UILU-ENG-90-2250 ACT-114

Applied Computation Theory

AN O(LOG N) TIME COMMON CRCW PRAM ALGORITHM FOR MINIMUM SPANNING TREE

Michael M. Wu

Coordinated Science Laboratory College of Engineering UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Approved for Public Release. Distribution Unlimited.

REPORT DOCUMENTATION PAGE OMERAGE OF COULE REPORT SECURITY CLASSIFICATION Ib. RSTRICTIVE MARKINGS None Minited Section A Distribution Analasitity of REPORT Approved for public relass; distribution unlimited DECLASSIFICATION AUTHORITY Approved for public relass; DECLASSIFICATION AUTHORITY A MONTORING ORANIZATION REPORT NUMBERS; DECLASSIFICATION OF MARGENER; S. MONTORING ORANIZATION REPORT NUMBERS; NAME OF PERFORMING ORANIZATION MUMBERS; The MONTORING ORANIZATION OF FUNCTION NUMBERS; NAME OF FUNCING / SPONSORING Bb. OFFCE SYMBOL P. POCUMENT INSTRUMENT IDENTIFICATION NUMBERS; NOODIA-BST (CON, Stare, and ZP Code) Art lington, VA 22217 Processary MORE PERFORMINERS; NO (flog on) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree TASK MORE PERFORMINER; Num, Michael M. Nu, Michael M. Nu, Michael M. Nu, Michael M. Nu, Michael M. Nu, Michael M. <t< th=""><th>RITY CLASSIFICATION OF THIS PAGE</th><th>N PAGE</th><th></th><th></th><th>Form Approved OMB No. 0704-0188</th></t<>	RITY CLASSIFICATION OF THIS PAGE	N PAGE			Form Approved OMB No. 0704-0188	
None None SECURTY CLASSIFICATION AUTHORITY 3. Distribution unlimited SECURTY CLASSIFICATION AUTHORITY 4. Distribution unlimited SECURTY CLASSIFICATION AUTHORITY 5. OFFICE SYMBOL ADDRESS (CD; State, and 20°Code) N/A ADDRESS (CD; Sta				MARKINGS		-
SECURITY CLASSIFICATION AUTHORITY 1. DEFINISUIDUR/AVAULABLET OF REASES Approved for public release; distribution unlimited Approved for public release; distribution unlimited SECURITY CLASSIFICATION / COWNGRADING SCHEDULE 5. MONITORING ORGANIZATION REPORT NUMBER(S) UILU-ENG-90-2250 (ACT #114) NAME OF FERFORATION REPORT NUMBER(S) 5. MONITORING ORGANIZATION REPORT NUMBER(S) UILU-ENG-90-2250 (ACT #114) NAME OF FERFORATION OFFICE SYMBOL University of 1111ning N/A 7. NAME OF NUMBER(S) 5. OFFICE SYMBOL (# applicable) NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 7. NAME OF NUMBERS NODOLSS (CD, State, and ZP Code) 10. SOURCE OF FUNDING NUMBERS Arlington, VA 22217 10. SOURCE OF FUNDING NUMBERS TITLE (Include Security Classification) 10. SOURCE OF FUNDING NUMBERS An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree 10. FERSONAL AUTHORIS) 11. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) 13. SUPPLEMENTARY NOTATION 14. DATE OF REPORT (Year, Month, Dary) 15. PAGE COUNT 10 FELD GROUP 11. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Asstract (Continue on averas if necessary and identify by block number)			None			
DECLASSIFICATION / DOWINGRADING SCHEDULE Approved Toy PUBLIC TEREST, distribution unlimited ERFORMING ORGANIZATION REPORT NUMBER(S)	SECURITY CLASSIFICATION AUTHORITY					
CONSTRUCTOR CONTORING ORGANIZATION REPORT NUMBER(S) ERROMING ORGANIZATION REPORT NUMBER(S) S. MONTORING ORGANIZATION (ACT #114) NAME OF PERFORMING ORGANIZATION Sb. OFFICE SYMPLO. (M* applicable) 7a. NAME OF MONITORING ORGANIZATION Office of Naval Research University vof Ellinois N/A NAME OF FERFORMING ORGANIZATION Outvoint vof Ellinois N/A NAME OF FERFORMING ORGANIZATION University vof Ellinois N/A NAME OF FERFORMING ORGANIZATION University vof Ellinois N/A NAME OF FERFORMING ORGANIZATION University vof Ellinois N/A NAME OF FERFORMING ORGANIZATION ORGANIZATION Office of Naval Research Standiation volume No0014-85K-0570 An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree MOX_CET PROGRAM AUTHORS) NO NO May, Michael M. No NO May, Michael M. 13. TIME COVERED REMOMAL AUTHORS) NO No, Michael M. 13. TIME COVERED REMOMAL AUTHORS) NO A. STPECE OF REPORT 13. SUBSECT TERMS (Continve on reverse if			Approved	for public	release;	
