

基于离散 Lagrange 方法的分布式 SAT 问题求解^{*}

唐 屹

(中山大学数学系, 广东 广州 510275 // 广州大学数学系, 广东 广州 510405)

摘 要: 基于对离散 Lagrange 方法 (DIM) 的扩充, 提出一个分布式 SAT 求解算法: EDLMSAT。求解过程中, 单个 Agent 的行为由预先定义的 EDIM 规则所决定, 这些局部的行为聚集起来, 形成整个系统对问题的求解趋势。设计了一些对 3-SAT 基准问题的模拟实验, 实验结果表明了这个算法良好的求解性能。

关键词: 离散 Lagrange 方法; 分布式问题求解; 可满足性问题

中图分类号: TP18 **文献标识码:** A **文章编号:** 0529-6579 (2003) 06-0008-04

对问题进行分布式求解, 是近年来人工智能领域内一个活跃的研究领域。分布式问题求解是将问题的求解工作在多个合作的节点间划分, 创建一个协作的节点群体, 这些群体能够共同工作以完成求解。求解可以在一个分布式多 agent 系统中完成, 而问题则可以为约束满足 (CSP) 问题^[1,2]。基于多 agent 系统的分布式 CSP 问题的求解是通过各个 agent 的行为聚合, 找出问题的变量的一组赋值, 使得这组赋值能够满足所有的约束。

本文讨论一个对 SAT 问题进行分布式处理的算法 EDLMSAT, 这个算法基于扩充的离散 Lagrange 方法 EDLM, 各 agent 依据 EDLM 规则来决定自己的局部行为, 而 Lagrange 乘子作为全局的约束, 影响 agent 的局部行为, 模拟实验表明这提高了求解系统的整体性能。

1 相关的背景和工作

可满足性问题 (SAT) 是一类重要的 CSP 问题。它有多种表示形式, 以命题逻辑的形式可描述如下:

给定 N 个变量 x_j 上的 m 个子句 cl_i , 其中每个 x_j 的值域均为 $\{\text{false}, \text{true}\}$, 每个子句具有形式 $\bigvee_{k \in I_{ik}} l_{ik}$, 文字 l_{ik} 或为变量或为变量的非, 所有子句构成一个合取范式: $\bigwedge_i cl_i$ 。一个 SAT 问题即是寻找一组变量的赋值, 使得上述合取范式为真, 或者确认不可能存在那么一组赋值, 使得合取范式为真。

特别地, 若一类 SAT 问题中的每个子句包含了 3 个文字, 则称这一类问题为 3-SAT 问题。3-SAT 问题具有 NP 完全问题的最简单的表示, 因而

吸引了许多的关注与研究。

局部搜索方法是流行的求解 SAT 问题的方法^[3], 由于对解的搜索是在局部范围内进行的, 容易碰到局部最优, 脱离局部最优的策略构成局部搜索算法的重要组成。离散的 Lagrange 方法 (DIM) 是其中的一种^[4], 基于禁忌搜索, 将传统的 Lagrange 方法用于 SAT 问题的求解, 脱离局部最优的策略是动态子句定权。Lagrange 乘子扮演着子句权重的角色, 当系统碰到局部最优时, 通过修改 Lagrange 乘子来改变子句的权重, 进而改变评估函数, 翻转原来不能翻转的某个变量, 引导系统脱离局部最优。

ERA 系统提供一个模式实现多 agent 合作求解 CSP^[2]。在这个系统中, 每一个 agent 代表着一个变量, agent 始终在一个类似格子的空间中活动, 其大小为这个变量的值域大小, 空间中的每一个位置对应着 agent 所代表变量的一个赋值。求解时, 每一个 agent 会依据其他 agent 的当前位置 (代表着其余变量的赋值) 来计算其活动空间中不同位置的约束违反数量, 以此来决定其要移动的下一个位置, 这隐含着改变变量的赋值。为达到解的状态, agent 可采用的策略为: better move、least move 和 random-move。当系统碰到局部最优时, random move 策略是唯一能够引导系统脱离局部最优的策略。MASSAT 采用 ERA 系统的策略, 实现了对 SAT 问题的分布式求解^[5]。

2 基于 EDLM 的分布式 SAT 求解

以 3-SAT 问题为例, 考虑采用新的 agent 响应

* 收稿日期: 2003-01-03

基金项目: 国家自然科学基金资助项目 (10071097)

作者简介: 唐屹 (1968 年生), 男, 博士, 副教授; E-mail: ytang@gzhu.edu.cn

策略实现对 SAT 问题的分布式求解。不失一般性, 将 N 个变量分成 $n(n \leq N)$ 个分组: $G_1, G_2, \dots, G_n$, $G_i = \{x_{n_{i-1}+1}, x_{n_{i-1}+2}, \dots, x_{n_i}\}$, $n_0 = 0, n_n = N, n_{i-1} < n_i, i \in \{1, 2, \dots, n\}$, 这些分组对应的值域为 $D_1, D_2, \dots, D_n$ 。

