无向图的边极大匹配并行算法及其应用*

马 军[1] 岩间一雄[2] 顾谦平[3]

[1](山东大学计算机科学系 济南 250100)

[2](京都大学计算机科学系 日本京都市)

[3](会津大学软件系 日本若松市)

E-mail: majun@sdu.edu.cn

摘要 在EREW PRAM(exclusive-read and exclusive-write parallel random access machine)并行计算模型上,对范围很广的一类无向图的边极大匹配问题,给出时间复杂性为$O(\log n)$,使用$O((n+m)/\log n)$处理器的最佳、高速并行算法.

关键词 并行图算法,边极大匹配.

中图法分类号 TP301

设$G(V,E)$为无向图,$V=\{1,2,\ldots,n\}$为G的顶点集,$\forall v\in V$,v表示顶点标号.E为G的边集,$n=|V|$,$m=|E|$.子集$M\subseteq E$被称为G的边匹配,若$\forall e_1,e_2\in M$,e_1与e_2无共同顶点.若M不被G的任何边匹配所真包含,则M被称为G的极大边匹配MM(maximal matching).MM的并行求解算法已成为许多应用问题并行求解的基础算法.[1,2]目前,在EREW PRAM(exclusive-read and exclusive-write parallel random access machine)的并行计算模型上,对MM的费用最好的算法解为使用$O(n+m)$个处理机,时间复杂性为$O(\log^4 n)$的并行算法.[2]本文中,在范围很广的一类图集合上,提出对MM的新并行算法,该算法的运算步数比此前的最好算法在该图集合上减少$O(\log n)$因子,为在该图集合上的最佳算法.

1 基本术语

设边集$E=F_1\cup\ldots\cup F_k$,F_i为森林且当$i\neq j$,$F_i\cap F_j=\varnothing$.称所有这样的边分割中,具有最少森林数k的分割为G的裁减(Arboricity).其最少森林数目记为$a(G)$.设$\Pi=\{G\,|\,G$为无向图且$a(G)=O(1)\}$,则已知Π含有平面图、种类(Genus)受限图及最大顶点度数受限图等.[3,4]设$A\subseteq V$,$\Gamma(A)=\{x\,|\,(x,v)\in E,v\in A$且$x\notin A\}$为顶点集合$A$的邻域;$T_i=(V_i,E_i,r_i)$为$G$的一棵有向根树,满足$V_i\subseteq V$,$E_i\subseteq E$,$r_i$为$T_i$的根.$depth(v)$被定义为顶点$v$到$r_i$路径上的边数.定义$depth(r_i)=0$.$F$被称为$G$的一个有向林,若$F$由$G$的$k(>1)$棵有向根树$T_1\cup\ldots\cup T_k$组成,满足:① $E(T_i)\cap E(T_j)=\varnothing$,$i\neq j$;② $V(T_1)\cup\ldots\cup V(T_k)=V(G)$.$F$可由一维数组$F(1..n)$表示,即$F(i)=j$,当且仅当在子树$T_k$上,$j$为$i$的父结点.定义$F(r_i)=r_i$.

定理1.[3] 设$W(n)$为在PRAM模型上,在$O(1)$时间内可并行完成的操作步数,则在有p台处理机的PRAM上,$W(n)$个操作步可在$O(W(n)/p)$时间内被p台处理机并行完成.

2 极大边匹配的高效并行算法

算法. Matching

* 本文研究得到国家自然科学基金、国家863高科技项目基金、山东省自然科学基金和山东大学跨世纪人才基金资助.
作者马军,1956年生,博士,教授,主要研究领域为算法分析与设计,人工智能.岩间一雄,1951年生,博士,教授,博士生导师,主要研究领域为算法分析与设计,人工智能.顾谦平,1956年生,博士,副教授,主要研究领域为算法分析与设计,人工智能.

