随机性及其应用研究

On Randomness and Its Application

杨　帆　（石家庄市公路桥梁投资开发管理中心 050011）
郑建武　刘明生　（石家庄铁道学院信息工程系 050043）

摘要：本文首先讨论“随机性”与“高效计算”之间的关系，并强调引入“随机性”于问题求解的意义与重要性。随后给出产生“随机性”的现实途径及为计算引入“随机性”的两种不同方式，即“在线”方式与“离线”方式；通过对概率图灵机求解判定问题的讨论，来说明两种引入“随机性”方式之间的等价关系。最后，本文指出现实的随机算法设计与实现并没有为计算引入真正的“随机性”。

关键词：随机性　随机算法　概率图灵机　高效计算　在线　离线

1　简介

1977 年 Solovay 和 Strassen 发表了一篇关于素数测试的蒙特卡罗（Monte Carlo）方法[1]，把随机性引入算法设计与实现。在该篇文章中，素数测试的算法通过“抛硬币”的方式以帮助找到证明所输入的整数不是素数的实例。该算法基于如果输入的不是素数，那么就能够以很高的概率找到这样的反例。

1.1　随机性与“高效计算”

文献[1]的发表，使人们认识到必须改变对“高效计算”的认识与理解。在这之前，人们是把在多项式时间内由确定型图灵机（Deterministic Turing Machine）所完成的计算认为是“高效计算”，且把能够被确定型图灵机在多项式时间内解决的问题看作是“容易”的计算问题，即存在多项式 $p(\cdot)$，求解这些计算问题的算法复杂性为 $O(p(\cdot))$（不同类型的计算问题，多项式 $p(\cdot)$ 可以不同）。

事实上，当引入随机性于计算模型之后，应该把“高效计算”和能够在多项式时间内由概率图灵机完成的计算建立联系，即原来被认为是“难”处理的部分计算问题被归入“容易”的语言集合中。由此开辟了复杂性理论研究的新时代，概率计算（Probabilistic Computation）成为“高效”计算研究的一个重要领域。

本文随后的讨论，对随机性及其应用的论述是从计算模型生成与使用随机数的能力、方式等方面予以切入。

1.2　利用随机性的现实计算模型

提出概率图灵机（Probabilistic Turing Machine）是计算理论研究中的一个创新，该计算模型具有生成与使用随机数的能力。区别于传统或确定型图灵机（Deterministic Turing Machine），概率图灵机除具有输入、输出及工作条带（Input, Output, Scratch Tapes）之外，它还具有一承载随机数的条带，该条带被称为随机条带（Random Tape），这是从图灵机的构成上区别概率型与确定型图灵机。如果从图灵机的工作方式上来比较，概率图灵机相比确定型图灵机，具备生成与使用随机数的能力，是图灵机计算模型上的突破，而不是简单的改进。

除了确定型图灵机，还存在非确定型图灵机（Non-Deterministic Turing Machine）。相比非确定型图灵机，概率图灵机是一个现实的计算模型，而非确定型图灵机只是一个虚构的计算模型，提出该虚构模型，目的仅在于为方便复杂性类或问题集合的描述，如非确定型图灵机被用于描述问题。

2　计算过程引入随机性

本节讨论在计算过程中如何引入随机性，主要解决两个方面的问题，首先是随机性的产生，其次是如何引入随机性。

2.1　随机性的产生

于计算过程中引入随机性能够提升算法的计算能力并提高计算效率，则摆在我们面前的第一个问题就

是:"算法实现过程或概率图灵机执行过程中随机性如何产生?"更具体地讲,就是解决随机数的供应问题(随机位(数)是宝贵的计算资源)。

"抛硬币"试验为我们提供一条得到随机数的最基本、最简单的途径。但是,该随机试验每次所能提供的随机数的位数是有限的,即每次随机试验所引起的随机事件仅产生一个二进制随机位。

在求解计算问题时,我们把图灵机等价于现实世界中的计算机,当赋予图灵机产生与使用随机数的能力时,所对应的现实世界中的计算机是否能够执行"抛硬币"这样的基本操作? 基于对随机性的论述及计算机具备使用随机数生成设备(如不可靠电子电路)的能力,可以得出:计算机能够执行"抛硬币"这样的基本操作。