定义 1 两个向量 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 与 $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ 的间距为 k ($\text{interval}(\mathbf{x}, \mathbf{y}) = k$) 当且仅当存在一个整数 $k: 1 \leq k \leq n$, 使得 $x_i \neq y_i, 1 \leq i \leq k$ 且 $x_j = y_j, k+1 \leq j \leq n$, 其中 $(i_1, i_2, \dots, i_n)$ 为 $(1, 2, \dots, n)$ 的一个排列。

定义 2 向量 $\mathbf{x}$ 在分组 G_i 上的邻域为: $N_i(\mathbf{x}) = \{\mathbf{y} \mid \text{interval}(\mathbf{x}, \mathbf{y}) = 1 \text{ 且 } x_i \neq y_i, x_i, y_i \in D_i\}$

下面, 考虑与分组 G_i 相关的函数

$$f_i(\mathbf{x}) = \sum_{k=1}^3 \sum_{j=1}^m \Psi_{i,j,k}(\mathbf{x}) \quad (1)$$

其中

$$\Psi_{i,j,k}(\mathbf{x}) = \begin{cases} c_j(\mathbf{x}) & \text{来自包括 } G_i \text{ 在内的 } k \text{ 个分组,} \\ 0 & \text{其它} \end{cases}$$

且 $c_j(\mathbf{x}) = \prod_{l=1}^N a_{j,l}$, 这里

$$a_{j,l} = \begin{cases} (1 - x_l)^2 & \text{若 } x_l \text{ 在 } cl_j \text{ 中,} \\ x_l^2 & \text{若 } \bar{x}_l \text{ 在 } cl_j \text{ 中,} \\ 1 & \text{其他} \end{cases} \quad (2)$$

约定 false 为 0, true 为 1。于是, 若存在一组赋值 $\mathbf{x}$ 使得 $c_j(\mathbf{x}) = 0$, 则该组赋值满足子句 cl_j ; 若同时使 $f_i(\mathbf{x}) = 0$, 则与 G_i 的变量相关的所有子句都被满足, 因此 SAT 问题可被转化为如下的最优问题:

$$\begin{aligned} \min f(\mathbf{x}) &= \sum_{i=1}^n f_i(\mathbf{x}) \\ \text{s.t. } C(\mathbf{x}) &= 0 \end{aligned} \quad (3)$$

这里 $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$,

$$C(\mathbf{x}) = (c_1(\mathbf{x}), c_2(\mathbf{x}), \dots, c_m(\mathbf{x}))^T$$

对应的 Lagrange 函数为

$$L(\mathbf{x}, \lambda) = \sum_{i=1}^n f_i(\mathbf{x}) + \sum_{i=1}^m \lambda_i \circ c_i(\mathbf{x}) \quad (4)$$

定义评估函数为

$$h_i(\mathbf{x}) = \sum_{k=1}^3 \sum_{j=1}^m (k + \lambda_j) \circ \Psi_{i,j,k}(\mathbf{x}) \quad (5)$$

令

$$\partial_i(\mathbf{x}, \lambda) = \mathbf{x} - \mathbf{y}$$

其中, $\mathbf{y} = \arg \min_{\mathbf{x}' \in N_i(\mathbf{x}) \cup \{\mathbf{x}\}} h_i(\mathbf{x}', \lambda)$, 称 $\partial_i(\mathbf{x}, \lambda)$ 为函数(4) 在集合 G_i 上的最陡梯度, 式 (5) 也称为 EDIM 规则。

定义 3 函数 $L(\mathbf{x}, \lambda)$ 的分散的 k 次顺序最陡

梯度记为 $L^k(\mathbf{x}, \lambda)$, 其中 $1 \leq k \leq n$ 。它可由以下的方法计算得到: 选定 $1, 2, \dots, n$ 的一个排列: $i_1, i_2, \dots, i_n$, 按照次序 $j = 1, 2, \dots, n$ 依次计算 $\partial_{i_j}(\mathbf{x}, \lambda)$, 若 k 是第一个这样的 j 使得 $\partial_{i_j}(\mathbf{x}, \lambda) \neq 0$, 则 $L^k(\mathbf{x}, \lambda)$ 的值为 $\partial_{i_j}(\mathbf{x}, \lambda)$; 若不存在这样的 k , 则 $L^k(\mathbf{x}, \lambda) = 0$ 。定义迭代如下:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \partial^k(\mathbf{x}^k, \lambda^k) \quad (6)$$

