

## 基于 OpenMP 的并行遗传算法求解 SAT 问题

吴贯锋<sup>1,2</sup>, 徐 扬<sup>2,3</sup>, 常文静<sup>1,2</sup>, 陈树伟<sup>2,3</sup>, 徐 鹏<sup>2,3</sup>

(1. 西南交通大学信息科学与技术学院, 四川 成都 610031; 2. 西南交通大学系统可信性自动验证国家地方联合工程实验室, 四川 成都 610031; 3. 西南交通大学数学学院, 四川 成都 610031)

**摘 要:** 为了提高 SAT (boolean satisfiability) 问题求解效率, 在 OpenMP (open multi-processing) 编程框架下, 将遗传算法与局部搜索算法结合, 改进了混合遗传算法中的选择算法, 将原有选择操作的时间复杂度降低到  $O(N)$  级别. 算法采用 OpenMP 中的编译制导语句 `#pragma omp parallel` 粗粒度并行化驱动混合遗传算法, 采用 `#pragma omp single` 语句块实现了子种群间个体的同步迁移操作. 与同类算法 HCGA (hybrid cloud genetic algorithm) 比较分析表明: 改进算法 HGA (hybrid genetic algorithm) 以及并行后的混合遗传算法 CGPHGA (coarse-grained parallel hybrid genetic algorithm) 在求解成功率和求解效率上都有显著提高, 部分问题求解成功率提高达 5 倍.

**关键词:** SAT 问题; OpenMP; 并行混合遗传算法; 粗粒度模型

**中图分类号:** TP311.1 **文献标志码:** A

## Parallel Genetic Algorithm for SAT Problems Based on OpenMP

WU Guanfeng<sup>1,2</sup>, XU Yang<sup>2,3</sup>, CHANG Wenjing<sup>1,2</sup>, CHEN Shuwei<sup>2,3</sup>, XU Peng<sup>2,3</sup>

(1. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China; 2. National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Southwest Jiaotong University, Chengdu 610031, China; 3. School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** To improve solving efficiency for SAT (boolean satisfiability) problems, a combination of genetic algorithm (GA) with local search algorithm (LSA) on the OpenMP (open multi-processing) framework was proposed. This combination improved the selection algorithm in the hybrid genetic algorithm (HGA) and reduced the time complexity of the original selection operation to  $O(N)$ . The compiler guide statement `#pragma omp parallel` in OpenMP was applied to coarse-grained parallelization driven HGA, and the `#pragma omp single` statement block was used to implement the synchronization migration operation of the individuals in different sub-groups. Compared with the similar algorithm, HCGA (hybrid cloud genetic algorithm), both the improved algorithm (HGA) and the coarse-grained parallel hybrid genetic algorithm (CGPHGA) significantly improved solution success rate and problem solving efficiency. Solution success rate for some problems was increased by 5 times.

**Key words:** SAT problem; OpenMP; parallel hybrid genetic algorithm; coarse-grained model

SAT (boolean satisfiability) 问题是逻辑学的一个基本问题, 是第一个被证明的 NP 完全问题<sup>[1]</sup>, 也是当今计算机科学和人工智能研究的核心问题. 工

程技术、军事、工商管理、交通运输及自然科学研究中的许多重要问题, 如程控电话的自动交换、大型数据库的维护、大规模集成电路的自动布线、软件自动

收稿日期: 2017-09-20 修回日期: 2018-04-04 网络首发日期: 2018-05-30

基金项目: 国家自然科学基金资助项目(61673320); 中央高校基本科研业务费专项资金资助项目(2682017ZT12)

作者简介: 吴贯锋(1986—), 男, 博士研究生, 研究方向为智能信息处理、自动推理与并行计算, E-mail: w1520gx@gmail.com

引文格式: 吴贯锋, 徐扬, 常文静, 等. 基于 OpenMP 的并行遗传算法求解 SAT 问题[J]. 西南交通大学学报, 2019, 54(2): 428-435.

WU Guanfeng, XU Yang, CHANG Wenjing, et al. Parallel genetic algorithm for SAT problems based on OpenMP[J]. Journal of Southwest Jiaotong University, 2019, 54(2): 428-435.

开发、机器人动作规划等,都可转化成 SAT 问题.因此致力于寻找求解 SAT 问题的快速而有效的算法,不仅在理论研究上而且在许多应用领域都具有极其重要的意义.在某些应用领域,不完备的启发式搜索算法则具有较高的效率.