2.2 引入随机性的不同方式

如何为计算过程引入随机性,通过研究,我们可以总结以下两种不同的方式,即"在线"方式(On – Line)与"离线"方式(Off – Line)。

"在线"方式强调输入图灵机的变量只有一个,即问题的实例,图灵机运行过程中所需要的随机数(实质上是需要引入的随机性),是该图灵机通过某特定的随机性产生方式(不仅限于"抛硬币")自己生成的,即不依赖于其他设备。"在线"方式依靠图灵机自身完成随机数生成,那么该随机性产生的途径被认为是"内部抛硬币"(Internal Coin Toss)。

"离线"方式在强调图灵机使用随机数的能力的前提下,并不要求图灵机自己产生随机性的能力。即随机数的生成是由"外部抛硬币"(External Coin Toss)得到。这里的"外部"是指随机数的生成工作由外部随机数生成设备来完成,并把所生成的随机数提供给在算法执行过程中需要随机数的图灵机。

2.3 概率图灵机的语言判定问题

假定研究的是计算复杂性中的"判定"问题,即图灵机求解以下判定关系:"M(x)是否与X(x)相等?"其中X(x)为语言L的特征函数(Characteristic Function),函数定义如下:

$$X(x)=\begin{cases}1, \forall x\in L;\\ 0, \forall x\notin L.\end{cases}$$

由于在算法执行过程中引入了随机性,依"随机算法"的描述,即算法行为不仅取决于算法输入(实质上是问题实例),同时还取决于算法执行过程中所生成及使用的随机数。则在求解判定问题时,因随机性的引入,相同的输入(同一个问题实例),图灵机多次运行且停机之后的输出可能并不相同。即输入问题实例,因"内部抛硬币"(以下讨论,假定引入随机性的方式为"在线"方式)所生成的随机数不同,相应输出不同的判定结果$M_r(x)$,脚标r表示算法执行过程生成并使用的随机数为r。

因此基于概率图灵机的问题判定,就难以简单地描述为"M(x)是否与X(x)相等?",而应该以概率的方式来描述。即求解:

如果$x\in L$,$P_r[M(x)=X(x)]$,的概率取值;

如果$x\notin L$,$P_r[M(x)=X(x)]$的概率取值。

现在我们来计算对于任意给定的问题实例x,它被某一个概率图灵机M(·)判定并接受的概率为多少(允许$x\notin L$,$P_r[M(x)=1]>0$)? 我们已经知道,概率图灵机的判定输出不仅取决于输入的问题实例x,同时还取决于算法执行过程中的随机数r,但这里所考虑的问题是对"相同的问题实例",即x固定,因此该概率图灵机的判定输出就完全取决于算法执行过程中随机数r的取值。

不失一般性,假定概率图灵机$M_r(\cdot)$在给定问题实例x的前提下,需要生成并使用的随机数的位数是一个关于输入问题实例x的二进制描述位数的多项式,记为$T_M(|x|)$。则该概率图灵机在算法执行过程中所有可能被使用的随机数个数为$2^{T_M(|x|)}$。首先定义一个随机数集合S_{Accept},该集合中的任意单元为使得$M_r(x)=1$且位数为$T_M(|x|)$的随机数,即使得概率图灵机$M_r(\cdot)$判定"$x\in L$"的所有随机数r的集合。

$$S_{Accept}=\{r\in\{0,1\}^{T_M(|x|)}, M_r(x)=1\}。$$

由此,可以得到给定的问题输入x,经由概率图灵机M(·)判定并得到"$x\in L$"的概率可表示为:

$$P_r[M(x)=1]=\frac{|S_{Accept}|}{2^{T_M(|x|)}}。$$

相应概率图灵机M(·)判定并得到"$x\notin L$"的概率即为:

$$P_r[M(x)=0]=1-\frac{|S_{Accept}|}{2^{T_M(|x|)}}。$$

在计算复杂性的研究中,我们可以根据$x\in L$,$P_r[M(x)=1]$及$x\notin L$,$P_r[M(x)=1]$的不同概率取值,分

别定义不同的概率复杂性类，包括 RP，ZPP，BPP 等。

2.4 "在线"与"离线"方式的等价性讨论

"离线"方式眼中的图灵机能够接受两个输入，即实际问题实例的二进制描述与算法执行过程需要使用的随机数，因此可以把该随机数看作一辅助输入。则该图灵机在给定问题实例 x 和随机数 r 的判定输出可表示为 M(x,r)，即"离线"方式的 M(x,r) 等价于"在线"方式的 $M_r(x)$，即"在线"与"离线"两种方式针对算法执行过程利用随机性的能力是等价的，只是引入随机性的不同方式与描述。