本文通讯联系人:马军,济南250100,山东大学计算机科学系

本文1997-03-27收到原稿,1998-01-05收到修改稿

输入：图 G 的邻接矩阵.
输出：G 的一个极大边匹配 M.
(1) $i:=1$;$G_1:=G$;$M:=\varnothing$;$\forall v\in V$，计算顶点度数 $degree(v)$.
(2.1) 若 $G_i=\varnothing$，则返回；
(2.2) 调用过程 Forest，建立 G_i 的一个生成林 F_i^1.
(2.3) $\forall v\in V(T_i)$，$T_i\in F_i^1$，计算 $root(v)$，$root(v)$ 为 T_i 的根顶点标号. 对每一边 $(v,w)\in T_i$，用弧 $\langle v,w\rangle$，$\langle w,v\rangle$ 替代，T_i 变为有向欧拉图 C_i，通过 C_i 把 $root(v)$ 并行地送到 T_i 的每个顶点.[5] 然后把 C_i 复原为 T_i.
(2.4) 调用过程 F-Matching，找出 F_i^1 的一个极大边匹配 M_i^1.
(2.5) 建立子图 $G'_i(V',E')$，$E'=\{(v,w)\mid(v,w)\in E(G_i)$，满足在 F_i^1 中 $root(v)\neq root(w)$ 且 $v,w\notin V(M_i)\}$.
(2.6) 调用过程 Forest，建立 $G'i$ 的一个生成林 F_i^2.
(2.7) 调用过程 F-Matching，找出 F_i^2 的一极大边匹配 M_i^2.
(2.8) $M_i:=M_i^1\cup M_i^2$.
(2.9) 删除 G_i 中至少有一端点在 $V(M_i)$ 的边及孤立顶点. 称残留的子图为 G_{i+1};$M:=M\cup M_i$; goto (2.1).
end Matching.

Procedure Forest
输入：子图 G_i 的邻接矩阵.
输出：由数组 $F(1..n)$ 表示的 G_i 的生成林 F_i.
(0) 对每个 $v\in G_i$，$F(v)=v$；
(1) 设 $\Gamma_i(v)=\{x\mid(x,v)\in E(G_i)$ 为顶点 v 在 G_i 中的邻域$\}$；w 为 $\Gamma_i(v)$ 中具有最大顶点度数的顶点，若 $degree(w)\geqslant degree(v)$，则 $F(v)=w$；
(2) for $v\in V(G_i)$ par-do if $F(F(v))=v$ and $(w<v)$ then $F(v)=v$; endif; endfor;
(3) if $F(v)=v$ then 随机选择 $w\in\Gamma_i(v)$，令 $F(v)=w$; endif;
end Forest.

Procedure F-Matching
输入：由数组 $F(1..n)$ 表示的生成林 F_i.
输出：F_i 的一个极大边匹配 M_i.
Local array $B(1..n,1..2)$ of integer;
Sub-Procedure Sort_Matching
(1) 对 B 按字典序排序并存到 B;
(2) For all i，$2\leqslant i\leqslant n$ par-do if $B(i,1)=B(i-1,1)$ then $B(i,1):=\infty$;
(3) For all i，$1\leqslant i\leqslant n$ par-do
  if $B'(i,1)\neq\infty$ then {送 $(B'(i,1),B'(i,2))$ 到 M_i；标记顶点 $B'(i,1)$ 和 $B'(i,2)$ 为 M_i 的顶点}；
  endif;
end Sort_Matching;
Sub-Procedure $Match(x)$
  if $x=0$ then
   for $1\leqslant i\leqslant n$ par-do
    if $(F(i)=i)$ or ($depth(i)$ 为偶数) then $B(i,1):=\infty$
     else $\{B(i,1):=F(i);\ B(i,2):=i;\}$; endif;
    调用过程 Sort_Matching; endfor;
  else for $1\leqslant i\leqslant n$ par-do
     if $(F(i)=i)$ or ($depth(i)$ 为奇数) or (顶点 i 或 $F(i)$ 已为 M_i 的顶点) then
      $B(i,1):=\infty$ else $\{B(i,1):=F(i);\ B(i,2):=i;\}$ endif;
     调用过程 Sort_Matching; endfor;
end $Match(x)$;
/steps of algorithm F-Matching/

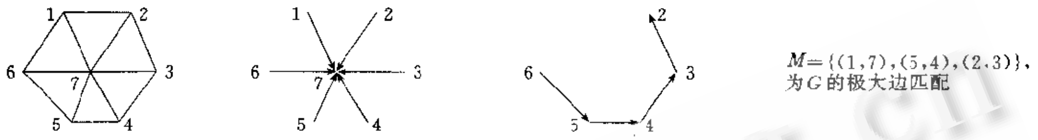
(1) for each $v\in V(G_i)$ and $v\in V(T_j)$ par-do

计算 $depth(v)$,$depth(v)$为顶点 v 在子树 $T_j\in F_i$ 的深度；标记 v 为 M_i 的顶点. endfor;

(2) for $x=0$ to 1 do 调用过程 $Match(x)$;

end F-Matching.

图1给出对算法 Matching 的一个执行过程图解.