遗传算法属于启发式搜索算法的一种,然而单纯地将遗传算法应用到 SAT 问题求解,必然会遇到效率瓶颈,简单遗传算法中的比例选择操作已经被证明是不收敛的,而使用保存最佳个体策略进行选择遗传算法一定能收敛到全局最优解.文献[2]用三路快速排序和局部搜索算法对遗传算法进行改进;文献[3]提出多智能体社会进化算法求解 SAT 问题;文献[4]研究了随机局部搜索的双重配置检查方法,这些研究在一定的程度上与一般的遗传算法相比,提高了求解速度、成功率和求解问题的规模.将遗传算法和其他算法相结合的混合遗传算法求解 SAT 问题是遗传算法在 SAT 问题上的研究趋势<sup>[5]</sup>.本文对混合遗传算法求解 SAT 问题的算法进行改进,并使用 OpenMP (open multi-processing) 将其并行化,通过对比改进算法和并行化算法都有取得显著效果.

## 1 背景

### 1.1 SAT 问题的一般描述

SAT 问题一般用合取范式 (conjunction normal form, CNF) 表示.假设有  $n$  个布尔变量  $x_1, x_2, \dots, x_n$ , 取值为 0 或者 1. 定义文字  $l$  要么是  $x_i$  (正文字) 要么是  $\neg x_i$  (负文字),  $i = 1, 2, \dots, n$ , 一个子句  $C$  是若干文字的析取, 一个子句集  $S$  是若干个子句的合取.

求解 SAT 问题就是要找到  $x_1, x_2, \dots, x_n$  的一组赋值,使得 CNF 子句集中的每个子句在当前赋值条件下为真.当且仅当所有赋值组合都不能使  $S$  为真,则称子句集  $S$  不可满足.

### 1.2 遗传算法

遗传算法 (genetic algorithm, GA) 是由美国 Michigan 大学的 Holland 教授于 1975 年提出,后经 DeJong、Goldberg 等归纳总结所形成的一类模拟进化算法<sup>[6-8]</sup>.

经典遗传算法的基本流程描述如下所示:

- (1) 初始化迭代次数  $t$  和种群  $P(t)$ ;
- (2) 评估  $P(t)$  中的个体适应度;
- (3) 判断是否达到终止条件,达到则跳转到步骤(8);

(4) 从上一代种群中选择个体进行加入到种群  $P_1$  中(有比例选择、遍历随机抽样、锦标赛选择等选择算法<sup>[9]</sup>);

(5) 对  $P_1$  中的个体进行交叉操作得到种群  $P_2$ ;

(6) 对  $P_2$  中的个体进行变异操作得到种群  $P(t+1)$ ;

(7)  $t = t + 1$  跳转到步骤(2);

(8) 输出最优个体以及对应的函数值.

### 1.2.1 并行遗传算法

遗传算法因其本身的特性,个体之间的相互独立,个体之间的适应度可并行计算,划分的不同种群之间也可相互独立的进化演变.并行遗传算法充分发挥了并行计算机的性能,加速了搜索过程,极大地促进了遗传算法的研究与应用.优于并行节点独立运行,并行遗传算法扩大了种群规模,隔离子种群,保持了种群的多样性,能够有效防止过早收敛,提高了解的质量<sup>[10]</sup>.

利用并行遗传算法求解大规模工程应用问题取得不错的成绩,例如:文献[11]将其应用到军用无人机路径规划问题;文献[12]利用其解决无线网络通信优化问题;文献[13]在电力机组组合优化问题上使用并行遗传算法,提出双向环形拓扑结构在组合优化问题上有较好的效果.

### 1.2.2 并行遗传算法模型

并行遗传算法的模型主要有 4 种:主从式模型、粗粒度模型、细粒度模型以及混合式模型<sup>[10]</sup>.

主从模式是最简单的并行遗传算法模型,主节点负责遗传操作,从节点只并行的计算适应度函数.并行节点的计算能力主要服务于适应度计算,因此该模式适用于计算量集中在适应度评价上的小种群问题.

粗粒度模型应用最为广泛,各个并行节点中的个体作为一个种群独立进化,经过一定代数后,各个并行节点中的种群交换优秀个体,从而提高种群的多样性,防止过早收敛.粗粒度模型通信开销小,而且配置灵活,扩展方便,应用广泛.本文就采用此模型来设计求解 SAT 问题的并行遗传算法.

细粒度模型又称为邻域模型,与粗粒度模型结构类似,但是子种群划分结构细小,甚至到每个个体占用一个并行节点.交叉选择只能在并行节点间进行,对带宽要求较高而且要求并行节点数量较多.

混合模型就是把以上 3 种模型混合使用,通过一些策略将 3 种模型混合,形成一定的层次结构,不

同层次采用不同的模型. 混合模型结构复杂, 同时又灵活多变, 因此也可以达到很好的并行效果.

