

并行蚁群算法求解加权 MAX-SAT *

孙如祥^{1,2}, 唐天兵¹, 李炳慧¹

(1. 广西大学 计算机与电子信息学院, 南宁 530004; 2. 广西职业技术学院 计算机与电子信息工程系, 南宁 530226)

摘 要: 为了使得算法对蚁群进化的控制更加直接、算法更加高效, 针对加权 MAX-SAT 的特点, 以重离散化方式简化蚁群算法模型, 提出取值概率的概念, 并以之替换传统蚁群算法中信息素, 最后对该算法作并行化改进。实验结果表明, 得到的基于改进后并行化的蚁群算法更具有效性, 搜索时间明显降低, 取得了较好的加速比和效率。

关键词: 蚁群算法; 加速比; 并行; 最大化可满足性问题 (MAX-SAT); 加权 MAX-SAT; 多核

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695 (2012)01-0049-03

doi:10.3969/j.issn.1001-3695.2012.01.013

Parallel ant colony algorithm for weighted MAX-SAT

SUN Ru-xiang^{1,2}, TANG Tian-bing¹, LI Bing-hui¹

(1. School of Computer, Electronics & Information, Guangxi University, Nanning 530004, China; 2. Dept. of Computer & Information Electronic Engineering, Guangxi Vocational & Technical College, Nanning 530226, China)

Abstract: For making the algorithm more direct control, the algorithm more efficient, this paper weighted the characteristics of the MAX-SAT, simplified discrete approach to re-model of ant colony algorithm and proposed the concept of probability values, replaced the traditional elements of pheromone ant colony algorithm, finally made parallel improvements of the algorithm. Experimental results show that the new algorithm is more parallel efficiency, reduces the search time significantly, and achieves good speedup and efficiency.

Key words: ant colony algorithm; speedup; parallel; maximum satisfiability problem (MAX-SAT); weighted MAX-SAT; multi-core

0 引言

可满足性问题 (SAT) 是计算复杂性理论与人工智能科学中的一个核心问题。最大化可满足性问题 MAX-SAT 是 SAT 问题的一个扩展, 它具体包含有两种形式: 以最大化满足子句的数为目标; 针对子句加权的情况, 以最大化满足子句权重和为目标。MAX-SAT 问题重要的求解器主要包括美国的新墨西哥技术大学 Borchers 等人开发出来的 BF^[1]、法国研究人员 De Givry 等人研究的 toolbar^[2]、西班牙科研人员 Alsinet 等人开发的 Lazy^[3], 还有我国中山大学的 Lin Han 等人研究的 LB-SAT^[4]。经典的局部算法含有 WSAT、TSAT、GSAT、NSAT 等。

并行计算能够提高计算机处理问题的效率和运算速度。并行计算的优势在于它具有庞大的数据处理和数值计算的能力, 有很强的实际应用背景, 如新型武器的设计开发、卫星图像的处理、天气预报和地壳运动的监测、能源的勘探和开采、虚拟与现实技术的实现等。蚁群算法具有天然的并行性, 人们普遍认为在并行化的蚁群算法中, 信息交流模式和周期是算法的重要因素^[5], 这些会影响算法的优化性和解的成熟情况。但是现在的并行化算法^[5]会根据全局的最优解调整信息素, 导致产生相似解, 从而大大降低算法搜索效率。文献[6]中则提出

了规定周期数再进行信息交流。尽管常数化的周期对算法会有影响, 但同时也会对算法的收敛和多样性产生影响, 所以这些并行化的蚁群算法还不是最优的, 需要继续完善。

本文提出一种多核 CPU 环境下的并行蚁群算法, 用于求解加权 MAX-SAT 问题。首先, 将问题进行重离散化, 用取值概率代替信息素机制, 以提高算法求解的效率和质量; 其次, 将蚁群算法进行并行化改进, 让算法在多核环境下并行化地寻找问题的最优值, 再让算法求解加权 MAX-SAT 问题, 并分析并行算法的加速比和效率。实验结果表明, 该算法求解问题的时间明显减少, 取得了较好的加速比和效率。