(1) 无向图　(2) F_1^1 ，并求得 $M_1^1=\{(1,7)\}$　(3) F_1^1 ，并求得 $M_1^1=\{(4,5),(2,3)\}$　(4) 循环次数为2，算法结束

图1　关于算法Matching执行过程的图解

3　对算法 Matching 的理论分析

称算法 Matching 的第2步循环的第 i 次执行为阶段 i. 下面的命题显然成立.

引理 3.1. $M\subseteq E$ 为 $G(V,E)$的一个极大边匹配，当且仅当 M 为一个边匹配，并且任意 G 中的边，至少有一个端点在 M 中.

引理 3.2. $F_i^1(F_i^2)$为 $G_i(G'_i)$的有根生成林.

证明：显然，在过程 Forest 的步骤(1)～(2)中，所有 G_i 的顶点均被加到 F_i^1 中. 步骤(2)删除可能的长度为数的回路，并且由于<关系为自反、非对称和传递的. 因此过程结束后，F_i^1 中无回路. 同理可证 F_i^2 为 G'_i 的有根生成林. □

引理 3.3. $M_i^1(M_i^2)$为 $F_i^1(F_i^2)$上的极大边匹配.

证明：因在过程 F-Matching 的步骤(2)～(3)中，F_i^1 中的每条边 $e=(v,F(v))$都试图成为 M_i^1 的边，e 未能成为 M_i^1 的边，当且仅当 e 的一个端点已在 $V(M_i^1)$中. 由引理3.1，引理得证. 同理可证 M_i^2 为 F_i^2 的极大边匹配. □

根据上述引理及算法 Matching 的步骤(2.9)，即可得到推论3.1.

推论 3.1. 在阶段 i 后，所有 F_i^1，F_i^2 的边将被删除.

因对 G_i 中任意边 $e=(v,w)$，若 $v,w\notin V(M_i)$，e 将留在 G_{i+1}中，并且 e 在阶段 i 被从 G_i 中删除，仅当有一端点在 $V(M_i)$中. 由引理3.1和3.3，可直接推出定理3.1.

定理 3.1. $M=\bigcup_i M_i$ 为 G 的一个极大边匹配.

引理 3.4. 在 EREW PRAM 并行计算模型上，算法 Matching 的每个阶段的运算，均可在 $O(\log n)$时间内被 $O((m+n)/\log n)$处理机完成.

证明：因已知对 n 元素排序、树函数 $depth(v)$、$root(v)$及在树上欧拉回路的计算，均可在 $O(\log n)$时间内被 $O((m+n)/\log n)$处理机完成[5]，而其他步骤的工作量均不超过 $O(n+m)$，由定理1.1，引理得证. □

引理 3.5. 设 G_i 为连通子图，S_i 为在阶段 i 被删除的边集合，则只用 S_i 中的边，足以建造 G_i 的一棵生成树.

证明：因 $M_i:=M_i^1\cup M_i^2$ 且$\forall$ $e\in E(G_i)$，在阶段 i，e 被删除当且仅当 e 的一个端点在 $V(M_i)$中. 令 $S_i=S_1\cup S_2$，满足 $e\in S_1(S_2)$，当且仅当 e 的一个端点在 $V(M_i^1)(V(M_i^2))$中. 设 $F_i=F_i^1\cup F_i^2$，由 $k(\geqslant 1)$棵子树组成，显然，$E(F_i^1)\cap E(F_i^2)=\varnothing$. 下面对 k 做归纳证明.

(1) $k=1$，则 $F_i=F_i^1$，为 G_i 的生成树. $S_2=\varnothing$. 引理成立. 对 $k=2$，由算法 Matching 的第(2.5)、(2.6)步的执行可知，$E(G'_i)=\{(v,w)|v,w$ 分别为 T_1 和 T_2 的顶点，且 $v,w\notin V(M_i)\}$. 因 $S_2\neq\varnothing$，故 $E(G'_i)\neq\varnothing$，则存在边 $e\in S_2$，使得 $T_1\cup T_2\cup\{e\}$为 G_i 的一棵生成树. 类似地，可推出 $k=3$ 时也成立.

(2) 设引理对任意 $k\leqslant t(t>2)$成立，下面证明 $k=t+1$ 时也成立.

显然，假设 T_x 和 T_y 之间无边属于 S_2，由归纳假设，引理对子图 G_i-T_x 和 G_i-T_y 均成立. 在过程 Forest 执行中，若边 e 的最大顶点度<3，则 e 必为树边，则推知若 G_i-T_x 和 G_i-T_y 在 G_i 中均不为空(若有一为空，则由

归纳假设,引理成立),必存在 $v\in T_x$ 和 $w\in T_y$,其顶点度数在 G_i 中均大于 2,并在 G_i 中有一条连接 v 与 w 的路径 P,满足 P 的边均为树边.故引理也为真.由归纳法原理,引理得证. □

定理 3.2. 算法 Matching 的阶段数一定不会大于 $a(G)$.

