)) &= 1 - \frac{5}{6} e^{-\frac{v(3)}{4}} & ; \\ p(4,1,v(4)) &= \frac{1 - e^{-\frac{v(4)}{2}}}{1 + e^{-\frac{v(4)}{2}}} & , & & p(4,2,v(4)) &= \frac{e^{-\frac{v(4)}{2}}}{1 + e^{-\frac{v(4)}{2}}} & , & & p(4,3,v(4)) &= \frac{\frac{3}{5} e^{-\frac{v(4)}{2}}}{1 + e^{-\frac{v(4)}{2}}} & , \\ p(4,4,v(4)) &= \frac{\frac{2}{5} e^{-\frac{v(4)}{2}}}{1 + e^{-\frac{v(4)}{2}}} & . \end{aligned}$$

已知过程 Y 处于状态 i 且下一次转移到状态 j 时，在状态 i 的逗留时间服从区间

$$[0, jv(i)] \text{ 上的均匀分布, 即分布函数为 } F(i, j, v(i), t) = \begin{cases} \frac{t}{jv(i)}, & 0 \leq t \leq jv(i) \\ 1, & t > jv(i) \end{cases} \text{ , 则半 Markov}$$

核为 $Q(i, j, v(i), t) = P(i, j, v(i))F(i, j, v(i), t)$ ，性能函数为

$$f(i, v(i)) = \ln[(1+i)v(i)] + \frac{\sqrt{i}}{2v(i)}, \quad i=1, 2, 3, 4.$$

首先考虑用策略迭代算法求解, 取 $\alpha = 0.5$, $\varepsilon = 0.00001$, 并分别取初始策略 $v_0 = (1.5, 2, 1.2, 1), (10, 11, 12, 11), (1, 12, 27, 5)$, 则可以得到三种情况下的 ε -最优策略均为 $v_\varepsilon = (1.0202, 0.7323, 0.6372, 0.7736)$, ε -最优代价均为 $\eta^{v_\varepsilon} = (3.0717, 3.6080, 3.9801, 3.9801, 4.1861)$.

表 3.1 选择不同初始策略迭代的结果

v_0	v_ε	η^{v_ε}
1.5, 2, 1.2, 1	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861
10, 11, 12, 11	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861
1, 12, 27, 5	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861

然后考虑用数值迭代算法求解最优策略, 仍取初始策略分别为 $(1.5, 2, 1.2, 1), (10, 11, 12, 11), (1, 12, 27, 5)$, 得到下面与表 1 相同的结果:

表 3.2 选择不同初始策略迭代的结果

v_0	v_ε	η^{v_ε}
1.5, 2, 1.2, 1	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861
10, 11, 12, 11	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861
1, 12, 27, 5	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861

下面取定初始策略 $v_0 = (1,1,1,1)$, 并取不同的 ε 值, 可以得到下表:

表 3.3 不同 ε 值对应的数值迭代结果

ε	v_ε	η^{v_ε}
0.1	1.0087, 0.7533, 0.6441, 0.7437	3.0721, 3.6086, 3.9806, 4.1870
0.001	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861
0.00001	1.0202, 0.7323, 0.6372, 0.7736	3.0717, 3.6080, 3.9801, 4.1861

从这个例子可以看出, 本文给出的策略迭代算法和数值迭代算法都是正确可行的, 并分别验证了两个算法的收敛性.

3.3 半 Markov 控制过程基于性能势的并行优化算法

在前面给出的基于理论计算的算法基础上我们可以发展基于单样本轨道的仿真优化算法, 并进一步发展其相应的并行化算法.

对于大规模实际 CTMCP 系统, 由于其状态空间存在“维数灾”问题, 在对此类系统进行性能优化时, 性能势的计算开销非常大, 串行算法难以满足速度要求, 因此需要考虑发展并行算法. 性能势估计的并行算法已于第二章讨论过, 这里可以直接应用. 而优化过程中另一个重要步骤: 迭代子优化问题也能并行处理, 所以可以在原串行算法的基础上给出相应的并行算法. 在下一节正式介绍并行仿真算法之前, 下面先以并行数值迭代算法为例子来说明并行思想.

针对目前比较常用的并行机和编程模型, 我们采用的并行算法是基于分布存储系统和消息传递模型来提高其通用性, 具体地, 我们利用 MPI (Message Passing Interface) 并行编程标准[64]在曙光 2000 大型并行处理计算机[65. 66]上实现. 曙光 2000 是具有可扩展机群体系统结构的通用超级并行机系统, 通过显式的消息传递来交换数据, 以共同完成同一计算任务. MPI 是一个并行编程标准, 可以几乎不加修改地移植到所有的并行计算机上, 因此目前应用非常广泛. 假设在并行算法中, 我们使用 C 个处理节点, 且每个处理节点对 $[M/C]$ ($[\cdot]$ 代表取整运算) 个状态寻优(最后一个处理节点所处理的状态数略少). 下面是并行策略迭代算法的具体过程.

第一步: 令 $k=0$, $\varepsilon > 0$, 选择初始策略 v_k .

第二步: 计算 p^k 和 g^k , 进行广播(Broadcast)操作, 将 g^k 广播到各个节点.

这里的广播操作是 MPI 中的一个组通信操作, 它完成从源(root)节点(进程)将一条消息广播发送到组内的所有其他的节点(进程), 同时也包括它本身在内. 其执行结果是将源节点(进程)通信消息缓冲区中的消息拷贝到其他所有节点(进程)中去.

第三步: 在 N 个处理节点中并行寻优. 对每一状态 $i \in \Phi$ 寻找

$$v_{k+1}(i) \in \arg \min_{v(i) \in D(i)} \left\{ f(i, v(i)) + \sum a_{ij}(v(i)) g^k(j) \right\}$$

第五步:进行收集(Gather)操作,把各个处理节点中相应的 v_{k+1} 分量收集在一起,再进行广播(Broadcast)操作,把整个 v_{k+1} 向量广播到各个处理节点。

其中的收集操作也是 MPI 中的一个组通信操作,在收集调用中,每个节点(进程包括源节点本身)将其发送缓冲区中的消息发送到源节点(进程),源节点(进程)根据发送节点(进程)的序号将它们各自的消息依次存放到自己的消息缓冲区中。

第五步:

如果 $sp(f^{v_{k+1}} + A^{v_{k+1}}g^{v_k}) < \varepsilon$, 则记 $v_{\min} = v_k$; 算法停止。否则令 $k=k+1$, 转步骤 2。

例 3.3 考虑一半 Markov 过程, 状态空间 $\Phi = \{1, 2, \dots, 600\}$, 容许行动集为紧集 $D(i) = [0.5, 20], i \in \Phi$ 。在策略 $v = (v(1), v(2), \dots, v(600))$ 下, 它的一条嵌入 Markov 链的转移概率矩阵满足

$$P(i, 1, v(i)) = 1 - e^{-\frac{v(i)}{i}}, P(i, j, v(i)) = \frac{1}{K} e^{-\frac{v(i)}{i}}, i = 1, 2, \dots, 600; j = 2, 3, \dots, 600$$

已知过程 Y 处于状态 i 下一次转移到 j, 它在状态 i 的逗留时间服从区间 $[0, jv(i)]$ 上的均匀分布, 即分布函数为

$$F(i, j, v(i), t) = \begin{cases} \frac{t}{jv(i)}, & 0 \leq t \leq jv(i) \\ 1, & t > jv(i) \end{cases}$$

$$Q(i, j, v(i), t) = P(i, j, v(i))F(i, j, v(i), t),$$

$$\text{性能函数为 } f(i, v(i)) = \ln[(i + i)v(i)] + \sqrt{i/2v(i)}。$$

根据前面的数值迭代并行算法, 在相同的初始策略和参数 $\varepsilon, \tilde{\varepsilon}$ 下, 得到下面的表示参与并行计算的处理节点数目和运算速度关系的表格:

表 3.4 参与并行计算的处理节点数目和运行时间表

处理节点数目	所用时间(秒)	处理节点数目	所用时间(秒)
1(即串行)	22.07	15	5.21
2	16.69	20	3.82
3	14.64	25	3.82
4	13.46	30	3.56
5	12.00	40	3.20
6	9.62	50	2.42
8	7.82	60	2.37
10	6.13	75	2.30

可以看出并行算法的运行时间较串行算法的运行时间大为缩短。理论上并行算法应该具

有线性加速比，即当有 N 个处理结点参与并行计算时其运算时间为串行算法的 $1/N$ ，正如图中虚线所表示的。但是在实际的仿真中，由于各个处理节点之间协作计算，因此必然存在通讯问题，例如上面并行数值迭代算法中第二步的广播操作，第五步的收集和广播操作都不可避免得涉及到数据的相互传递。由于通讯处理的开销一般比计算机处理器的计算开销大的多，因此就降低了并行效率，而且随着处理节点数目的增多通讯开销进一步增大；另外算法中还有无法并行化的部分，这一部分的时间开销并不随着处理节点数目的增多而减少。因此随着处理节点数目的增加加速比越来越小。在实际的并行优化过程中应该注意到这一点，根据状态数目选择合适的处理节点数目以达到最大的效费比。在本例中取 20 个左右的节点数目比较合适。

第四章 半 markov 控制过程分析与优化的实际应用

本章中,我们将给出排队网络的仿真与优化在两个实际问题中的应用。第一个例子为航空军械综合系统的维护问题,我们应用前面介绍的半 markov 决策过程的方法来解决这个问题,得到了相应的有用的系数。第二个例子,我们考虑了一个单节点的 ATM 开关,该节点的信包长度符合 Cox 分布。由于 Cox 的分布的网络属于非 Markov 系统(通过扩大空间状态的方法可以将其转化为等价 Markov 系统),因此在本章第二节我们首先简单介绍了 Cox 分布的一半情形——相形分布的基本特征。然后给出了基于对系统一条样本轨道的仿真获得性能指标对服务率参数导数的估计方法,最后将这一方法应用于一个实际 ATM 节点的性能势估计与梯度仿真。

4.1 半 Markov 决策过程在航空军械综合系统维护中的应用

航空军械综合系统的维护是一项复杂的系统工程,在维护过程中系统具有多个不同的状态。在某一时刻处于某一状态之下,在下一时刻可能转移到另一状态当中。系统处于不同状态的概率以及状态转移的概率等十维护的重要特性。经过分析,可用半 markov 模型来研究这一维护过程,用以指导实际的维护工作。[83]

4.1.1 模型的建立

航空军械综合系统的维护过程具有以下 6 种状态:

状态 1: 保存; 状态 2: 技术状态检查; 状态 3: 技术维护; 状态 4: 修理; 状态 5: 运用准备; 状态 6: 战斗应用。

经过分析研究和实践检验,在系统维护过程中状态转移具有以下特点:系统从状态 i 转移到状态 j 的概率为 π_{ij} ,这一概率仅仅由状态 i 的特殊性所限制,而与状态以前的状态无关;

在状态 i 停留的时间 θ_i 是一连续时间变量, θ_i 得分配规律不仅取决于当前状态 i ,而且取决于下一个时刻系统可能转移到的状态号,转移时刻为随机时间 θ 的累积值。

由此可知,系统维护的多状态转移是一半 markov 过程。用图 4.1 来描述系统维护的状态转移过程,图中带标号的方框表示维护的状态,状态的特性显示为系统处在该状态里的最小时间 $\theta_i = \min[\theta_{ij}]$ 。定向曲线表示维护过程中的状态转移,状态转移的特性显示为转移的

概 π_{ij} 显然,从一个状态出发转移到其他状态的概率满足以下关系式:

$$\begin{cases} \pi_{11} + \pi_{12} + \pi_{15} = 1 \\ \pi_{21} + \pi_{23} + \pi_{24} + \pi_{25} = 1 \\ \pi_{31} = 1 \\ \pi_{41} + \pi_{45} = 1 \\ \pi_{56} = 1 \\ \pi_{62} + \pi_{65} + \pi_{66} = 1 \end{cases} \quad (4.1)$$

4.1.2 模型的应用

根据实际情况，维护中系统的 6 种状态可分为两个不同的阶段：状态集{1, 2, 3, 4}构成系统维护的等待阶段 $G_{\text{等待}}$ ，而状态集{2, 4, 5, 6}构成系统维护的运用阶段 $G_{\text{运用}}$ ，系统的维护过程 $G = G_{\text{等待}} \cup G_{\text{运用}}$ ，由等待阶段和运用阶段交替组成，如图 4.2 所示。

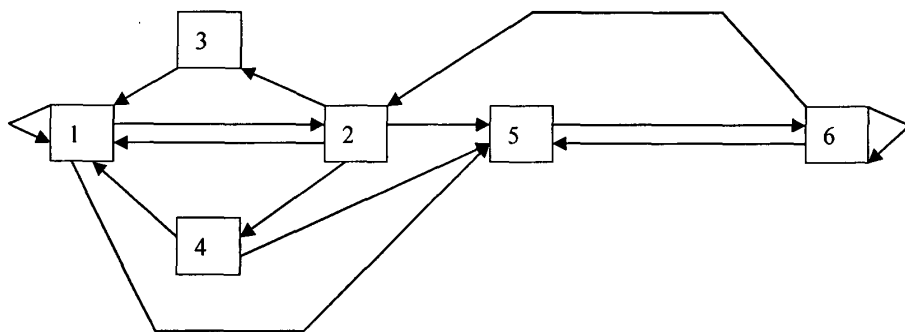


图 4.1 系统维护的状态转移过程

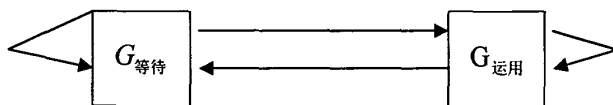


图 4.2 等待阶段与运用阶段的状态转移

4.1.2.1 等待阶段模型的应用

等待阶段状态转移曲线如图 4.3 所示。图 4.3 中有 4 个状态转移过程，分别是：1-1-1-2-1, 1-2-3-1, 1-2-4-1. 状态间转移概率记为 π_{ij}^* ，对于 π_{ij}^* 也满足类似 (4.1) 的关系式，从

而可进一步确定出 π_{ij}^* 与 π_{ij} 的关系式, 如 (4.2) 所示

$$\begin{cases} \pi_{11}^* = \frac{\pi_{11}}{1-\pi_{15}}, \pi_{12}^* = \frac{\pi_{12}}{1-\pi_{15}} \\ \pi_{21}^* = \frac{\pi_{21}}{1-\pi_{25}}, \pi_{23}^* = \frac{\pi_{23}}{1-\pi_{25}}, \pi_{24}^* = \frac{\pi_{24}}{1-\pi_{25}}, (4.2) \\ \pi_{31}^* = \pi_{31}, \pi_{41}^* = \frac{\pi_{41}}{1-\pi_{45}} \end{cases}$$