1 WMSAT

加权 MAX-SAT (WMSAT) 是一个典型的 NP 难问题。当给每个子句加上一定的权重后, MAX-SAT 就变成了加权的最大化可满足性问题 (weighted maximum satisfiability problem, WMSAT)。WMSAT 问题是要找到合取范式中的真值指派, 让所有不满足子句的权值加起来最小化, 或者是满足的子句的权值和最大化。

对于 WMSAT, 其构成要素与 SAT 基本相同, 不同之处在于其子句为加权子句。若用 (c, w) 表示, 则 c 为子句, w 为对应

收稿日期: 2011-06-16; **修回日期:** 2011-07-22 **基金项目:** 国家自然科学基金资助项目 (50605010); 广西教育厅科研资助项目 (200911LX15); 广西大学科研资助项目 (XJZ110585)

作者简介: 孙如祥 (1983-), 男, 广西北海人, 网络工程师, 硕士研究生, 主要研究方向为计算机网络技术、计算机教育 (srx@gxzyjy.com); 唐天兵 (1972-), 男, 四川成都人, 副教授, 硕导, 硕士, CCF 会员, 主要研究方向为并行分布式计算和优化算法; 李炳慧 (1986-), 女, 山东济南人, 硕士研究生, 主要研究方向为并行分布式计算和优化算法。

权重,其一般为一正整数,因此可将 WMSAT 看做 SAT 中子句 c 重复出现 w 次。WMSAT 的目标是寻找一真值指派,使所有为真的子句权重和最大。若以 $c(x)$ 表示子句 c 在真值指派 x 下进行逻辑运算的结果,子句为真时其为 1,否则为 0,则对具有 m 个子句的 WMSAT 的目标可以转换为如下的优化函数:

$$\max f(x) = \sum_{i=1}^m c_i(x) w_i \quad (1)$$

每次产生新的解以后,可以根据上述的函数对其进行评价,函数的值越大则对应的解越好,反之亦然。对这一类函数的优化,人们已经提出了大量的研究方法。通过式(1)的转换,可以非常方便地将这些方法运用到 WMSAT 或者 SAT(将各子句的权重视为 1)的求解上,如文献[7]通过构造图的方式,成功地将改进的最大最小蚁群系统(蚁群算法的一种变化)用于求解 SAT,并且取得了比较好的效果。

2 算法的改进与实现

2.1 改进策略

蚁群算法的传统模型将问题的解抽象为由一些边连接而成的路径,并以信息素的概念对蚂蚁激素进行描述。此种直观的方式能很好地模拟自然界蚂蚁的行为,但是对某些具体问题的求解,如对 SAT,却会增加不必要的复杂性。针对 WMSAT,本文对蚁群算法作了以下改进:

a)问题的重离散化。在 SAT 中,各变量的取值范围都相同,均为{T, F}(为方便计算,求解过程可取{1, 0}),即各变量可取相同的值,变量间不存在非重复的制约,因此若再以边将各变量关联起来则显得多余,问题也会变得更复杂。由此,用蚁群算法求解 WMSAT,考虑去除边的因素,而将顶点作为研究对象,独立处理各个变量,实现对问题的重离散化,从而简化问题的求解模型。

b)变量取值的直接化。在说明具体操作过程前,先引入一个概念——取值概率,其表示变量选取某值的概率大小。对于 WMSAT,每个变量可以选择的值为 T 或 F,若 T 的取值概率为 0.6,则表明在一次赋值过程中,给该变量赋 T 值的概率为 0.6。操作时可产生一个[0, 1]间的随机数,若其不大于 0.6 则可以给该变量赋值为 T,否则为 F。有了取值概率,则可以以之取代原算法中信息素,并通过它实现对变量取值的直接控制,最终实现对蚁群进化的直接控制。因为当各个变量对 T 与 F 的取值概率均确定后,算法就可以直接根据此概率进行新的真值指派,于是得到新的候选解。与信息素类似,取值概率的计算也发生在两种情形:

a)初始化阶段。此时由于还没有蚂蚁走过路径的信息,计算取值概率所需信息主要由问题本身提供。对具有 n 个变量、 m 个子句的 WMSAT,变量 $x_i (i = 1, 2, \dots, n)$ 的初始取值概率通过下式计算:

$$p(x_i) = \frac{\sum_{j=1}^m c_j(x_i) \times w_j}{\sum_{j=1}^m c_j(x_i) \times w_j + \sum_{j=1}^m c_j(-x_i) \times w_j} \quad (2)$$

$$p(-x_i) = \frac{\sum_{j=1}^m c_j(-x_i) \times w_j}{\sum_{j=1}^m c_j(x_i) \times w_j + \sum_{j=1}^m c_j(-x_i) \times w_j} \quad (3)$$

式(2)计算的是变量 x_i 正文字的取值概率(x_i 取 1 的概率),式(3)为变量 x_i 负文字的取值概率(x_i 取 0 的概率)。两式中,若文字 x_i 出现于 c_j 中则 $c_j(x_i)$ 为 1,否则为 0。从这两式可以看到, $p(x_i)$ 、 $p(-x_i)$ 分别通过正、负文字所在子句的权重

和计算得到,权重和越大则对应初始取值概率也越大,其反映的取值趋势与所求解的目标一致,因此初始取值概率的引导作用是明显的。另外,由于 $p(-x_i) = 1 - p(x_i)$,求解过程中可只计算 $p(x_i)$,然后按此式计算 $p(-x_i)$ 即可。

b)利用路径信息对取值概率进行更新。而选取的路径一般为当前最优路径或者全局最优路径,本文选择全局最优路径来更新各变量的取值概率。对 WMSAT 而言,全局最优路径与全局满足子句权重和最大的真值指派对应。设该真值指派为 $x_{\text{global_best}}(b_1, b_2, \dots, b_n)$ (b_i 用 0 或 1 表示),对应子句权重和为 $w_{\text{global_best}}$,而问题可能的最大子句权重和为 w_{object} ,则变量 $x_i (i = 1, 2, \dots, n)$ 的取值概率的更新式为

$$p(x_i) = (1 - \chi) \times p(x_i) + \chi \times b_i \times \frac{w_{\text{global_best}}}{w_{\text{object}}} \quad (4)$$

$$p(-x_i) = 1 - p(x_i) \quad (5)$$

其中, χ 是更新系数,表明新路径对蚁群进化的影响,其取值越大则新路径的影响越大。为避免蚁群的进化产生剧烈震荡,一般 χ 取较小的值,如 0.05。各变量的取值概率经过上述更新后,算法则可利用其进行新一轮的真值指派,产生更优的解。

2.2 并行化处理

本文中将采用两种并行策略:a)并行的蚂蚁搜索策略,此方法策略中,蚁群算法中的各个蚂蚁独立执行搜索过程;b)并行的蚂蚁评价策略,此方法策略中,蚁群算法以并行形式对各路径进行评价。

以上策略不需要对原有串行的蚁群算法进行结构上的改造,借助 OpenMP,只需要添加一定的编译指导语句就可以实现并行计算,从而提高了算法的运行速度和求解效率,具有很高的适应性。

根据算法的并行性特征和计算能力等通信代价,实验依据处理器的数目对蚂蚁进行群体分类,让不同的蚂蚁在不同的处理器上独立地执行搜索任务;在一定时间以后,进行各个蚂蚁的信息交换和最优解的比较。通过评价找出最短路径并且找出全局最优解。其中算法的程序流程如下(其中:loopnum1 表示算法迭代次数,loopnum2 表示蚁群中的蚂蚁数目):

```
begin
initiation
for i = 1 to loopnum1
do in parallel
for j = 1 to loopnum2
single ant search path
end for
end do
do in parallel
for j = 1 to loopnum2
evaluate an ant
end for
end do
select optimize path
scatter optimize path
end for
end
```

3 仿真实验

为验证算法改进及并行化的有效性,本文选取了两组不同类型的 WMSAT 对其进行测试,一组拥有不同变量个数,另一组拥有不同子句数,测试数据来源于 <http://www.satlib.org>。