### 1.3 OpenMP

OpenMP 是一个能够为共享内存计算机创建并行程序的编译制导指令和库函数的集合, 它基于派生和聚合(fork/join)的编程模型. 程序开始执行时只是一个单线程程序, 当程序需要并行执行的时候就会派生出另外的一些线程, 形成一个线程组, 多个线程并行执行并行区域的代码; 在并行区域的最后, 线程将等待知道线程组中所有线程都执行到该位置, 然后线程聚合在一起; 聚合后, 回归主线程继续执行或结束<sup>[14]</sup>. OpenMP 降低了并行程序设计的难度, 开发人员不必关注并行设计的详细内容, 仅仅围绕业务逻辑, 大大提升并行软件开发的效率.

## 2 相关工作

### 2.1 遗传算法求解 SAT 问题的一般用法

GA 在优化问题中广泛应用, 那么将其应用在 SAT 问题求解上也一定具有很大的潜力. 利用 GA 求解 SAT 问题, 它是将 SAT 问题转化为一个优化问题. 如何定义适应度函数  $f$  是一个关键问题, 简单的定义,  $f$  是真值指派使公式取得的真值, 其值域为  $\{0, 1\}$ ; 也有文献定义为真值指派所不能满足的子句数或所不能满足的子句权重之和.

如果 SAT 问题是可满足的, 存在一组由 0 和 1 构成的赋值使得所有子句取值为 1. 可以采用二进制串作为染色体编码方法, 这种编码方案充分利用了 SAT 本身的特点, 便于计算适应度函数和设计各种遗传操作.

对于由  $n$  个变元的构成的 SAT 问题, 染色体用  $n$  位二进制串表示, 它与由真值指派中变元的取值直接对应, 可直接解释成真值指派.

### 2.2 并行求解 SAT 问题

于此同时, 求解 SAT 问题的趋势则是伴随着计算机硬件的升级、集群计算、大数据、云计算等技术日益成熟, 采用并行化的方法求解 SAT 问题的研究一直都没有停止过.

并行的 SAT 求解器主要基于两种方式: 分治方法和组合并行方法. 其中分治方法主要是利用变量的 0 或者 1 取值, 将搜索树划分为包含变量正负两个文字的子树, 然后并行节点只在子树中进行搜索计算<sup>[15]</sup>. 另外一种方式是组合并行, 该并行方式组合了多个不同的求解器(或者是同一个求解器, 但是不同节点分配不同的参数), 在并行节点上针对同一个

问题运行不同的求解器, 各计算节点之间可以交流学习子句, 进一步简化搜索空间<sup>[16-17]</sup>. 本文提出的基于 OpenMP 的并行遗传算法求解 SAT 问题, 本质上属于组合并行, 并行节点中的任何一个种群中的任何一个个体进化到最优状态(个体取值使得所有子句可满足), 程序即可运行结束. 个体的解, 即是全局的解.

### 2.3 迁移策略

粗粒度模型的实现方法有很多种, 可以将子群体中的最优个体向外迁移<sup>[12-13]</sup>; 也可以随机选出迁移个体<sup>[18]</sup>. 有的算法用迁入者取代群体中的最差个体, 利用最优个体迁出, 同时取代最差个体被是最常用的方法<sup>[19]</sup>.

子种群迁移方式上又可分为同步迁移和异步迁移两种.

同步迁移, 又叫做确定间隔迁移, 就是个体迁移的代数间隔是固定的. 迁移时, 要保证各个子种群所在的计算节点进化的代数相同, 利用消息传递机制, 同步进行. 实验结果表明, 间隔过大, 会未成熟收敛(收敛到次优解), 虽然收敛快, 但解的质量不高; 间隔过小, 使子群体之间的多样性被破坏, 在可接受的时间内依然得不到高质量的解. 本文采用同步迁移方式, 迁移的代数间隔要具体问题具体分析.

异步迁移方式又叫做非确定性迁移方式, 一个子群发展进入平稳期的时候, 突然来了外部个体, 能够带动子种群快速进化. 异步迁移方法就是根据这个现象提出的. 当一个子群的发展(经过一定的进化代)一直停滞不前, 则向其它子群体发出申请——申请个体迁入; 而当子群体的最佳个体的适应度提高后, 向其邻居子群体发送这个最佳个体. 子种群之间互相协同, 共同进化, 也等同于有个公共的最佳个体资源池, 每个子群在发展停滞时去资源池获取优秀个体, 自己进化最优时贡献优秀个体<sup>[20]</sup>.

## 3 混合并行遗传算法

### 3.1 局部搜索算法

本文将遗传算法和局部搜索算法结合求解 SAT 问题, 局部搜索是解决优化问题的启发式算法. 且局部搜索算法是以时间换精度的近似算法, 通过改变当前解的某一部分而产生当前解的邻域解, 判断邻域解的质量. LSA (local search algorithm) 算法复杂度是基于其线性阶的复杂度, 可以利用多长迭代来求解问题<sup>[21-22]</sup>.