系统处在等待阶段的时间 θ , 应等于系统按任一状态转移过程转移的时间长度, 即

$$\theta_{\text{等待}} = \begin{cases} \theta_1, \text{如果 } 1-1 \text{ 且实现概率为 } \pi_{11}^* \\ \theta_1 + \theta_2, \text{如果 } 1-2-1 \text{ 且实现概率为 } \pi_{12}^*, \pi_{21}^* \\ \theta_1 + \theta_2 + \theta_3, \text{如果 } 1-2-3-1 \text{ 且实现概率为 } \pi_{12}^*, \pi_{23}^*, \pi_{31}^* \\ \theta_1 + \theta_2 + \theta_4, \text{如果 } 1-2-4-1 \text{ 且实现概率为 } \pi_{12}^*, \pi_{24}^*, \pi_{41}^* \end{cases}, (4.3)$$

为此, 系统处于等待阶段的时间 $\theta_{\text{等待}}$ 的数学期望为

$$\theta_{\text{等待}} = \theta_1 \cdot \pi_{11}^* + (\theta_1 + \theta_2) \cdot \pi_{12}^* \cdot \pi_{21}^* + (\theta_1 + \theta_2 + \theta_3) \cdot \pi_{12}^* \cdot \pi_{23}^* \cdot \pi_{31}^* + (\theta_1 + \theta_2 + \theta_4) \cdot \pi_{12}^* \cdot \pi_{24}^* \cdot \pi_{41}^*, (4.4)$$

系统处在等待阶段的平均时间为

$$t_{\text{等待}} = E[\theta_{\text{等待}}] = t_1 + \pi_{12}^* \cdot t_2 + \pi_{12}^* \cdot \pi_{23}^* \cdot t_3 + \pi_{12}^* \cdot \pi_{24}^* \cdot t_4, (4.5)$$

式中 $t_i = M[\theta_i]$ -- 系统处于第 i 状态的时间的数学期望。

最后可以得到, 在足够长的维护时间内, 在稳定状态下系统处在等待阶段不同状态的概率。

$$\rho_1 = \frac{t_1}{t_{\text{等待}}}, \rho_2 = \frac{\pi_{12}^* \cdot t_2}{t_{\text{等待}}}, \rho_3 = \frac{\pi_{12}^* \cdot \pi_{23}^* \cdot t_3}{t_{\text{等待}}}, \rho_4 = \frac{\pi_{12}^* \cdot \pi_{24}^* \cdot t_4}{t_{\text{等待}}}, (4.6)$$

通过记录统计获得 t_i 和 π_{ij} 后可以求得 $\rho_1, \rho_2, \rho_3, \rho_4$ 。

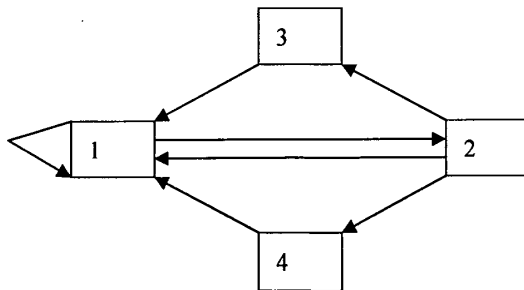


图 4.3

在等待阶段, 总的要求是使系统完好的处于保存状态 1 中以准备运用。它的 4 个状态转

移过程均是从状态 1 出发, 经过技术状态检查、技术维护或修理后, 最后返回到状态 1 中。因此, 等待阶段总的有效性指数称为技术准备系数, 可用系统处在状态 1 中的概率来表示,

$$\text{即 } K_{\text{技术准备}} = \rho_1 = \frac{t_1}{t_{\text{准备}}}, \quad (4.7)$$

系数 $K_{\text{技术准备}}$ 反映了系统转被战斗应用的保障情况。

4.1.2.1 运用阶段模型的应用

运用阶段状态转移曲线如图 4.4 所示。

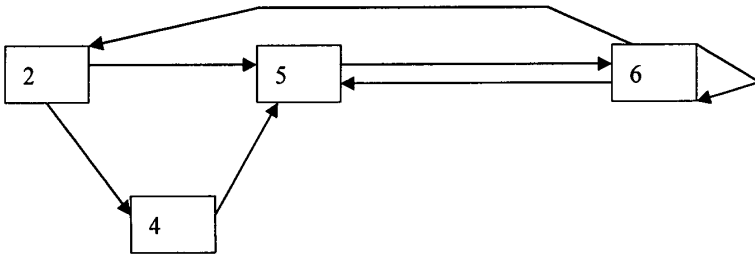


图 4.4 运用阶段的状态转移

同 (4.2) 相似, 可得图 4.4 中的转移概率, 满足下列关系式

$$\begin{cases} \pi_{24}^* = \frac{\pi_{24}}{1 - \pi_{21}}, & \pi_{25}^* = \frac{\pi_{25}}{1 - \pi_{21}}, & \pi_{45}^* = \frac{\pi_{45}}{1 - \pi_{41}}, \\ \pi_{56}^* = \pi_{56}, & \pi_{62}^* = \pi_{62}, & \pi_{65}^* = \pi_{65}, & \pi_{66}^* = \pi_{66} \end{cases}, \quad (4.8)$$

图 4.4 中有 4 个状态转移过程: 6-6, 6-5-6, 6-2-5-6, 6-2-4-5-6。利用与上节中类似的方法进行分析, 可以得到系统在运用阶段的平均时间

$$t_{\text{运用}} = E[\theta_{\text{运用}}] = t_6 + (\pi_{62}^* + \pi_{65}^*) \cdot t_5 + \pi_{62}^* \cdot \pi_{25}^* \cdot t_2 + \pi_{62}^* \cdot \pi_{24}^* \cdot \pi_{45}^* \cdot t_4, \quad (4.9)$$

在稳定状态下, 系统处在运用阶段不同状态的概率。

$$\rho_2 = \frac{\pi_{62}^* \cdot \pi_{25}^* \cdot t_2}{t_{\text{运用}}}, \rho_4 = \frac{\pi_{62}^* \cdot \pi_{24}^* \cdot \pi_{45}^* \cdot t_4}{t_{\text{运用}}}, \rho_5 = \frac{(\pi_{65}^* + \pi_{62}^*) \cdot t_5}{t_{\text{运用}}}, \rho_6 = \frac{t_6}{t_{\text{运用}}}, \quad (4.10)$$

在运用阶段, 总的要求是系统功能正常的投入到战斗应用之中, 其 4 个状态转移过程是从战斗应用出发, 最后返回战斗应用状态。因此, 运用阶段总的有效性指数成为战斗运用系数, 可用系统处于战斗应用状态的概率来表示, 即

$$K_{\text{战斗运用}} = \rho_6 = \frac{t_6}{t_{\text{运用}}}, \quad (4.11)$$

系数 $K_{\text{战斗运用}}$ 反映了系统出勤战斗任务的情况。

4.1.2.3 等待阶段和运用阶段交替过程中模型的应用

如图 4.2 所示，状态转移过程有： $G_{\text{等待}} - G_{\text{等待}}$ ， $G_{\text{等待}} - G_{\text{运用}} - G_{\text{等待}}$ ， $G_{\text{等待}} - G_{\text{运用}} - G_{\text{运用}}$ 。对照图 4.1 分析，在等待阶段系统可以从状态 1, 2, 4 出发分别以概率 π_{15} , π_{25} , π_{45} 进入到运用阶段的状态 5 中，而在运用阶段可以从状态 2, 4 出发以概率 π_{21} , π_{41} 进入到等待阶段的状态 1 中，若以 $\pi_{\text{等待}}$ 表示由等待阶段进入运用阶段的条件转移概率， $\pi_{\text{运用}}$ 表示由运用阶段进入等待阶段的条件转移概率，则

$$\begin{aligned} \pi_{\text{等待}} &= \frac{\rho_1 \pi_{15} + \rho_2 \pi_{25} + \rho_4 \pi_{45}}{\rho_1 + \rho_2 + \rho_4} \\ \pi_{\text{运用}} &= \frac{\rho_2 \pi_{21} + \rho_4 \pi_{41}}{\rho_2 + \rho_4}, \end{aligned} \quad (4.12)$$