测试结果与著名的 WMWALKSAT 算法的结果作比较。在 WMWALKSAT 的第 35 个版本中^[8],一共提供了七种不同翻转变量选择策略,分别为 Random、Productsum、Reciprocal、Additive、Best、Exponential、Tabu。经测试,其中 Best 策略是效果最好的一种。Best 基本思想为从不被满足的子句含有的变量中挑选权重和最小的变量进行翻转,不断接近最优解。实验环境使用 AMD Athlon TM 64 × 2 Dual Core Processor 4400 + 2.30 GHz,1.00 GB 内存。本文对每个问题用并行蚁群算法与串行蚁群算法分别求解 10 次,每次以最大搜索次数(对目标函数的评价次数)为停止准则,然后比较它们求得的解的质量。结果如表 1~3 所示。其中,表 1、2 中#1 表示本文算法,表 3 中加速比 = 串行时间/并行时间,并行效率 = 加速比/2(2 表示为双核 CPU)。对每个测试实例,若子句数超过 1 000 的则设最大搜索次数为 50 000,否则设为 20 000。

表 1 不同变量数的计算结果

序号	变量数	子句数	搜索 次数	目标值	最优值		最优值		最优值		最优值	
					#1	文献[8]	#1	文献[8]	#1	文献[8]	#1	文献[8]
1	40	1 132	50 000	6 394	5 140	5 123	5 113	5 091	5 130	5 106	10.8	11.0
2	43	1 212	50 000	6 904	5 771	5 762	5 771	5 725	5 771	5 752	0.0	14.0
3	60	1 200	50 000	6 826	5 628	5 580	5 575	5 534	5 601	5 558	16.7	16.7
4	64	1 408	50 000	7 966	6 729	6 661	6 682	6 616	6 701	6 636	13.7	16.9
5	70	800	20 000	4 419	4 286	4 272	4 268	4 254	4 275	4 264	6.2	6.5

表 2 不同子句数的计算结果

序号	变量数	子句数	搜索 次数	目标值	最优值		最优值		最优值		最优值	
					#1	文献[8]	#1	文献[8]	#1	文献[8]	#1	文献[8]
1	42	136	20 000	772	675	675	675	671	675	673	0.0	1.6
2	42	400	20 000	2 298	1 894	1 865	1 878	1 791	1 887	1 836	5.9	27.4
3	42	840	20 000	4 712	3 760	3 755	3 742	3 729	3 751	3 741	6.5	9.0
4	42	1 528	50 000	8 602	6 806	6 769	6 787	6 744	6 798	6 754	6.3	8.2
5	42	1 690	50 000	9 440	7 390	7 371	7 368	7 343	7 380	7 354	7.5	8.1

表 3 算法并行化后的计算结果

变量数	子句数	搜索 次数	#1 运行 加速比	#1 并 行效率	最小运行时间/s		平均运行时间/s	
					串行时间	并行时间	串行时间	并行时间
40	1 132	5 000	1.106	0.553	1.014	0.78	1.03	0.931
41	1 640	5 000	1.598	0.799	1.326	0.826	1.339	0.838
42	1 528	5 000	1.675	0.838	1.326	0.795	1.348	0.805
43	1 212	5 000	1.529	0.765	1.045	0.686	1.061	0.694
60	1 200	5 000	1.605	0.803	1.185	0.733	1.202	0.749
64	3 648	5 000	1.825	0.913	2.855	1.606	2.95	1.616
70	700	5 000	1.535	0.767	1.005	0.645	1.01	0.658

表 1、2 中,目标值表示该问题可能取得的最大权重和,其只有在所有子句均成立时方可获得。最优值、最差值、平均值 为利用两算法进行 10 次求解得到的统计值,三者的值越大、越

接近目标值,则表明对应解的质量越高;而标准差表示算法的稳定性,其值越小则算法越稳定。从表 1、2 的结果比较中可以看到,本文算法较 WMWALKSAT 算法在解的质量上有了明显改善,并且取得了较小的标准差,说明本文算法有更好的稳定性。从表 3 可知,并行化后的算法取得了较好的加速比效果及并行效率。