ENFORMING UNGAUNCED (NOT BOTH STATE) DILU-ENG-90-2250 (ACT #114) DILU-ENG-90-2250 (ACT #114) NAME OF ENFORMING ONGANIZATION 66. OFFICE SYMBOL (If applicable) 74. NAME OF MONITORING ONGANIZATION Office of Naval Research Coordinated Science Lab University of Illinois N/A 74. NAME OF MONITORING ONGANIZATION Office of Naval Research NAME OF FUNDING/SFONSORING ONGANIZATION Office of Naval Research 85. OFFICE SYMBOL (If applicable) 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 ADDRESS (Chy, Stete, and ZIP Code) Arlington, VA 22217 10. SOURCE OF FUNDING NUMBERS N00014-85-K-0570 WORK UNIT AND OF FUNDING / Stete, and ZIP Code) Arlington, VA 22217 IN SOURCE OF FUNDING NUMBERS N00014-85-K-0570 TITLE (Include Security Classification) An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree IN SOURCE OF FUNDING NUMBERS N00. WORK UNIT ACCESSION NO 10 TYPE OF REPORT Technical 13b. TIME COVERED FROM TO		1				
NAME OF PERFORMING ORGANIZATION Coordinated Science Lab 66. OFFICE SYMBOL (If applicable) 74. NAME OF MONITORING ORGANIZATION Office of Naval Research Coordinated Science Lab N/A 7b. ADDRESS (Cry, State, and ZIP Code) Analysis (Stev, and ZIP Code) 100 W. Springfield Ave- Urbana, IL 61801 N/A 7b. ADDRESS (Cry, State, and ZIP Code) Arlington, VA 22217 NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable) 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 Ann O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PROBECT INC. No. ACCESSION N ACCESSION N ACCESSION N TITLE (Include Security Classification) An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree 10. NORE COMMON COMMENT Include Security Classification) A. DYPLEMENTARY NOTATION 13b. TIME COVERED IFCO REPORT TROM 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT 10. SUPPLEMENTARY NOTATION 14. UBJECT TERMS [Continue on reverse if necessary and identify by block number) 10. PRED COSATI CODES 14. SUBJECT TERMS [Continue on reverse if necessary and identify by block number) PRED GROUP 14. SUBJECT TERMS [Continue on reverse if necessary and identify by block number) PRED GROUP 14. SUBJECT TERMS [Continue on reverse if necessary and identify by block numb	ERFORMING ORGANIZATION REPORT NUMB	ER(S)	5. MONITORING	ORGANIZATION	REPORT NORT	
NAME OF PERCONNING UNDARLATION Office of Naval Research Office of Naval Research Office of Naval Research ADDRESS (City, State, and ZiP Code) N/A NAME OF FUNDING /SPONSORING Bb. OFFICE SYMBOL ORGANIZATION Office of Naval Bb. OFFICE SYMBOL Research 10. SOURCE OF FUNDING NUMBERS ADDRESS (City, State, and ZiP Code) 10. SOURCE OF FUNDING NUMBERS Research 10. SOURCE OF FUNDING NUMBERS ADDRESS (City, State, and ZiP Code) 10. SOURCE OF FUNDING NUMBERS An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PERSONAL AUTHOR(S) Nu, Michael M. Nu, Michael M. 13b. TIME COVERED ITEC (notube Security Classification) 14. DATE OF REPORT (Year, Month, Day) IS UPPLEMENTARY NOTATION 13b. TIME COVERED TRED GROUP SUB-GROUP I * SUBJECT TERMS (Continue on reverse if necessary and identify by block number) ASSTRACT (Continue on reverse if necessary and identify by block number) * We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Constant such that 0 < c ≤ 1/2.	JILU-ENG-90-2250 (ACT #1)	14)				
Coordinated Science Lab N/A University of Illinois N/A DOMES (Cry, State, and ZP Code) Arlington, VA 22217 Itol W. Springfield Ave. Arlington, VA 22217 And Of FUNDING (SPONSONING ORGANIZATION Office of Naval Bb. OFFICE SYMBOL (If applicable) 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 Research ADORESS (Cry, State, and ZP Code) N.A NO.014-85-K-0570 Arlington, VA 22217 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 WOORK UNIT AND (Ig n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PERSONAL AUTHOR(S) M. Michael M. N.A YPE OF REPORT 13. TME COVERED FROM_TO	NAME OF PERFORMING ORGANIZATION					