假设在维护的开始时刻 ($t=0$)，系统处于等待阶段。以 $\theta_{\text{维护}}$ 表示整个维护过程的时间长度，

$$\theta_{\text{维护}} = \begin{cases} \theta_{\text{等待}}, & \text{如果 } G_{\text{等待}} - G_{\text{等待}} \text{ 且实现概率为 } 1 - \pi_{\text{等待}} \\ \theta_{\text{等待}} + \theta_{\text{运用}}, & \text{如果 } G_{\text{等待}} - G_{\text{运用}} - G_{\text{等待}} \text{ 且实现概率为 } \pi_{\text{等待}} \cdot \pi_{\text{运用}} \\ \theta_{\text{等待}} + \theta_{\text{运用}}, & \text{如果 } G_{\text{等待}} - G_{\text{运用}} - G_{\text{运用}} \text{ 且实现概率为 } \pi_{\text{等待}} \cdot (1 - \pi_{\text{运用}}) \end{cases} \quad (4.13)$$

$$\text{平均维护时间 } t_{\text{维护}} = E[\theta_{\text{维护}}] = t_{\text{等待}} + \pi_{\text{等待}} \cdot t_{\text{运用}} \quad (4.14)$$

如果假设维护开始在运用阶段，那么维护的时间长度将由下列表达式限制

$$\theta_{\text{维护}} = \begin{cases} \theta_{\text{运用}}, & \text{如果 } G_{\text{运用}} - G_{\text{运用}} \text{ 且实现概率为 } 1 - \pi_{\text{运用}} \\ \theta_{\text{等待}} + \theta_{\text{运用}}, & \text{如果 } G_{\text{运用}} - G_{\text{等待}} - G_{\text{运用}} \text{ 且实现概率为 } \pi_{\text{等待}} \cdot \pi_{\text{运用}} \\ \theta_{\text{等待}} + \theta_{\text{运用}}, & \text{如果 } G_{\text{运用}} - G_{\text{等待}} - G_{\text{等待}} \text{ 且实现概率为 } \pi_{\text{等待}} \cdot (1 - \pi_{\text{等待}}) \end{cases} \quad (4.15)$$

$$\text{平均维护时间 } t_{\text{维护}} = E[\theta_{\text{维护}}] = t_{\text{运用}} + \pi_{\text{运用}} \cdot t_{\text{等待}} \quad (4.16)$$

由于 (4.14) 与 (4.16) 所表达的维护时间是相等的，由此可得下列关系式

$$\frac{t_{\text{运用}}}{t_{\text{等待}}} = \frac{1 - \pi_{\text{运用}}}{1 - \pi_{\text{等待}}}, \quad (4.17)$$

根据 (4.14) 与 (4.16)，最后可以得到系统处在等待阶段和运用阶段的概率

$$K_{\text{等待}} = \frac{t_{\text{等待}}}{t_{\text{维护}}}, \quad K_{\text{运用}} = 1 - K_{\text{等待}}, \quad (4.18)$$

知道概率 $K_{\text{等待}}$ 和 $K_{\text{运用}}$ ，再结合系统处于等待阶段和运用阶段不同状态的概率，确定系统

处于各个不同状态的概率。

我们从航空军械综合系统多状态维护的特点出发，建立了半 markov 模型。将该模型应用于系统维护的等待阶段，运用阶段，等待和运用交替阶段，得到系统在各个不同状态的概率，

以及 $K_{\text{技术准备}}$ ， $K_{\text{战斗运用}}$ ， $K_{\text{等待}}$ ， $K_{\text{运用}}$ 等重要系数。这些概率和系数在维护中具有很重要

的作用，在平时和战时航空军械系统的维护中，对人力分配，资源配置，维修决策，计划和组织工作都能提供十分有力度的帮助。

4.2 M/PH/1 排队系统的性能分析及其应用

本节我们考虑一类非指数分布的排队网络的性能分析。因为在许多实际问题中这类网络有着较为广泛的应用。我们利用等价 Markov 性能势理论研究了 M/PH/1 排队系统的稳态性能灵敏度分析问题。并在此基础上，给出了一种该排队系统计算性能势及性能导数的仿真算法。最后给出了一个用于 ATM 交换机性能分析的实际算例。[82]

4.2.1 相型分布简介

下面我们简要地介绍一下相型分布的有关知识。首先考虑一个具有状态 $1, 2, \dots, m+1$ 的等价 Markov 过程，其无穷小矩阵为

$$\hat{Q} = \begin{bmatrix} T & t^0 \\ 0 & 0 \end{bmatrix}. \quad (4.19)$$

这里 T 为 $m \times m$ 矩阵，满足 $T_{ii} < 0$ ， $1 \leq i \leq m$ ； $T_{ij} \geq 0$ ， $i \neq j$ ； $t^0 = -Te$ ， $e = (1, 1, \dots, 1)'$ 。

“ $'$ ” 表示矩阵的转置。设 (α, α_{m+1}) 为等价 Markov 过程 \hat{Q} 的初始概率向量，

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m), \quad \alpha e + \alpha_{m+1} = 1.$$

我们设状态 $1, 2, \dots, m$ 都是瞬态，则等价 Markov 过程从任意初始状态出发，最终将肯定被状态 $m+1$ 吸收。

定义 1 定义在 $[0, \infty)$ 上的一个概率分布 $F(x)$ 称为一个 m 阶的相型分布，简称 PH-分布，如它是一个由 (4.19) 式定义且具有初始概率向量 (α, α_{m+1}) 的有限等价 Markov 过程，直到吸收的时间的分布。序对 (α, T) 称为 $F(x)$ 的一个表示。

状态 $1, 2, \dots, m$ 是瞬态的一个有用的等价条件，由下列引理给出。

引理 1 状态 $1, 2, \dots, m$ 是瞬态的充分必要条件为矩阵 T 是非奇异的。

此外, 不难证明

引理 2 定义概率分布 $F(x)$ 为

$$F(x) = 1 - \alpha \exp(Tx)e, \quad x \geq 0, \quad (4.20)$$

下面我们考虑一个具有实际意义的 PH-分布的例子。本节将主要研究这种形式的 PH-分布。考虑一个具有 k 个相位的服务台, 每个相位都具有指数服务分布, 服务率为 $\mu_i, i = 1, 2, \dots, k$, 到达服务台的顾客具有概率 α_i 进入相 i , $\sum_{i=1}^k \alpha_i = 1$ 。一位顾客

在相 i 完成服务后, 具有概率 $q_{i,j}$ 进入相 j ; 而具有概率 q_i 离开服务台, $\sum_{j=1}^k q_{i,j} + q_i = 1$ 。