应用于 SAT 问题,可设  $n$  表示子句集中的  $n$  个变元,  $X=\{x_1, x_2, \dots, x_n\}$  为一组赋值,  $S$  表示一个解的集合, 适应度函数  $f(X)$  的值为可满足子句数目,  $\text{TurnOver}(X, i)$  表示对变量  $x_i$  取反. 则 LSA 算法可用伪代码描述如下:

- (1) 随机产生一组赋值  $X$ ;
- (2) 计算  $X$  的适应度  $f(X)$ ;
- (3)  $X$  赋值给  $X_1$ ;
- (4) for (下标  $i$  从 1 到  $n$ ) 翻转  $X$  的第  $i$  个基因座, 得到  $X'$ , 若  $f(X') > f(X_1)$  则将  $X'$  赋值给  $X_1$ ;
- (5) If (达到终止条件) Print( $X_1$ ).

### 3.2 改进的混合遗传算法

遗传算法与局部搜索算法的结合能够加速收敛获得最优个体, 且 LSA 算法是线性的, 不会增加太多的额外开销, 而且在求解最优解质量和计算速度方面要优于同类算法. 改进的混合遗传算法 (hybrid genetic algorithm, HGA) 对选择操作做了优化改进, 有效降低选择算法的时间复杂度.

#### 3.2.1 对选择操作的改进

在遗传算法的选择过程中, 保留优秀个体, 淘汰较差的个体, 隐含一个排序的操作. 虽然文献[2]中与三路快速排序相结合提高了效率, 但实际上只需要知道较好的个体是哪些, 较差的个体是哪些即可, 则排序的问题可等价转换为求数组中较大的前  $K$  个数的问题.

对此类问题, 存在时间复杂度为  $O(N)$  的算法, 可以作为选择操作算法. 在本文算法中, 假设函数  $p(\cdot)$  是快速排序中的划分函数, 实现功能是将数组  $A$  中从索引  $l$  到索引  $h$  的元素按某个中位数 (一般取数组的第 1 个元素) 划分为大于该中位数和小于该中位数的两部分, 用  $q$  表示该中位数索引,  $u$  表示中位数是索引  $l$  后面的第几个数, 输入条件  $k$  表示要寻找数组中较大的前  $k$  个数,  $s$  表示算法执行结束后, 最后一个较大的数的索引. 则复杂度为  $O(N)$  算法 FKM (FindKMax) 伪代码描述如下:

- ① FKM( $A, l, h, k$ )
- ② {
- ③   if ( $l < h$ )
- ④   {
- ⑤      $q \leftarrow p(A, l, h)$ ;
- ⑥      $u \leftarrow q - l + 1$ ;
- ⑦     if ( $u = k$ )
- ⑧          $s \leftarrow q$ ;

- ⑨     else if ( $u < k$ )
- ⑩      $s \leftarrow \text{FKM}(A, q + 1, h, k - l)$ ;
- ⑪     else
- ⑫          $s \leftarrow \text{FKM}(A, l, q - 1, k)$ ;
- ⑬     }
- ⑭     return  $s$ ;
- ⑮ }

#### 3.2.2 HGA 算法流程

HGA 主要是将标准 GA 和 LSA 相结合, 充分利用 LSA 的局部搜索能力和 GA 的全局搜索能力, 有效弥补了 GA 在局部搜索能力较弱的问题. HGA 保持 GA 的基本框架, 在产生新一代个体后加入多步长的局部搜索 (步长由输入参数确定). 算法流程描述如下:

- (1) 随机生成一组大小为  $g$  的个体;
- (2) While (未到终止条件) Do;
- (3) 调用 Fitness 函数计算每个个体的适应度值;
- (4) 如果有最优个体 (使所有子句可满足) 则终止算法;
- (5) 如果连续多少代之后未产生优秀个体, 判断当前陷入了局部最优, 则进行灾变操作杀死当前所有个体, 随机产生新个体;
- (6) 调用 FKM 算法选出一半适应度较高的个体, 淘汰掉另一半个体;
- (7) 按照采用轮盘赌算法选择个体按照设定的交叉变异概率进行交叉和变异产生子代个体 (为了保持种群规模将会多次调用选择交叉变异操作);
- (8) 对新生代群体中的一部分调用多步长 LSA 算法, 寻找其邻域解;
- (9) 结束 While 循环.

