并行计算:提高 SAT 问题求解效率的有效方法*

金人超[1] 黄文奇[2]

[1](华中理工大学计算机科学与技术学院 武汉 430074)

[2](中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

E-mail: jrc@163.net

摘要 基于拟物拟人思想的 Solar 算法是一个求解 SAT 问题的快速算法. 实验和理论分析表明, Solar 算法具有易并行化的特性. 将 Solar 算法并行化可大幅度地提高求解 SAT 问题的效率.

关键词 合取范式,可满足性,拟物法,拟人法,并行计算.

中图法分类号 TP301

Solar 算法是一个基于拟物拟人思想的求解 SAT 问题的快速算法[1]. 文献[1]显示, Solar 算法求解随机难 3-SAT问题快于美国 Bell 实验室的 Bart Selman 等人于 1994 年公布的 GSAT+w 算法[2]. 本文探讨了 Solar 算法的易并行化特性,证明并行化确实能有效地提高 Solar 算法的效率. 这在计算机硬件价格大幅降低的实际背景下,尤其具有现实意义.

1 SAT 问题

定义 1. 给定命题变元集合 $V=\{x_1,x_2,...,x_m\}$,文字是 V 中任意一个命题变元或者命题变元的非,子句是若干文字的析取,合取范式(conjunctive normal formula,简称 CNF)是若干子句的合取.

定义 2. 给定命题变元集 $V=\{x_1,x_2,...,x_m\}$,命题变元 $x_1,x_2,...,x_m$ 的任意一组取值称为一个真值指派,记为 X. 若用 1 表示真,用 0 表示假,则 $X\in\{0,1\}^m$. 一个真值指派满足了某个 CNF(或子句)是指该 CNF(或子句)在这一真值指派下为真.

定义 3. 合取范式可满足性(SAT)问题是指,给定 CNF,问是否存在一个满足该 CNF 的真值指派. 3-SAT 问题是限定 CNF 中每个子句恰有 3 个文字的 SAT 问题.

定义 4. 选定命题变元集 $V=\{x_1,x_2,...,x_m\}$,每次随机地从 V 中挑选 3 个两两不同的命题变元,以 50% 的概率决定每个变元是否取非,这样得到 3 个文字组成一个子句. 重复上述步骤 l 次,得到 l 个子句组成一个 CNF. 按此方法产生的 CNF 称为一个长度为 l 的 m 元的随机产生的 3-SAT 样例. 当 $l/m\approx 4.3$ 时,称为随机产生的难 3-SAT 样例[3].

SAT 问题和 3-SAT 问题都是 NP 完全的[4],目前不存在完整的多项式时间算法. 随机产生的难 3-SAT 样例是目前学术界普遍采用的测试 SAT 问题算法的考题.

2 Solar 算法及其计算时间的分布

Solar 算法是一个随机型算法. 在同一个计算格局下,随机地从多个可能的计算动作中选取一个执行. 这样做是为了在尽量不增加算法的时间复杂度的前提下提高算法的完整性. 若采用贪心方法,则有可能使计算无法

---

* 本文研究得到国家自然科学基金(No. 19331050)、国家 863 高科技项目基金(No. 863-306-05-03-1)、国家"九五"攀登计划基金、高等学校博士学位点专项科研基金(No. 96048703)和中国科学院软件研究所计算机科学开放研究实验室课题基金资助. 作者金人超,1965 年生,博士,副教授,主要研究领域为计算复杂性理论,近似算法. 黄文奇,1938 年生,教授,博士生导师,主要研究领域为计算复杂性理论,求解 NP 难度问题的拟物拟人算法.

本文通讯联系人:金人超,武汉 430074,华中理工大学计算机科学与技术学院

本文 1998-06-22 收到原稿,1999-04-13 收到修改稿

远离局部最优的“陷阱”,导致计算发生一种弱的死循环;若采用穷举每条计算路径的方法,则又使计算时间呈指数上升.在目前没有更好的选择计算路径的策略的情况下,采取随机选择的方法是合理的[5].

Solar 算法中的这种随机性导致其对同一个样例的每次计算具有不同的计算时间.通过实验我们发现,对绝大多数随机产生的难 3-SAT 样例,Solar 算法对同一个样例的多次计算的计算时间之间的差别相当可观.