置 $Q = [q_{i,j}]$ 。我们设矩阵 $I - Q$ 是非奇异的 (其中 I 为 k 阶单位阵), 以使得一位顾客最终离开该服务台的概率为 1。

现在我们考虑随机过程 $\bar{Q} = \{\bar{Q}_t; t \geq 0\}$, 其状态空间是 $\{1, 2, \dots, k+1\}$, 且具有初始概率向量 $(\alpha, 0)$, 其中 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ 。 $\bar{Q}_t = i, i = 1, 2, \dots, k$ 表示 t 时刻服务顾客处在相 i ; $\bar{Q}_t = k+1$ 表示 t 时刻顾客完成服务离开服务台。显然, \bar{Q} 为一个等价 Markov 过程, 其无穷小矩阵为

$$\bar{Q} = \begin{bmatrix} T & q \\ 0 & 0 \end{bmatrix}. \quad (4.21)$$

这里, $q = (q_1, q_2, \dots, q_k)'$, $T = \text{diag}(\mu_1, \mu_2, \dots, \mu_k)(Q - I)$ 。由于矩阵 $I - Q$ 是非奇异的, 从而状态 $1, 2, \dots, k$ 都是 \bar{Q} 的瞬态。故顾客在服务台的服务时间, 即过程 \bar{Q} 直到被状态 $k+1$ 吸收的时间, 是一个具有 k 阶表示 (α, T) 的 PH-分布。

4.2.2 系统模型与灵敏度公式

考虑 M/PH/1 排队系统, 它有一个服务者, 服务时间服从 k 阶 PH-分布, 其表示为 (α, T) , 这里 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$, $T = [T_{ij}]$ 为一个 $k \times k$ 的非奇异矩阵, 且满足 $T_{ii} < 0, T_{ij} \geq 0, i \neq j$ 。平均服务时间是 $\mu' = -\alpha T^{-1}e$ 。顾客的到达过程是一个到达率为 λ 的 Poisson 过程, 到达过程与服务分布相互独立。服务者有一个无限容量的缓冲器, 服务规

则是先到先服务 (FCFS)。

设 (Ω, F, P) 是服务时间随机变量定义在其上的概率空间, 则这样一个系统的状态过程可以用一个准生灭 (QBD) 过程 $N = \{N_t; t \geq 0\}$ 来加以描述, 其状态空间是 $E = \{0, (n, i): n \geq 1, 1 \leq i \leq k\}$ 。状态 0 相应于空排队; 状态 (n, i) 相应于有 n 个顾客在系统中, 且服务过程在相 i , N 的无穷小矩阵为

$$A = \begin{bmatrix} -\lambda & \lambda\alpha & 0 & 0 & \dots \\ t^0 & T - \lambda I & \lambda I & 0 & \dots \\ 0 & t^0\alpha & T - \lambda I & \lambda I & \dots \\ 0 & 0 & t^0\alpha & T - \lambda I & \dots \\ \cdot & \cdot & \cdot & \cdot & \dots \end{bmatrix}. \quad (4.22)$$

这里 $t^0 = -Te$, $t^0\alpha$ 表示 $k \times 1$ 矩阵 t^0 与 $1 \times k$ 矩阵 α 的乘积。

我们设 $\rho = \lambda\mu' < 1$, 则由文献[67]可知, N 有唯一的稳态概率向量 $\pi = (\pi_0, \pi_1, \dots)$, 这里 π_0 为一个数, π_m , $m = 1, 2, \dots$, 都是 k 维行向量。设 $\Phi = \{n: n \geq 0\}$ 为 QBD 过程 N 的离散状态空间, 这里 n 表示系统中的顾客数。设 $f: \Phi \rightarrow R$ 为一个性能函数, 且级数 $\sum_{n=0}^{\infty} |f(n)|\pi(n)$ 收敛, 这里 $\pi(n) = |\pi_n|$, $n = 0, 1, 2, \dots$, $|\pi_n|$ 表示向量 π_n 的所有分量之和。

系统的性能测度 η 定义为 f 关于稳态概率 π 的期望值, 即

$$\eta = \sum_{n=0}^{\infty} f(n)\pi(n) = \pi f. \quad (4.23)$$

其中 $f = (f_0, f_1, \dots)'$, f_n 是与 π_n 具有相同维数的行向量, 且它的每个分量都是 $f(n)$, $n \geq 0$ 。

现在设向量 α , 矩阵 T 均为在区间 $\Gamma \subset R$ 上变化的一个参数 θ 的可微函数, 则矩阵 A 的每个元素在 Γ 上都是 θ 的可微函数。我们也设 f 依赖于 θ , 且是 θ 的可微函数。因为性能向量 g 满足 Poisson 方程, 即 $Ag = -f + \eta e$, 两边对 θ 求导

$$\frac{\partial A}{\partial \theta} g + A \frac{\partial g}{\partial \theta} = -\frac{\partial f}{\partial \theta} + \frac{\partial \eta}{\partial \theta} e, \quad (4.24)$$

该式两边同时左乘 π , 由于 $\pi A = 0$, $\pi e = 1$, 故我们得到性能 η_f 关于参数 θ 的导数为

$$\frac{\partial \eta}{\partial \theta} = \pi \frac{\partial A}{\partial \theta} g + \pi \frac{\partial f}{\partial \theta}. \quad (4.25)$$

其中 $g = -A^\# f$ 为 Markov 过程 N 关于性能函数 f 的性能势向量, $A^\#$ 为无穷小矩阵 A 的群逆.

下面我们讨论一种特殊的情形。设上述 k 阶 PH-分布为

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k), \quad T = \text{diag}(\mu_1, \mu_2, \dots, \mu_k)(Q - I).$$

其中 $Q = [q_{i,j}]$ 。参数 θ 为服务率 $\mu_i, i = 1, 2, \dots, k$ 。用 T_i 表示矩阵 T 的第 i 行元素组成的行向量, 则易知

$$\mu_i \frac{\partial T}{\partial \mu_i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (4.26)$$

$$\mu_i \frac{\partial^0}{\partial \mu_i} = -\mu_i \frac{\partial T}{\partial \mu_i} e = - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T_i e \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (4.27)$$

和

$$\mu_i \frac{\partial(i^0 \alpha)}{\partial \mu_i} = \mu_i \frac{\partial^0}{\partial \mu_i} \alpha = \mu_i q_i \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (4.28)$$

这里 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$, 表示顾客在相 i 时离开服务台的概率。

现设 $\pi_m = (\pi_{m1}, \pi_{m2}, \dots, \pi_{mk})$, $m = 1, 2, \dots$, 我们有

$$\frac{\partial \eta}{\partial \mu_i} = \frac{\sum_{m=1}^{\infty} \pi_m B_m^{(i)} g}{\mu_i} + \pi \frac{\partial f}{\partial \mu_i}, \quad i = 1, 2, \dots, k. \quad (4.29)$$

其中

$$B_1^{(i)} = \begin{bmatrix} 0 & 0 & 0 & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \\ -T_i e & T_i & 0 & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \end{bmatrix}, \quad (4.30)$$

$$B_m^{(i)} = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & \dots & \dots \\ \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & \dots & 0 & \mu_i q_i \alpha & T_i & 0 & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & \dots \\ \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & \dots \end{bmatrix}, \quad m = 2, 3, \dots. \quad (4.31)$$

在上式右边的矩阵中, 前 $m-1$ 列的元素全为零。

4.2.3 仿真算法与算例

考虑一个无阻塞的单输入链接的 ATM 交换机, 它具有单类信源, 信源的到达是一个服从参数为 λ 的 Poisson 过程。输入链接有无限容量的缓冲器, 信源按照先到先服务 (FCFS) 规则进行排队, 通常可以假设信源在交换机内部的传输时间服从 Cox 分布 C_k [68]。系统的稳态通过量就是交换机的通量, 从而可以讨论交换机的性能分析。