假设输入为标准 3SAT 问题的子句集, 含有  $m$  条子句,  $n$  个变元, 遗传算法种群大小为  $g$ , 进化终止代数为  $t$ , 每个个体进化后, 对种群中的一半个体执行的 LSA 算法, 且步长为  $x$ . 则算法时间复杂度为关于变量  $g, x, m, n, t$  的函数. 选择操作时间复杂度为  $O(g)$ , 变异操作时间复杂度计为 1, 则改进后的混合遗传算法的时间复杂度如式(1)所示.

$$T(m, n, g, t, x) = gn + t \left( 3mg + g + \frac{gn}{2} + 1 + \frac{3mgnx}{2} \right) = O(mgntx + mgt + gnt + gt + gn + t). \quad (1)$$

### 3.3 粗粒度并行遗传算法 CGPHGA 描述

采用 OpenMP 设计粗粒度并行遗传算法需要解

决的迁移策略和数据共享问题. OpenMP 默认在 `#pragma omp parallel` 语句块外面的变量为线程间共享变量, 也可采用 `share` 制导语句设置共享变量, 不必担心因数据共享导致的线程死锁问题.

但是需要注意的是多个线程都能修改共享变量, 对于有线程依赖关系的共享变量, 需要额外控制, 防止因数据依赖导致运行结果不可控. 基于 OpenMP 的 CGPHGA (coarse-grained parallel hybrid genetic algorithm) 算法中, 设置一个全局共享的二维向量 `ShareBest` 作为各线程迁移个体使用的缓冲区, 第 1 维度为线程编号, 第 2 维度为需要迁移的个体集合, 如果各 OpenMP 子线程到达迁移代数 (`currentIteration%Interval==0`), 则设置 OpenMP 线程同步制导语句 `#pragma omp barrier` 使得线程同步, 在两个 `barrier` 之间, 使用 `#pragma omp single` 语句块让任意空闲线程执行迁移操作. 假设有  $N$  个线程, 则迁移操作的拓扑结构如图 1 所示.

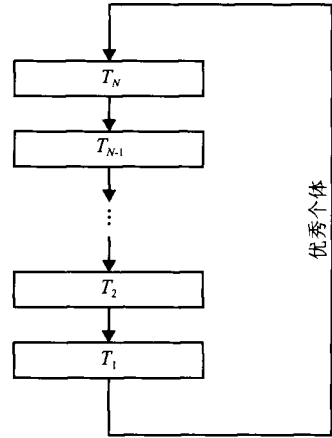


图 1 迁移拓扑  
Fig. 1 Topology of the migration

迁移操作完成后, 各个子线程获取从邻域线程中的优秀个体替换较差的个体, 然后继续迭代进化. 任何一个线程寻找到最优解后输出结果, 结束整个算法运行. 其主要算法流程如图 2 所示.