$$\lambda^{k+1} = \lambda^k + C(\mathbf{x}^k) \quad (7)$$

其中, k 表示第 k 次循环。特别地, 当从 $\lambda \geq 0$ 出发开始迭代时, 有以下定理。

定理 问题 (3) 的一组可满足的赋值可以找到, 如果迭代 (6)、(7) 可结束。

3 求解环境及算法设计

求解环境与 ERA 系统类似, 每个 agent 管理一组变量, agent 在其活动空间的移动代表着变量赋值的改变, 可以用 agent 的移动次数来反映算法的效率。然而 agent 是否发生移动, 依赖于各个 agent 对其局部环境的评估。两种情况使得 agent_{*i*} 保持原有的位置不变, 其一是当前的位置已使得 $h_i(\mathbf{x}) = 0$, 称之为空移(零位置), 其二是尽管当前的位置使得 $h_i(\mathbf{x}) \neq 0$, 但这个位置使得当前状态的 $h_i(\mathbf{x})$ 达到局部最小, 称之为空移(局部最优)。

式 (7) 的非递减的性质, 可能导致某些 λ_j 达到较大的值, 增加求解的难度, 本文采用周期性减少非零 λ 的策略来避免这种现象的出现。其策略如下:

$$\lambda^{k+1} = \lambda^k - \omega(\lambda^k) \quad (8)$$

其中,

$$\omega(\lambda_j) = \begin{cases} 1 & \lambda_j \neq 0 \\ 0 & \lambda_j = 0 \end{cases}$$

算法由 2 个部分构成:

第 1 部分: 主过程:

```
1 counter1=0, counter2=0, inc=1;
2 While not find solution or not time out do
3 For  $a_i \in A$  do
4 compute new—position of  $a_i$  by (6)
5 If current position=new—position then
6 stay;
7 else
8  $a_i$  moves to new—position;
9 Endif;
10 Call update—lambda—procedure
11 If current state is a solution state goto 15;
12 If time out goto 16;
```

```
13 Endfor;
14 Endwhile;
15 Output solution;
16 Return.
第 2 部分: update—lambda—procedure
17 If  $a_i$  encounters agent movement then  $inc = 1$ ;
18 If  $a_i$  encounters non null movement ( zero position)
then
19  $counter_1++$ ;
20 If  $counter_1 \% (\#agent \cdot inc) = 0$  then
21  $counter_2++$ ;
22 If  $counter_2 \% \#agent = 0$  then
23 update  $\lambda$  by ( 8 );
24 Else
25 update  $\lambda$  by ( 7 );
26 Endif
27 If (  $inc < \#agent / 2$  ) then  $inc++$ ;
28 Endif;
29 Endif;
30 Return.
```

在这个算法中, $counter_1$ 被用来计算 agent 移动次数与空移(局部最优)次数的和, $counter_2$ 用来记录 λ 的更新次数, 变量 inc 被用来确定何时更新 λ 。

4 实验结果与分析

实验是一个模拟实验, 分布式 agent 的并行操作是通过顺序指派 agent 来完成, agent 的指派是完全随机的。所处理的问题是文 [6] 中的 $uf100-430$ 基准测试集中的 1000 个可满足的 3 SAT 例子, 每个例子包含 100 个变量, 430 个子句。按变量序号将变量平均分为 33 个分组, 对每个例子均求解 100 次, 每次求解时, x 与 λ 都初始化为 0, 算法标志为 EDIMSAT。

实验 1 与一些局部搜索方法的比较

局部搜索算法间进行比较的主要指标是求解所需的时间和平均步数。常见的局部搜索算法在每一步上是翻转一个变量, 对应于 EDLMSAT, 相当于移动次数。尽管从时间上来说, 两类算法无从相比, 因为 EDLMSAT 基于多 agent 系统, 而局部搜索算法通常在单机上实现, 但仍然可以从平均求解步骤上了解 EDLMSAT 在 SAT 问题求解中的性能。比较结果见表 1, 其他算法的结果来自^[3]。结果显示, EDLMSAT 能够以较少的求解步数找到所测试问题的解。

实验 2 与其他分布式求解算法的比较

在这里比较的是 MASSAT 的 2 个版本的算法^[5], 如表 2 所示, 与已有的 SAT 的分布式求解系

统 MASSAT 相比, 在 agent 的移动次数方面, 我们的算法具有较明显的优势。

表 1 与一些局部搜索算法的比较

Tah 1 Comparison with some local search algorithms

算法 ID	平均求解步数
Novelty (. 6)	2 788
Novelty+ (. 6, . 01)	1 581
DLM (par4)	1 020
SDF (. 00085)	864
EDLMSAT	948