上面所提到的 Cox 分布, 是本节所讨论的 M/PH/1 系统的一个特例, 即 $\alpha_1 = 1$, $\alpha_i = 0$, $i = 2, 3, \dots, k$, $q_{i,j+1} = a_i$, $q_{i,j} = 0$, $j \neq i+1$, $i = 1, 2, \dots, k-1$, $q_k = 1$ 的情形, 这样一个排队网络称为 k 阶 Cox 分布 C_k 。记为 $M/C_k/1$ 。显然, 它的表示 (α, T) 可简化为 $\alpha = (1, 0, \dots, 0)$, 及

$$T = \text{diag}(\mu_1, \mu_2, \dots, \mu_k) \begin{bmatrix} -1 & a_1 & & 0 \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & & & a_{k-1} \\ 0 & & & -1 \end{bmatrix}. \quad (4.39)$$

在此情况下, 将 $T_i = (\underbrace{0, \dots, 0}_{i-1}, -1, a_i, 0, \dots, 0)$ 和 $\alpha = (1, 0, \dots, 0)$ 代入即可求得稳态性能指标对服务率参数的导数。

对于上述问题, 我们可以通过对系统一条样本轨道的仿真, 计算系统的性能势与性能导数。

在对性能势的仿真估计中, 我们仍然采用第三章介绍的算法, 即通过对系统一条样本轨道上的有关信息 (即从状态 m 出发到达状态 n 的平均首达时间 $E\{S^{(m)}(n)\}$, 从状态 m 出发到达状态 n 的时间内性能的平均累积值 $E\{\int_0^{S^{(m)}(n)} f(N_t^{(m)}, v^{(0)}) dt\}$, 各状态的稳态概率及系统的平均性能指标 $J(v)$), 估计各状态对之间的实现因子 d_{nm} , $n, m \in \Phi$, 及相应的实现矩阵 $D = [d_{nm}]$, 再利用计算性能势向量 g 。

与第三章所讨论的闭排队网络不同的是, 本节我们讨论的是一个开排队网络, 且系统容量是无限的 (缓冲器容量无限), 因此, 状态数是无限的。为了避免无限维矩阵与向量计算带来的困难, 更由于实际仿真中无法对无限的状态对之间的有关量进行统计, 故在实际仿真运算中, 我们截取有限个状态来进行。因为实际系统中, 许多状态的出现概率是极小的, 所以这种近似可以达到相当的精度。

在 $M/C_k/1$ 排队系统中, 用 (s, i) 表示系统状态, $s \geq 0, 1 \leq i \leq k$ 。 s 表示系统内的顾客数, i 表示正在进行服务的位相。当 $s = 0$ 时, i 是没有意义的。我们称这个状态为 0 状态, 其序号亦为 0。当 $s \geq 1$ 时, 对状态 (s, i) 按字典序从 1 开始编号。显然, 状态 (s, i) 的序号为 $(s-1)k + i$ 。

另外, 我们所截取的状态数为 $K = Gk + 1$, 这里 G 为截取时相应的顾客数。当系统参数满足一定条件时, 系统中顾客数目超过 G 的概率是很小的。在实际仿真中, 若顾客数等于 G , 且又有新顾客到达, 则令顾客数保持不变。其余仿真运算均不变。

由于本仿真运算与第三章中所使用的算法有许多类似之处, 这里仅将基本步骤给出。

步 1. 在一条单一样本轨道上, 估计 $E[\int_0^{S^{(m)}(n)} f(N_t^{(m)}) dt]$, $E[S^{(m)}(n)]$, π 和 η (具体实现参见第三章第三节)。

步 2. 计算实现因子 d_{nm} 及相应的实现矩阵 D , 性能势 g 。

步 3. 计算 $\frac{\partial \eta}{\partial \mu_i}$ 。

现在，我们利用上述算法给出一个数值例子。

例考虑一个无阻塞的单输入链接的 ATM 交换机，具有 2 个服务相位，可看作一个 2 阶 Cox 分布 C_2 的排队网络。网络的参数为 $\lambda = 0.5$ ， $\mu_1 = 1$ ， $\mu_2 = 2$ ， $a_1 = 0.5$ 。性能函数为

$f(n) = n$ ， $n \in \Phi$ ，即 η 为服务者的稳态平均响应时间。

对一个 1,000,000 次的仿真，在状态空间 E 中分别取前 $K = 51, 101, 201$ 个状态，则势能和性能导数的均值如下。

对 $K = 51$ ，势能均值的前 14 个分量为：

-11.0497, -7.8563, -10.0621, -1.4815, -5.6348, 8.0026, 1.4398,
19.9349, 12.6265, 33.0496, 24.4435, 49.8409, 41.6763, 71.4179.

性能导数分别是 $\frac{\partial \eta}{\partial \mu_1} = -1.2355$ 与 $\frac{\partial \eta}{\partial \mu_2} = -0.8902$ 。

对 $K = 101$ ，势能均值的前 14 个分量为：

-11.4277, -8.2972, -10.8883, -1.8970, -6.4851, 8.3322, 2.1519,
20.0634, 12.5409, 36.0982, 23.6641, 53.1424, 39.4780, 73.2723.

性能导数分别是 $\frac{\partial \eta}{\partial \mu_1} = -1.2702$ 与 $\frac{\partial \eta}{\partial \mu_2} = -0.9129$ 。

对 $K = 201$ ，势能均值的前 14 个分量为：

-11.0872, -8.1040, -10.1992, -1.8670, -6.1492, 7.7688, 1.8232,
20.5913, 12.1763, 37.6151, 24.8103, 57.8630, 41.0601, 81.7367.

性能导数分别是 $\frac{\partial \eta}{\partial \mu_1} = -1.1571$ 与 $\frac{\partial \eta}{\partial \mu_2} = -0.9272$ 。

第五章 总结与展望

由于理论分析的方法对于一些复杂的问题难以进行全面、深入、细致的分析,尤其是在很多情况下难以得到系统主要指标的解析解。因此,计算机仿真方法也就成为分析这些复杂系统的重要手段,这一方法能够为各种类型的实际问题提供数值解。另外,理论研究中有时要作一些假设,这些假设需要鉴定是否合乎实际;而理论研究的一些结论也应该通过实际系统来检验。由于通常的实际系统都可以用计算机仿真来实现,这样就可以利用它来代替某些费用昂贵的试验,并且能够更快地积累数据,以便选择合适的假设及检验预测的结论。另外在计算机仿真过程中速度问题常常是需要关注的重点之一。

在本文中,我们利用仿真和并行计算的工具,研究了基于样本轨道的性能势的一种仿真算法,相对于基于实现矩阵的仿真方法,这种算法的缺点是精度有一定的降低,但是也具有突出的优点:运算量比较小,因而运算速度有较大的提高,并且仿真过程中需要的内存容量非常少,对于大状态空间的仿真问题中可以不考虑内存的限制。而在基于实现矩阵的仿真方法中内存容量不足常常成为算法应用的瓶颈之一。因此在实际优化问题中可以考虑该算法的应用,尤其是在硬件条件相对较低或者仿真速度要求较高的场合,例如半 Markov 控制过程优化问题中经常需要在新的策略下仿真系统的一条新的样本轨道,并在其上估计性能势等有关参数,进一步用于迭代计算。如果每次迭代过程中都能够提高运算速度减少仿真时间,则对优化过程提高速度是很有帮助的。该算法的并行化则通过在参与并行计算的各个处理节点上平均分配样本轨道,每个处理节点都执行相应的串行算法,仿真完毕后平均各个处理节点的结果作为最后的估计值。

对于半 Markov 控制过程优化问题,我们研究了几种优化算法,数值迭代算法和策略迭代算法,分别给出相应的基于仿真的优化算法、并行优化算法和并行仿真优化算法。分别比较了几种算法的结果,对于并行算法着重比较的是其运行时间并得到了满意的结果。