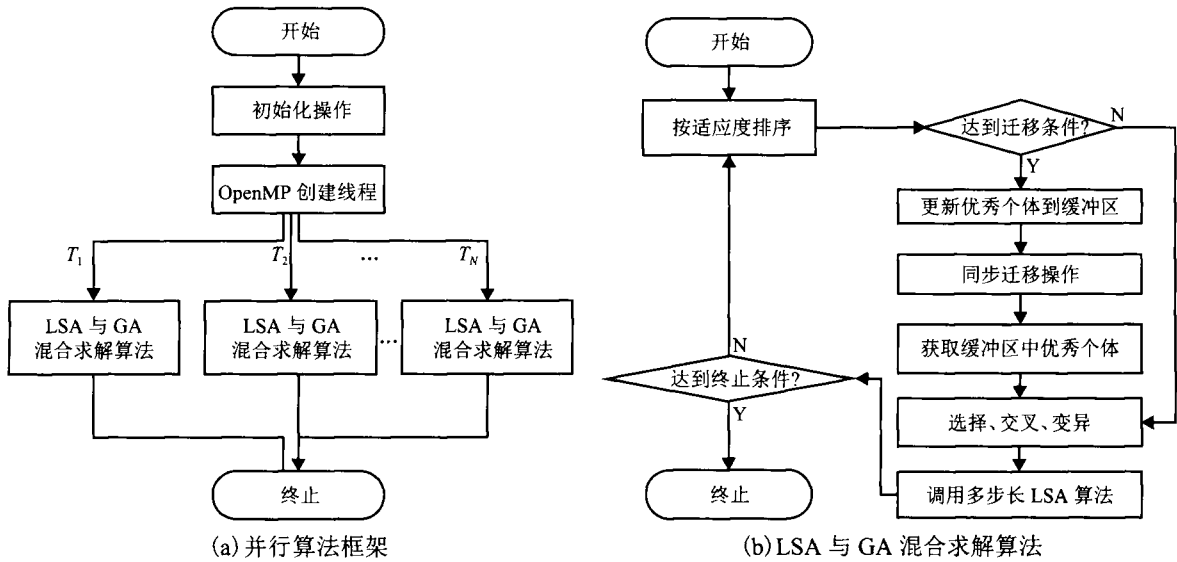


图 2 CGPHGA 算法执行流程  
Fig. 2 Flow of CGPHGA

### 4 实验及分析

实验数据均来自 SATLib 库中的基准例. 文献[2]中已经对混合遗传算法和标准遗传算法的效率进行了对比, 本实验对改进的 HGA 和 CGPHGA 与文献[2]中的算法 HCGA (hybrid cloud genetic algorithm) 对比. 随机选择 100 个子句集作为一组, 每组运行 10 次, 每个实例限时 200 s, 计算每组平均求解个数作为求解成功率  $S$ . 本文所用实验电脑为 HP 8300 一体机, 具体参数对比如表 1 所示.

表 1 实验环境对比

Tab. 1 Experimental conditions

环境	操作系统	CPU	内存/GB	限时
本文	Win10 x64	i3-3240	4	200 s
对比	Win7 x64	i5-3470	4	23 h

实验中改进的混合遗传算法和并行遗传算法求解每个 SAT 问题均限时 200 s, 交叉率 0.7, 变异率 0.1, 灾变间隔 10 (表示连续 10 代不产生最优个体执行灾变操作), 其他运行参数信息如表 2 所示.

表 2 改进混合遗传算法和并行遗传算法运行参数信息  
Tab. 2 Parameters of HGA and CGPHGA

变元	子句	算法	种群大小	LSA 步长	并行数	迁移间隔	迁移率
50	218	HGA	100	10	4	8	0.10
		CGPHGA	25	10			
75	325	HGA	100	10	4	10	0.08
		CGPHGA	25	10			
100	430	HGA	100	10	4	8	0.10
		CGPHGA	50	10			
125	538	HGA	150	15	4	10	0.10
		CGPHGA	40	15			

图 3 展示了改进的混合遗传算法和并行化之后的混合遗传算法与文献中的算法在求解成功率上的对比结果. 其中  $L$  表示染色体长度(也是 SAT 问题的变元个数),  $S$  表示百个问题的求解成功率. 其中红方框标注曲线代表对比算法 HCGA, 蓝色三角标注为改进的混合遗传算法, 棕色五角星标注为基于 OpenMP 将 HGAG 并行化后的并行算法 CGPHGA, 从图中可以看到, 改进的混合算法和并行算法, 在实验主机对比主机硬件配置差的情况下依然有较高的求解成功率. 特别是在变元个数为 50 时, 求解成功率为对比算法的 5 倍.

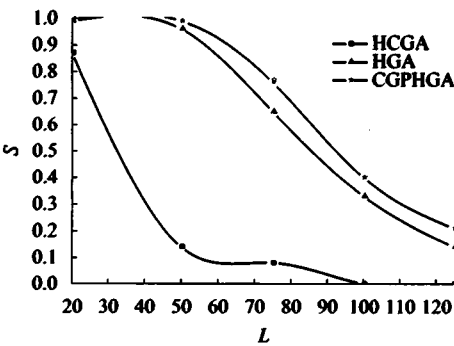


图 3 HCGA、HGA 和 CGPHGA 求解成功率比较  
Fig. 3 Success rates of HCGA, HGA and CGPHGA

计算每个算法对 100 个实例中求解出的实例平均时间, 对比算法直接采用文献中的实验数据. 改进算法在求解时间上明显优于对比算法. 各算法求解平均时间对比如表 3 所示.

依然以变元个数为 75 的 CNF 公式集为例, HGA 与 CGPHGA 求解最优解时所进化代数的分布频率如图 4 所示. 蓝色实线代表改进的混合遗传算法 HGA, 红色虚线为基于 OpenMP 的并行化算法 CPGHGA. 从图中可以看到, HGA 在 5~10 代内获取最优解的个数最多, 而 CGPHGA 则相对分布较为平缓, 并行

节点中每个节点所设定的子种群大小较小, 计算量相对较少, 能够在有限时间内进化代数增大, 这也是 CGPHGA 优于 HGA 的原因之一.