4 结束语

本文提出一种用于多核 CPU 中求解加权 MAX-SAT 的并行蚁群算法,针对加权 MAX-SAT 的特点改进了蚁群算法模型,并使其并行化。改进后的算法对蚁群进化控制更加直接,运行效率也得到大大的提高,同时算法搜索效率也更加高效,算法在正反馈性、内在并行性上都能有效地避免求解问题陷入局部最优的缺点;同时,算法还能更好地控制求解问题的时间和代价,在确保寻找到最优解的前提下极大地提高了算法的收敛速度。实验表明,该算法是有效并切实可行的。

参考文献:

[1] BORCHERS B,FURMAN J. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems[J]. *Journal of Combinatorial Optimization*,1999,2(4):299-306.

[2] De GIVRY S,LARROSA J,MESEGUER P,et al. Solving MAX-SAT as weighted CSP[C]//*Principles and Practice of Constraint Programming*. Berlin:Springer,2003:363-376.

[3] ALSINEL T,MANYA F,PLANES J. An efficient solver for weighted MAX-SAT[J]. *Journal of Combinatorial Optimization*,2008,41(1):61-73.

[4] LIN Han, SU Kai-le. Exploiting inference rules to compute lower bounds for MAX-SAT solving[C]//*Proc of the 20th International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers,2007: 2334-2339.

[5] ELLABIB I,CALAMAI P,BASIR O. Exchange strategies for multiple ant colony system[J]. *Information Sciences*,2007,177(5): 1248-1264

[6] MANFRIN M,BIRATTARI M,STUTZLE T, et al. Parallel ant colony optimization for the traveling salesman problem[C]//*Proc of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence*. Berlin:Springer,2006: 224-234.

[7] 林奋,周育人. 求解可满足问题的改进的蚁群算法[J]. *计算机工程与应用*,2009,45(3):42-44.

[8] Walksat Home Page. Walksat[EB/OL]. <http://www.cs.rochester.edu/~kautz/walksat/>,2009.

(上接第 46 页)方法。仿真实验显示,新算法既能求解到较高精度的解,也有较强的全局搜索能力。

参考文献:

[1] GOLDBERG D E,RICHARDSON J. Genetic algorithms with sharing for multimodal function optimization [M]. Hillsdale: Lawrence Erlbaum,1987.

[2] 于歆杰,王贇基. 对适应值共享遗传算法的分类及评价[J]. *模式识别与人工智能*,2001,14(1):42-47.

[3] ANTONIO D C,CLARDIO D S. On the role of population size and niche radius in fitness sharing[J]. *IEEE Trans on Evolutionary Computation*,2004,8(6):580-592.

[4] 方必和,于蕾蕾. 基于淘汰机制的双种群遗传算法[J]. *计算机技术与发展*,2009,19(9):101-104.

[5] 李军华,黎明,袁丽华. 一种改进的双种群遗传算法[J]. *小型微型*

计算机系统,2008,29(11):2099-2102.

[6] 雷振宇,蒋玉明. 一种基于自主计算的双种群遗传算法[J]. *计算机工程*,2010,36(24):189-191.

[7] TAEJIN P,KWANG R R. A dual-population genetic algorithm for adaptive diversity control[J]. *IEEE Trans on Evolutionary Computation*,2010,14(6):580-592.

[8] TAEJIN P,KWANG R R. A dual-population genetic algorithm with evolving diversity[C]//*Proc of Congress on Evolutionary Computation*. [S. l.]:IEEE Press,2007:3516-3522.

[9] TAEJIN P,CHOE R,KWANG R R. Adjusting population distance for the dual-population genetic algorithm[C]//*Lecture Notes in Computer Science*,vol 4830. Berlin:Springer-Verlag,2007:171-180.

[10] 焦李成,杜海峰,刘芳,等. 免疫优化计算、学习与识别[M]. 北京:科学出版社,2006.