最后,我们给出了通讯中两个实际网络的优化与导数估计的应用实例。一是半 Markov 决策过程方法在航空军械综合系统维护问题中的应用;二是 M/PH/1 排队系统的性能灵敏度估计与仿真及其在 ATM 交换机性能分析中的应用。

排队网络的分析、仿真与优化还有许多问题需要解决。首先,当开排队网络的系统容量无限时,将涉及到(可数)无穷维状态空间的问题。其次是非半 Markov 型排队网络的理论分析问题。第三是缓冲器容量为有限情形时的研究,这时将涉及实际系统中面临的阻塞问题。

另外,在排队网络的优化研究中,需要特别关注的是基于仿真的优化。这是因为对很多实际系统而言,其网络参数并不是事先已知,而是需要随着系统的不断运行而不断测得的。在进行仿真时,须考虑两个问题。第一是运算速度问题。另一个则是该方法的结束条件在系统性能估计值具有随机性时,将导致假收敛。我们将对这些问题进行进一步的研究。

本论文是对作者近几年工作的总结,同时也是将来工作的起点。我们将在此基础上,继续深入进行这方面的理论研究,并继续考虑在一些实际问题中的应用。

参考文献

- [1] Ho Y C, Cassandras C. Computing costate variable for discrete-event systems, Proc. IEEE Conf. On Decision and Control, 1980, 690-700.
- [2] 郑大钟, 赵千川. 离散事件动态系统. 北京: 清华大学出版社, 2001.
- [3] 曹希仁. 离散事件动态系统. 自动化学报, 1985, 11(4):438-447.
- [4] Cao X R, Ho Y C. Models of discrete event dynamic systems. IEEE Control Systems Magazine, 1990, 10(4): 69-76.
- [5] Ho Y C, Shi L, Dai L, Gong W B. Optimizing discrete event dynamic systems via the gradient surface method. Decision and Control, 1991., Proceedings of the 30th IEEE Conference on, Vol.1, 1991, Page(s):104-109.
- [6] Ho Y C, Cassandras C. Computing costate variable for discrete-event systems, Proc. IEEE Conf. On Decision and Control, 1980, 690-700.
- [7] 胡奇英. 离散事件动态系统中的控制综合问题. 控制理论与应用, 1994, 11(5):557-563.
- [8] Di G S. On structural properties of discrete event dynamic systems Systems, Man, and Cybernetics, 1994. Humans, Information and Technology, 1994 IEEE International Conference on, Vol.1, 1994, Page(s): 1018-1023.
- [9] Uryasev, S Analytic perturbation analysis of discrete event dynamic systems. Computer Integrated Manufacturing and Automation Technology, 1994., Proceedings of the Fourth International Conference on, 1994, Page(s): 397-402.
- [10] Dai L Y, Ho Y C. Structural infinitesimal perturbation analysis (SIPA) for derivative estimation of discrete-event dynamic systems. Automatic Control, IEEE Transactions on, 1995, 40(7): 1154-1166.
- [11] Shi L. Discontinuous perturbation. analysis of discrete-event dynamic systems. Automatic Control, IEEE Transactions on, 1996, 41(11):1676-1681.
- [12] Dai L Y. Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems. Decision and Control, 1995., Proceedings of the 34th IEEE Conference on, Vol.3, 1995, Page(s): 2604-2609
- [13] Holloway L E, Krogh B H, Giua A. A survey of Petri net methods for controlled discrete event systems. Discrete Event Dynamic Systems: Theory and Applications, 1997, 7(2):151 — 190.
- [14] Retchkiman Z. Stability and stabilization techniques for discrete event systems modeled by coloured Petri nets. Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on, 2000, Page(s): 924-928
- [15] Me X. Performance optimization of discrete event systems with failures using fluid Petri nets, Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, Vol. 1, 2000, Page(s): 430-435
- [16] 徐心和, 李政国, 李彦平. 一类混杂系统的广义 Petri 网模型. 自动化学报, 1997, 23(3):297-301.
- [17] 蒋昌俊, 郑应平, 疏松桂. 基于行为表达式的任意随机 Petri 网的品质分析. 自动化学报, 1997, 23(3):370-376.
- [18] 徐志斌, 郑大钟. 同步和自环连接 Petri 网的性能保持分析. 控制理论与应用, 自动化学报, 1995, 15(1):109-113.
- [19] 殷保群. Markov 排队网络的性能指标灵敏度分析. 中国科技大学博士学位论文, 1998, 合肥.

- [20] Cao X R. First-order perturbation analysis of a single multi-class finite source queue. *Performance Evaluation*, 1987, 7(1): 31-41.
- [21] Ho Y C, Cao X R. *Perturbation analysis of discrete event dynamic systems*. Kluwer Academic Publisher, Boston, 1991.
- [22] Glasserman P. *Gradient estimation via perturbation analysis*. Kluwer Academic Publisher, Boston, 1991.
- [23] Fu M C, Hu J Q. Extensions and generalization of smoothed stochastic a generalized semi-Markov process framework. *IEEE Transactions on Automatic Control*, 1992, 37(10): 1483—1500.
- [24] Cao X R. *Realization probabilities: The dynamics of queueing systems*. New York: Springer-Verlag, 1994.
- [25] Fu M C, Hu J Q. Smoothed perturbation analysis derivative estimation for Markov chains. *Oper. Res. Letter*, 1994, 15(2): 241-251.
- [26] Bremaud P, Massoulié L. Maximal coupling rare perturbation analysis with a random horizon. *Discrete Event Dynamic Systems: Theory and Applications*, 1995, 5(2): 319-342.
- [27] Cao X R, Chen H F. Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control*, 1997, 42(10): 1382-1393.
- [28] Cao X R, Wan Y W. Algorithms for Sensitivity Analysis of Markov System through Potentials and Perturbation Realization. *IEEE Trans. on Control Systems Technology*, 1998, 6(4): 482-494.
- [29] 丁连宇, 杨德礼. 离散事件动态系统的摄动分析. *大连理工大学学报*, 1995, 35(1): 114-120.
- [30] 刘自宽, 谭思彤, 涂奉生. 具有固定生产周期的制造系统的控制策略及其实现. *控制与决策*, 1999, 14(3): 253-256.
- [31] Yao D D, Schechner Z. Decentralized control of service rates in a closed Jackson network. *IEEE Trans. Automat. Contr.*, 1989, 34(2): 236-240.
- [32] Ma D J, Cao X R. A direct approach to decentralized control of service rates in a closed Jackson network. *IEEE Trans. Automat. Contr.*, 1994, 39(7): 1460-1463.
- [33] Chong E K P, Ramadge P J. Stochastic optimization of regenerative systems using infinitesimal perturbation analysis. *IEEE Trans. Automat. Contr.*, 1994, 39(7): 1400-1410
- [34] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. New York: Wiley, NY 1994.
- [35] Bertsekas D P. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.
- [36] 沈挺, 赵千川, 郑大钟. 一种库存策略. *自动化学报*, 1999, 25(3): 337-343.
- [37] 康一梅, 郑应平. Job-shop 多机实时调度的并行算法. *控制与决策*, 1994, 9(2): 131-135.
- [38] 贾春福, 李勇建, 涂奉生. m 一机器 Flow Shop 随机调度问题. 第二十届中国控制会议, 2001, 8, 大连(2001): 766-770.
- [39] 贾春福, 涂奉生. 拖后工件数最少的单机随机调度问题. *南开大学学报(自然科学版)*, 1997, 30(4): 46-51.
- [40] 唐昊. Markov 控制过程的优化理论和算法. 中国科技大学博士学位论文, 2002, 合肥.
- [41] Cao, X R. The relations among potentials, perturbation analysis, and Markov decision processes. *Discrete Event Dynamic Systems: Theory and Applications*, 1998, 8(1): 71-78.
- [42] Cao X R. A unified approach to Markov decision problems and performance sensitivity

analysis. *Automatica*, 2000, 36(5): 771-774.