表 3 HCGA、HGA 和 CGPHGA 平均时间对比  
Tab. 3 Average time required by HCGA, HGA and CGPHGA

变元	子句	HCGA	HGA 平均时间/s	CGPHGA 平均时间/s
50	218	38.089	23.451	22.779
75	325	128.754	65.064	61.132
100	430	85 714.200	93.930	93.492
125	538	0	158.960	122.74

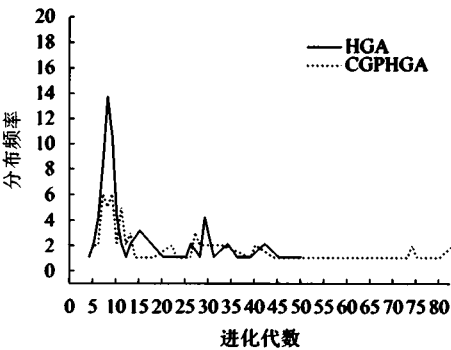


图 4 HGA 和 CGPHGA 进化代数分布  
Fig. 4 Distribution graph of evolution of HGA and CGPHGA

固定子种群数量时, 迁移规模对求解成功率的影响如图 5 所示. 图中横坐标为迁移个体占子种群的比例, 纵坐标为当前设置下求解成功率,  $L = 75$ , 100, 125 这 3 条曲线分别表示变元个数为 75、100 和 125 时对应的求解成功率. 从图 5 中可以得出, 固定子群数量时, 增加迁移个体能够有效提高子群的多样性, 提高求解成功率. 迁移太少, 导致子群不能有效利用其它子群的信息, 但是过多地迁移个体则会破坏种群多样性, 同样不利于寻找最优解. 图 5 中

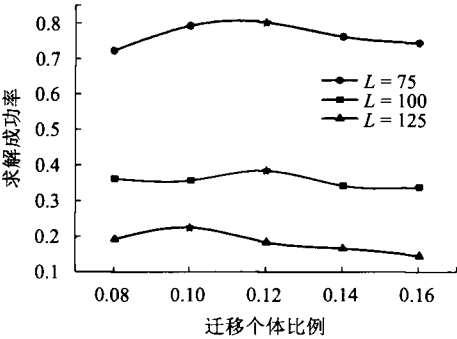


图 5 迁移规模对求解成功率的影响

Fig. 5 Effect of volume of migration on the success rate

标星的点代表迁移比例在该点处达到相对最优。

在并行效率对比实验中主要和同类算法中的

PGSAT (parallel greedy sat algorithm)<sup>[23]</sup>以及完备的 PDPLL (parallel davis-putnam-logemann-loveland algorithm)<sup>[24]</sup>算法做对比,每组实例运行 10 次,限时 200 s,每个实例均采用 4 个线程并行.其中 CGPHGA 参数设置与表 2 保持一致,PGSAT 中尝试次数 2 000 次,Moves 参数设置为 10.实验结果如表 4 所示,其中 Time-avg 为求解出的问题平均用时.

PGSAT 是并行贪心算法,在限制时间内,只有一组实例能够求解. PDPLL 算法属于并行完备算法,其本质是在做遍历操作.当变元个数增加,计算资源有限的情况下,搜索空间几何增长,取变元数为 125 和 150 时,PDPLL 比本文算法效果要好.后续,变元个数增加时,比本文算法运行效果差.

表 4 CGPHGA 与其他并行算法对比情况

Tab. 4 Comparison of CGPHGA with the other parallel algorithms

变元	子句	PGSAT		PDPLL		CGPHGA	
		S	平均时间/s	S	平均时间/s	S	平均时间/s
125	538	0.10	186.400	0.41	101.206	0.21	122.541
150	645			0.14	152.957	0.10	173.612
175	753			0.02	199.298	0.05	196.793
225	960					0.01	198.570

5 结 论

本文提出使用 OpenMP 实现粗粒度并行混合遗传算法,给出了详细的算法设计思路,通过实验验证,改进的混合算法 HGA 明显优于对比算法,且基于 OpenMP 的并行化后的算法也取得较好的效果,但是局限于实验环境和实验时间关系,实验结果中并行的加速效果并不符合阿姆达尔定律,需要在更好的硬件上做更多的实验分析其中原因.

新的并行编程框架如 Spark、Map reduce 等在解决并行优化问题上都有相关的研究,推动了并行遗传算法在解决优化问题的发展.更深入的研究也可在这些方面进行.

致谢:感谢中国无线电协会对项目的资助(T/RAC 015-2016).

参考文献:

[1] 黄拙,张健. 由一阶逻辑公式得到命题逻辑可满足性问题实例[J]. 软件学报,2005,16(3): 329-325.

HUANG Zhuo, ZHANG Jian. Generating SAT instances from first-order formulas[J]. Journal of Software,

2005, 16(3): 329-325.

[2] LI Bingfen, ZHANG Yu'an. A hybrid genetic algorithm to solve 3-SAT problem.[C]//2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. Changsha: IEEE Press, 2016: 476-480.

[3] 潘晓英,焦李成,刘芳. 求解 SAT 问题的多智能体社会进化算法[J]. 计算机学报,2014,37(9): 2011-2020.

PAN Xiaoying, JIAO Licheng, LIU Fang. A multi-agent social evolutionary algorithm for SAT problem[J]. Chinese Journal of Computers, 2014, 37(9): 2011-2020.