证明:G 可按下法分解为 $F_1,F_2,\ldots,F_{a(G)}$:对每个 G_i 的连通分支 C_l.找出一棵生成树 T_l,且令 $F_i:=\bigcup T_l$;$G_{i+1}:=G_i-F_i$;重复此过程直至 $G_i=\varnothing$.

设 A_i,B_i 分别为在上述算法与算法 Matching 的第 i 步删除的边集,由引理 4.5,$\bigcup A_i\subseteq\bigcup B_i$,$1\leqslant i\leqslant a(G)$,故算法 Matching 的阶段数不会超过 $a(G)$. □

定理 3.3 小结了上述讨论.

定理 3.3. 算法 Matching 可在 $O((n+m)/\log n)$ 处理器的 CREW(concurrent-read and exclusive-write) PRAM 模型上,在 $O(a(G)\log n)$时间内完成.

4 结语及应用

由定理 3.3 和集合 Π 的定义,对任意无向图 $G\in\Pi$,算法 Matching 在具有 $O((n+m)/\log n)$处理器的 EREW PRAM 并行计算模型上的运行时间为 $O(\log n)$.显然 $\Omega(m)$为 MM 求解的时间下界,所以我们的算法为集合 Π 上的最佳并行算法.我们认真地分析了已知对 MM 的最好的并行算法[2,6~9]在平面图上的执行情况,这些算法的执行时间至少为 $O(log^2n)$.故我们的并行算法在集合 Π 上为最快的算法.我们推测,在使用多项式个数处理器的前提下,在 EREW PRAM 上不会存在比 $O(\log n)$更快的对 MM 的并行算法.

文献[1]中给出一种求解 G 的极大顶点不交路径 MVDP(maximal vertex disjointed path)的并行算法,在 CREW PRAM 上的执行时间为 $O(\sqrt{n}\log^4 n)$,其中 $\log^4 n$ 为并行计算 MM 的时间.显然对任意 $G\in\Pi$,应用我们的算法可使 MVDP 的计算时间减少到 $O(\sqrt{n}\log n)$.新的 MVDP 算法又可改进在集合 Π 上对下列问题的并行求解时间:求解 0~1 网络流、作业调度、无向图的深度优先搜索和哈密顿回路求解问题.[1]

参考文献

1 Goldberg A V, Plotkin S A. Sublinear—time parallel algorithms for matching and related problems. Journal of Algorithms, 1993,14:180~213
2 Iseaeli A, Shiloach Y. An improved parallel algorithm for maximal matching. Information Processing Letters, 1986,22, 57~60
3 Chiba Norishige, Nishizeki Takao. Arboricity and subgraph listing algorithms. SIAM (Society for Industrial and Applied Mathematics) Journal of Computers, 1988,14(1):210~223
4 Harary F. Graph Theory, Revised. Reading, MA: Addison-Wesley Publishing Company, 1972. 35~130
5 Jaja J. An introduction to parallel algorithms. Reading, MA: Addison-Wesley Publishing Company, 1992. 120~230
6 Chen Z. A fast and efficient NC algorithm for maximal matching. Information Processing Letters, 1995,55:303~307
7 Han Y. An improvement on parallel computation of a maximal matching. Information Processing Letters, 1995,56:343~348
8 Iseaeli A, Itai A. A fast and simple randomized parallel algorithm for maximal matching. Information Processing Letters, 1986,22:77~80
9 Kelsen P. An optimal parallel algorithm for maximal matching. Information Processing Letters, 1994,55:223~228

A Parallel Maximal Matching Algorithm for Undirected Graphs with Applications

MA Jun[1] IWAMA Kazuo[2] GU Qian-ping[3]

[1](*Department of Computer Science Shandong University Ji'nan* 250100)
[2](*Department of Computer Science Kyoto University Kyoto Japan*)
[3](*Department of Software University of Aizu Wakamatsu Japan*)

Abstract A fast and optimal parallel maximal matching algorithm is proposed for a class of graphs. It runs in $O(\log n)$ time with $O((m+n)/\log n)$ processors on a EREW PRAM (exclusive-read and exclusive-write parallel random access machine).

Key words Parallel graph algorithms, maximal matching.