[43] Liu Z K, Tu F S. Single sample path-based sensitivity analysis of Markov processes. *IEEE Trans. on Automatic Control*, 1999, 44(4):872-875.

[44]殷保群, 周亚平.开排队网络的性能指标灵敏度公式.中国科学技术大学学报, 1998, 28(1): 27-32.

[45] Yin B Q, Zhou Y P, Xi H S, Sun D M. Sensitivity formulas of performance in two-server cyclic queueing networks with Phase-Type Distributed Service Times. *International Transactions in Operation Research*, 1999, 6(6): 649-663.

[46]殷保群, 周亚平, 杨孝先, 奚宏生, 孙德敏.状态相关闭排队网络中的性能指标灵敏度公式.控制理论与应用, 1999, 16(2): 255-257.

[47]王利存, 郑应平.基于仿真的可重入排队网络灵敏度分析.控制与决策, 2001, 16(1):37-41.

[48]殷保群, 奚宏生, 周亚平.MJG/1 排队系统的性能灵敏度分析, 高校应用数学学报, 2001, 6(3):235-242.

[49]周亚平, 奚宏生, 殷保群, 孙德敏.Markov 控制过程基于性能势的平均代价最优策略.自动化学报, 2001, 28(6): 896-902.

[50]奚宏生, 唐昊, 殷保群.Optimal policies for a continuous time MCP with compact action set.自动化学报(英文版), 2001., 29(2):245-250.

[51]唐昊, 奚宏生, 殷保群.Markov 控制过程在紧致行动集上的迭代优化算法.控制与决策, 2002, 18(3):348-352. .

[52] Tang H, Xi H S, Yin B Q, Zhou Y P. Performance optimization for a class of controlled closed queueing networks. The 2002 International Conference on Control and Automation, Xiamen, P. R. China. June,2002.

[53]奚宏生, 唐昊, 殷保群, 周亚平.Markov 控制过程在紧致行动集上的性能优化.第 21 届中国控制会议, 杭州, 2002 年 8 月.

[54]周亚平, 奚宏生, 殷保群, 唐昊.连续时间 Markov 决策过程在呼叫接入控制中的应用.控制与决策(增刊), 2001,16 (suppl.) :795~799.

[55] Zou C C, Xi H S, Yin B Q, Zhou Y P, Sun D M. Derivative Estimates Parallel Simulation Algorithms Based on Performance Potentials Theory, Proceedings of IFAC 14th World Congress, Beijing, 1999.

[56]邹长春, 周亚平, 殷保群, 奚宏生, 孙德敏.基于性能势理论对闭排队网络进行梯度估计的并行仿真算法.中国科学技术大学学报, 1999, 29(1):21-29.

[57]周亚平, 殷保群, 奚宏生, 谭小彬, 孙德敏.一类闭排队网络基于性能势的并行优化算法.中国科技大学学报, 2000, 30(2): 151-157.

[58] Cao X R. Single sample path-based optimization of Markov chains.*Journal of Optimization Theory and Applications*, 1999, 100(3):527-548.

[59]梁静国.管理系统仿真.哈尔滨船舶工程学院出版社, 1993.

[60]冯允成等.离散系统仿真.机械工业出版社, 1998.

[61]顾启泰.离散事件系统建模与仿真.清华大学出版社;1999.

[62]陈国良.并行计算—结构.算法.编程.高等教育出版社, 1999.

[63]黄恺, 徐志伟.可扩展并行计算技术、结构与编程.机械工业出版社, 2000.

[64]都志辉.高性能计算并行编程技术—MPI 并行程序设计.清华大学出版社, 2001.

[65]国家高性能计算中心(合肥).曙光 2000 用户手册, 合肥, 2001.

[66]曙光信息产业有限公司.曙光 2000 超级服务器用户手册, 1997.

- [67]邹长春.一类闭排队网络基于性能势梯度估计的并行仿真算法.中国科技大学硕士学位论文, 1999, 合肥.
- [68]高旭东.一类闭排队网络基于性能势梯度估计的并行仿真算法.中国科技大学硕士学位论文, 2000, 合肥.
- [69] Bertsekas D P, Tsitsiklis J N. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- [70] Y.C.Ho. *Overview of Ordinal Optimization, Proceedings of the 33rd Conference on Decision and Control*. Lake Buena Vista, Florida, 1975-1977.
- [71]盛友招.排队论及其在计算机通信中的应用.北京邮电大学出版社, 1998.
- [72] 代桂平, 殷保群, 周亚平, 奚宏生. M/G/1 排队系统的性能灵敏度估计与仿真.系统仿真学报, 2003, 15(7):950-952.
- [73] 殷保群, 代桂平, 周亚平, 奚宏生.闭排队网络基于并行仿真的灵敏度估计和优化算法.控制与决策, 2003, 18(3):348-350.
- [74] 代桂平, 殷保群, 王肖龙, 奚宏生.受控 M/G/1 排队系统的性能优化及迭代算法.系统仿真学报, 2004, 16(8):1683-1685.
- [75]代桂平, 殷保群, 李衍杰, 周亚平, 奚宏生.半 Markov 控制过程基于性能势仿真的并行优化算法.中国科学技术大学学报, 2005, 35 (2) :202-207
- [76] Yin Baoqun, Li Yanjie, Dai Guiping, Zhang Hu and Xi Hongsheng. Markov decision problems with unbounded transition rates under discounted-cost performance criteria. In *Proceeding of WCICA*. Hangzhou, RR. China. 2004
- [77]李衍杰, 殷保群, 奚宏生, 周亚平, 代桂平.半 Markov 过程基于性能势的灵敏度分析和性能优化.控制理论与应用, 2004.21(6):1032-1035.
- [78]殷保群,李衍杰, 唐昊, 代桂平, 奚宏生.半 Markov 决策过程折扣模型与平均模型之间的关系.控制理论与应用, 2004.19(6):691-694
- [79]代桂平 半 Markov 控制过程的性能灵敏度分析、优化及其算法研究, 中国科学技术大学博士学位论文, 2005, 合肥.
- [80] 马珊珊 半 Markov 控制过程在折扣代价准则下的性能分析及优化算法, 中国科学技术大学硕士学位论文, 2005, 合肥.
- [81] 周亚平, 刘剑宇, 殷保群, 奚宏生.半 Markov 过程性能势的并行仿真估计.系统工程, 2005, 23 (12) :103-108.
- [82]聂光戌, 魏贤智, 胡朝晖.航空军械综合维护系统的半马尔科夫模型.系统工程与电子技术, 2003, 25 (5): 534-536.
- [83]周亚平.Markov 型排队网络的优化理论及其仿真算法研究.中国科技大学博士学位论文, 2001, 合肥.

附录

[1] 周亚平, 刘剑宇, 殷保群, 奚宏生.半 Markov 过程性能势的并行仿真估计.系统工程, 2005, 23 (12) :103-108..

致谢

本论文是在我的导师周亚平副教授的精心指导下完成的。在课题的研究和论文的撰写工作以及学习生活中一直给予我耐心细致的指导和帮助。周老师深厚的学术功底，严谨的治学态度和浓厚的研究热情使我深受教益。在此谨向周老师表示衷心的感谢，致以真挚的敬意。

此外要感谢同一课题组的同学在课题研究所给予我的启示和帮助。

最后特别感谢我的家人和朋友对我的关心和帮助。

感谢所有帮助过我的人们。