图 1 显示了 Solar 算法对一个有 1 000 个命题变元随机产生的难 3-SAT 样例的 500 次计算的计算时间分布情况.图 1 中每个竖长方条的宽度是 2s,表示一个时间段;长方条的高度表示计算时间落在该时间段内的计算的次数.从图中可以看出,500 次计算的平均时间为 35s,而计算时间分布的峰值在 $t_0=6\sim8$s 这一时间段.在此时间段之后,计算时间的分布量呈递减的趋势.

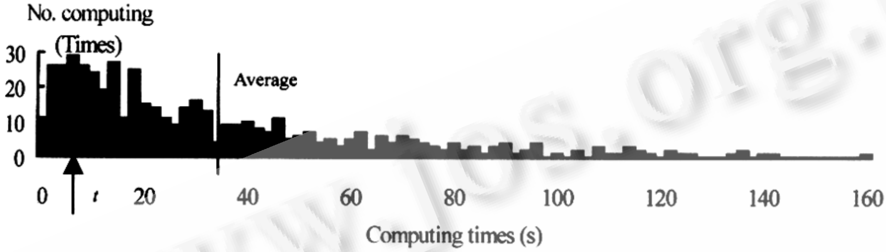

Fig. 1

图1

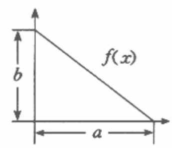

Fig. 2

图2

3 并行化

为简化讨论,我们用如图2所示的概率密度函数 $f(x)=b-\frac{b}{a}x$ 来近似图1中的概率分布.

由 $\int_0^a f(x)\mathrm{d}x=\int_0^a\left(b-\frac{b}{a}x\right)\mathrm{d}x=1$ 可知,$b=\frac{2}{a}$,故 $f(x)=\frac{2}{a}-\frac{2}{a^2}x=\frac{2}{a^2}(a-x)$,一次计算的时间不超过 t 的概率为

$$p(x\leqslant t)=\int_0^t f(x)\mathrm{d}x=\frac{1}{a^2}(2at-t^2),$$

k 次计算的最短时间不超过 t 的概率为 $p^{(k)}(x\leqslant t)=1-(1-p(x\leqslant t))^k=1-\frac{1}{a^{2k}}(a-t)^{2k}$,$k$ 次计算的最短时间 t 的分布概率密度函数为 $f^{(k)}(t)=\frac{\mathrm{d}}{\mathrm{d}t}(p^{(k)}(x\leqslant t))=\frac{2k}{a^{2k}}(a-t)^{2k-1}$,$k$ 次计算的最短时间 t 的期望值为 $E^{(k)}=\int_0^a t\cdot f^{(k)}(t)\mathrm{d}t=\int_0^a t\cdot\frac{2k}{a^{2k}}(a-t)^{2k-1}\mathrm{d}t$,作变量代换 $y=a-t$,得

$$E^{(k)}=-\int_a^0(a-y)\cdot\frac{2k}{a^{2k}}\cdot y^{2k-1}\mathrm{d}y=-\frac{2k}{a^{2k-1}}\int_a^0 y^{2k-1}\mathrm{d}y+\frac{2k}{a^{2k}}\int_a^0 y^{2k}\mathrm{d}y=\frac{1}{2k+1}a,$$

当 $k=1$ 时,$E^{(1)}=\frac{1}{3}a$ 就是一次计算的时间的期望值.因此,k 台计算机并行计算的加速比为

$$S_k=E^{(1)}/E^{(k)}=\left(\frac{1}{3}a\right)\Big/\left(\frac{1}{2k+1}a\right)=\frac{2k+1}{3},$$

并行效率为 $F_k=S_k/k=\left(\frac{2k+1}{3}\right)/k=\frac{2}{3}+\frac{1}{3k}$.

由此可见,对于我们的近似模型,采用并行计算可以达到线性加速,即计算速度的提高倍数与参加计算的计算机台数成正比,这是并行算法所能达到的理想境界.但 $f(x)=b-\frac{b}{a}x$ 的假定与实际情况有误差,主要表现在实际计算时间分布的峰值处在一个大于0的 t_0 位置上,尽管 t_0 相对 a 来说非常小.这样,当我们用多台计算机将计算时间加速到 $\leqslant t_0$ 时,进一步加速的效率会有所下降.但在此之前,加速效果应该是好的.

我们在10台微机上同时对两组随机生成的难3-SAT 样例进行了试算,每组算出10个可满足的样例统计计算时间.结果见表1和表2.所用的微机为多能奔腾586(Intel MMX pentium),主频166MHz,内存32M,操作系统为

Linux,程序用C语言实现.实验结果接近于我们的理想近似模型.