表 2 与 MASSAT 的比

Tah 2 Comparison with MASSAT

算法 ID	移动次数平均值	移动次数中值
MASSAT1	2 950	2 563
MASSAT2	1 462	1 198
EDIMSAT	948	430

5 结 论

基于扩充的离散 Lagrange 方法, 本文提出了一个对 SAT 问题进行分布式求解的算法, 算法可以在一个多 agent 环境中实现。每个 agent 代表着多个变量, agent 的新的评估函数, 使得评估不仅仅与 agent 的局部环境相关, 也考虑了可能的移动对整个系统求解的影响。Lagrange 乘子的引入, 使得各个 agent 从更全局的角度考虑自己的移动, 加快了求解的进程。

参考文献:

[1] YOKOO M, DURFEE E H, ISHIDA T, et al. The distributed constraint satisfaction problem: Formalization and algorithms[J] . IEEE Transaction on Knowledge and Data Engineering, 1998, 10(5): 673— 685.

[2] LIU J, JING H, TANG Y Y. Multi-agent oriented constraint satisfaction[J] . Artificial Intelligence, 2002, 138: 101— 144.

[3] SCHUURMANS D, SOUTHEY F. Local search characteristics of incomplete SAT procedures [J] . Artificial Intelligence, 2001, 132(2): 121— 150.

[4] WU Z. The theory and applications of discrete constrained optimization using Lagrange multipliers[D] . Department Of Computer Science, University of Illinois, 2001.

[5] JIN X, LIU J. Multiagent SAT (MASSAT): autonomous pattern search in constrained domains[A] , In Proceedings of IDEAL02[C] . 2002, LNCS 2412: 318— 328.

[6] <http://www.intellektik.informatik.tu-darmstadt.de/SATLIB/benchm.html>, [EB/OL] 2002. 12. (下转第 18 页)

参考文献:

[1] FOIAS C, SELL G R, TEMAM R. Vanites inertills des equations differentielles dissipatives[J] . C R Acad Sci Paris Ser 1, 1985, 301: 139— 142.

[2] TEMAM R. Infinite-dimensional dynamical systems in mechanics and physics[M] . 2nd. New York: Springer-Verlag, 1997.

[3] FOIAS C, MANLEY O, TEMAM R. Sur l' interaction des petits et grands tourbillon dans des ecoulements turbulents [J] . C R Acad Sci Paris Ser 1, 1987, 305: 497— 500.

[4] 戴正德, 郭柏灵. 惯性流形与近似惯性流形[M] . 北京: 科学出版社, 2000.

[5] 王冠香, 刘曾荣. Kuramoto-Sivashinsky 方程的渐近吸引子 [J] . 应用数学学报, 2000, 23(3) : 329— 336.

[6] BABIN A V, VISHIK M I. Attractors or evolution equations [M] . Moscow: Nauka, 1989.

[7] BARTUCCELLM V, GOURLEY S A, ILYIN A A. Positivity and the attractor dimension in a fourth order reaction diffusion equation[J] . Proc R Soc Lond, 2002, 458: 1431— 1446.

[8] WANG G. On the dynamics of Kuramoto-Sivaskinsky equation[D] . Suzhou: Suzhou University, 1996.

Asymptotic Attractor of Fourth order Reaction diffusion Equation

KUANGXue song

(Department of Basic Courses, Zhanjiang Ocean University, Guangdong, Zhanjiang 524088, China)

Abstract: This article studies the asymptotic attractor of fourth order reaction diffusion equation $u_t + \alpha u_x^4 + u_x^2 + u^3 - u = 0$ by constructing a solution sequence. It is shown that the solution sequence doesn't go away from the global attractor in terms of mathematical induction. Then it is obtained that the solution sequence approaches to the global attractor of the equation in large time, and the dimensional estimate of the asymptotic attractor is obtained.

Key words: fourth order reaction diffusion equation; solution sequence; asymptotic attractor

(上接第 10 页)

Distributed SAT Solving Based on EDLM Method

TANG Yi

(Department of Mathematics, Sun Yat sen University, Guangzhou, 510275, China //
Department of Mathematics, Guangzhou University, Guangzhou, 510405, China)

Abstract: Distributed CSP solving is an active research field in artificial intelligence. Based on the method of extended discrete Lagrange multiplier (EDLM) , a method for distributed SAT solving is proposed. During the SAT solving, each agent will determine its local behavior on some predefined EDLM rules. Although the behaviors are local, the agents can be corporately organized and aggregated to a coherent behavior to a solution. The Lagrange multipliers provide a more global level for agents to consider the SAT solving. Simulated experiments on some 3 SAT benchmark problems are conducted. The results show a competitive performance in solving.

Key words: discrete Lagrange multiplier method; distributed problem solving; SAT