在前面讨论概率图灵机的问题判定部分，为分析"给定问题实例 x，概率图灵机判定实例 $x\in L$ 或 $x\notin L$ 的概率取值"问题，我们假定概率图灵机 M(·) 在算法执行过程中需要生成并使用("在线"方式)位数为 $T_M(|x|)$ 的随机数；对于"离线"方式，该假定仍然有效且是必要的。有效性体现于研究概率复杂性过程中，要求对任意输入的问题实例，即 $\forall x\in\{0,1\}^*$，存在 P(|x|)，P(·) 为某一具体的多项式，图灵机能够在至多 P(|x|) 步运算之后停机并输出判定结果 $M_r(x)$ 或 M(x,r)。因运算步数至多为 P(|x|)，则在算法执行过程中所需要的随机数位数不可能超过所执行运算步骤的步数，即 $T_M(|x|)\leq P(|x|)$ 肯定成立。必要性为外部随机数生成设备所要求，即只有在这样的前提下，外部随机数生成设备才能够明确为求解任意的问题实例 $\forall x\in\{0,1\}^*$，它所需要生成并输出给概率图灵机的随机数位数。

3 随机性应用于求解排序问题

讨论随机算法的材料[2][3]，及分析计算复杂性的材料[4][5]，基本上无一例外地示例应用随机性来求解"快速排序"问题。

为了解如何构造求解快速排序问题的概率图灵机，请参阅[5]；对查找与排序问题的详细讨论及其相应算法设计与算法分析，请参阅[6]。

4 随机算法实现中的随机性评判

在求解现实计算问题时，我们的算法设计与实现也经常需要利用随机性来提升算法效率或降低算法复杂性，但是我们所采用的方式并没有为我们提供"真正"的随机性。

以下两步通常是我们所采用。首先，设定随机化种子；其次，调用 rand(·) 等计算函数得到随机数。事实上通过这些步骤未能提供"真正"随机性，理由在于，第一，每次算法调用，随机化种子的值固定不变；第二，rand(·) 等函数实质上是执行确定的计算步骤，即可以构造出与其等价的确定型图灵机。

5 结论与后续研究

本文通过给出产生"随机性"的现实途径及对计算引入"随机性"的不同方式的讨论，建立"随机性"与"高效计算"之间的现实联系，这对于理论计算机科学的研究与实践有重要作用。示例快速求解排序问题的概率图灵机的构造方法，分析现实随机算法设计与实现中所存在的问题，这些内容对计算问题的分析与求解，算法设计与分析，推动计算理论研究成果应用于实践等具有现实意义。

随机位(数)作为一类重要的计算资源，与其相关的以下研究将为后续研究的重要内容。第一，如何找到产生该计算资源的方式与途径。如单向函数(One-Way Function)所具备的"核心断言"可被视为"真正"的随机位(数)；第二，如何快速、高效地产生该计算资源。伪随机生成器(函数)为我们提供了一条可行途径，但是伪随机生成器所输出的不是"真正"的随机数。

参考文献

1 R. Solovay and V. Strassen. A Fast Monte-Carlo Test for Primality [J], SIAM Journal on Computing, 1977, 6: 84-85.

2 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. 算法导论 [M], 第 2 版. 北京：高等教育出版社. 2002.

3 M. H. Alsuwaiyel, 算法设计技巧与分析 [M], 北京：电子工业出版社. 2003.

4 Michael Spiser. 计算理论导论 [M], 北京：机械工业出版社. 2002.

5 John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. 自动机理论、语言和计算导论 [M], 第 2 版. 北京：清华大学出版社. 2002.

6 D. E. Knuth, 计算机程序设计艺术，第 3 卷 排序与查找 [M], 第 2 版. 北京：清华大学出版社. 2002.