Table 1
表1

No. variables: 1000[①]; No. clauses: 4250[②] Time unit: s[③]

Problems[④]	1	2	3	4	5	6	7	8	9	10	Average[⑤]
Average computing times $\overline{T}$[⑥]	2.0	9.7	27.9	54.7	30.1	2.9	14.7	1.4	12.4	3.3	
Minimal computing times T_{min}[⑦]	0.7	1.8	1.7	5.4	3.3	0.9	4.5	0.8	3.2	0.5	
Accelerate rates $S=\overline{T}/T_{min}$[⑧]	2.9	5.4	16.4	10.1	9.1	3.2	3.3	1.8	3.9	6.6	6.3
Parallel efficiency $E=S/10$[⑨]	0.29	0.54	1.64	1.01	0.91	0.32	0.33	0.18	0.39	0.66	0.63

①命题变元个数:1000,②子句个数:4250,③计时单位:秒,④题号,⑤平均,⑥平均计算时间 $\overline{T}$,⑦最短计算时间 T_{min},⑧加速比 $S=\overline{T}/T_{min}$,⑨并行效率 $E=S/10$.

Table 2
表2

No. variables: 2000[①]; No. clauses: 8500[②] Time unit: s[③]

Problems[④]	1	2	3	4	5	6	7	8	9	10	Average[⑤]
Average computing times $\overline{T}$[⑥]	294.0	47.9	534.0	226.2	105.5	174.0	194.2	75.3	218.3	387.1	
Minimal computing times T_{min}[⑦]	18.7	5.0	107.5	21.7	14.2	64.7	36.0	15.0	38.6	113.3	
Accelerate rates $S=\overline{T}/T_{min}$[⑧]	15.7	9.6	5.0	10.4	7.4	2.7	5.4	5.0	5.7	3.4	7.0
Parallel efficiency $E=S/10$[⑨]	1.57	0.96	0.50	1.04	0.74	0.27	0.54	0.50	0.57	0.34	0.70

①命题变元个数:2000,②子句个数:8500,③计时单位:秒,④题号,⑤平均,⑥平均计算时间 $\overline{T}$,⑦最短计算时间 T_{min},⑧加速比 $S=\overline{T}/T_{min}$,⑨并行效率 $E=S/10$.

本文的结论是:(1) Solar 算法的计算时间的分布规律使其天然地适合并行计算;(2) 理论分析表明,并行计算可以使 Solar 算法得到线性加速;(3) 实验证明,对于规模为1 000和2 000个命题变元的随机生成的难3-SAT样例,用10台计算机即可将平均计算速度提高6~7倍左右,使并行效率达到0.6~0.7左右.因此,用 Solar 算法进行并行计算是求解 SAT 问题的有效方法.

参考文献

1 Huang Wen-qi, Jin Ren-chao. The quasiphysical and quasisociological algorithm solar for solving SAT problem. Science in China (Series E), 1997,27(2):179~186
(黄文奇,金人超.求解 SAT 问题的拟物拟人算法——Solar.中国科学(E辑),1997,27(2):179~186)

2 Selman B, Kautz H A, Cohen B. Noise strategies for improving local search. In: Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94). Seattle, Washington, D C: AAAI Press, 1994. 337~343

3 Mitchell D, Selman B, Levesque H J. Hard and easy distributions of SAT problems. In: Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92). San Jose, CA: AAAI Press, 1992. 459~465

4 Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman and Company, 1979

5 Motwani R, Raghavan P. Random Algorithms. London: Cambridge University Press, 1995. 127~129

Parallel Computing: An Effective Method for Improving the Efficiency of Solving SAT Problems

JIN Ren-chao[1] HUANG Wen-qi[2]

[1](*School of Computer Science and Technology Huazhong University of Science and Technology Wuhan* 430074)
[2](*Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing* 100080)

Abstract Based on the thought of quasiphysical and quasisociological, the Solar algorithm is an efficient algorithm for solving SAT problems. It is proved by the theoretical analysis and experimental results that the algorithm is naturally suitable for parallel computing. The efficiency of solving SAT problems can be greatly improved by taking a simple method that parallelizes the Solar algorithm.

Key words CNF (conjunctive normal formula), satisfiability, quasiphysical, quasisociological, parallel computing.