[4] LUO Chuan, CAI Shaowei, WU Wei, et al. Double configuration checking in stochastic local search for satisfiability[C]//Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. Québec City: AAAI, 2014: 2703-2709.

[5] 郭莹,张长胜,张斌. 求解 SAT 问题的算法的研究进展[J]. 计算机科学,2016,43(3): 8-17.

GUO Ying, ZHANG Changsheng, ZHANG Bin. Research advance of SAT solving algorithm[J]. Computer Science, 2016, 43(3): 8-17.

[6] HOLLAND J H. Adaptation in natural and artificial

- systems[M]. Cambridge: MIT press, 1992: 159-171.
- [7] DE JONG K A. The analysis of the behavior of a class of genetic adaptive systems[D]. Ann Arbor: University of Michigan, 1975.
- [8] GOLDBERG D E. Genetic algorithms in search optimization and machine learning[J]. Machine Learning, 1988, 3(2): 95-99.
- [9] 张琛, 詹志辉. 遗传算法选择策略比较[J]. 计算机工程与设计, 2009, 30(23): 5471-5474, 5478.  
ZHANG Chen, ZHAN Zhihui. Comparisons of selection strategy in genetic algorithm[J]. Computer Engineering and Design, 2009, 30(23): 5471-5474, 5478.
- [10] 高家全, 何桂霞. 并行遗传算法研究综述[J]. 浙江工业大学学报, 2007(1): 56-59, 72.  
GAO Jiaquan, HE Guixia. A review of parallel genetic algorithms[J]. Journal of Zhejiang University of Technology, 2007(1): 56-59, 72.
- [11] ROBERGE V, TARBOUCHI M, LABONTE G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning[J]. IEEE Trans. Industrial Informatics, 2013, 9(1): 132-141.
- [12] CALEGARI P, GUIDEC F, KUONEN P, et al. Parallel island-based genetic algorithm for radio network design[J]. J. Parallel Distrib. Comput, 1997, 47(1): 86-90.
- [13] YANG Hongtai, YANG Paichuan, HUANG Chinglien. A parallel genetic algorithm approach to solving the unit commitment problem: Implementation on the transputer networks[J]. IEEE Transactions on Power Systems, 1997, 12(2): 661-668.
- [14] TIMOTHY G. MATTSON B S, BERNA M. Patterns for parallel programming[M]. [S.l.]: Pearson Education, 2004: 76-78.
- [15] ASGHAR S, AUBANEL E, BREMNER D. A dynamic moldable job scheduling based parallel SAT solver[C]//2013 42nd International Conference on Parallel Processing (ICPP). [S.l.]: IEEE, 2013: 110-119.
- [16] HAMADI Y, JABBOUR S, SAIS L. ManySAT: a parallel SAT solver[J]. Journal on Satisfiability, Boolean Modeling and Computation, 2008, 1(6): 245-262.
- [17] WU Guanfeng, XU Yang, CHANG Wenjing, et al. Parallel genetic algorithm for SAT problems based on the coarse-grained model[C]//Uncertainty Modelling in Knowledge Engineering and Decision Making: Proceedings of the 12th International FLINS Conference. Roubaix: Springer, 2016: 489-495.
- [18] MASTSUMURA T, NAKAMURA M, OKENCH J, et al. A parallel and distributed genetic algorithm on loosely-coupled multiprocessor systems[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 1998, 81(4): 540-546.
- [19] BECKERSM L M, DERKS EPPA, MELSSSEN W J, et al. Using genetic algorithms for conformational analysis of biomacromolecules[J]. Computers & Chemistry, 1996, 20(4): 449-457.
- [20] 岳嵌. 粗粒度并行遗传算法的计算性能及其应用研究[D]. 武汉: 华中科技大学, 2008.
- [21] MENOUEUR T, BAARIR S. Parallel satisfiability solver based on hybrid partitioning method[C]//2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing. [S.l.]: IEEE, 2017: 54-60.
- [22] LAYEB A, SAIDOUNI D E. A hybrid quantum genetic algorithm and local search based DPLL for max 3-SAT problems[J]. Applied Mathematics & Information Sciences, 2014, 8(1): 77-87.
- [23] ROLI, A. Criticality and parallelism in structured SAT instances[C]//International Conference on Principles and Practice of Constraint Programming. Berlin: Springer, 2002, 714-719.
- [24] ZHANG Wenhui, HUANG Zhuo, ZHANG Jian. Parallel execution of stochastic search procedures on reduced SAT instances[C]//Pacific Rim International Conference on Artificial Intelligence. Berlin: Springer, 2002: 108-117.

(编辑: 唐 晴)