OUTVESTIG: State, and ZPCCode) 1101 W. Springfield Ave. Urbana, IL 61801 NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research ADDRESS (Cry, State, and ZPCCode) Arlington, VA 22217 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO0014-85-K-0570 Research ADDRESS (Cry, State, and ZPCCode) Arlington, VA 22217 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO0014-85-K-0570 Arlington, VA 22217 TITLE (include Security Classification) An (log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT T3b. TIME COVERED TO 14. DATE OF REPORT (Year, Month, Day) TS PAGE COUNT 10 Supersementary NOTATION *** COSATI CODES *** TATELE! algorithms, minimum spanning tree, CRCW PRAM **** AUTHORY NOTATION ***** OF eresent an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using m+n ¹⁺²⁴ processors and O(m+n ^{++*}) space in O(log n) time, where e is a constant such that 0 < e ≤ 1/2.	oordinated Science Lab		Office of Naval Research			
ADDRESS (GP, Jahr, and Lid Ave. Urbana, IL 61801 Arlington, VA 22217 NAME OF FUNDING (SPONSORING ORGANIZATION Office of Naval Research Bb. OFFICE SYMBOL (If applicable) 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 ARSesearch 10. SOURCE OF FUNDING NUMBERS N00014-85-K-0570 NMERS Arlington, VA 22217 In SOURCE OF FUNDING NUMBERS N00014-85-K-0570 WORK UNIT FLEMENT NO. Arlington, VA 22217 In SOURCE OF FUNDING NUMBERS N00014-85-K-0570 WORK UNIT FLEMENT NO. WORK UNIT FLEMENT NO. Arlington, VA 22217 In SOURCE OF FUNDING NUMBERS N00014-85-K-0570 WORK UNIT FLEMENT NO. WORK UNIT FLEMENT NO. WORK UNIT FLEMENT NO. TITLE (include Security Classification) An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree In Specific Count No. In Specific Count 10. PRECONT Technical Inter COVERED FRED Inter Coverse if necessary and identify by block number) Inter Coverse if necessary and identify by block number) ARSTRACT (Continue on reverse if necessary and identify by block number) Inter Processors and O(m+n ⁺⁺⁺) space in O(log n) time, where c is a constant such that $0 < c \le 1/2$. Intersectified 20. ODSTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USES Intersectified Intersectified 20. ODSTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USES Intersectified	University of Illinois	N/A	75. ADDRESS (C	ity, State, and 2	ZIP Code)	
Urbana, IL 61801 NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research ADDRESS (Civ, Stem, and ZiP Code) Arlington, VA 22217 TITLE (include Security Classification) An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree FERSONAL AUTHOR(S) Wu, Michael M. a. TYPE OF REPORT 13b. TIME COVERED Technical 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT Det of REPORT 16. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) 16. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) ASSTRACT (Continue on reverse if necessary and identify by block number) ASSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using m-n ¹⁺²⁺² processors and O(m+n ⁺⁺⁺) space in O(log n) time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION/AVAULABILITY OF ABSTRACT						
NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Bb. OFFICE SYMBOL (If applicable) 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-K-0570 Research ADDRESS (Cry, State, and ZIP Code) Arlington, VA 22217 10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. TASK NO. WORK UNIT ASK NO. TITLE (Include Security Classification) An 0(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree TASK NO. Work UNIT ACCESSION NO. TTEL (Include Security Classification) An 0(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree Tree PERSONAL AUTHOR(S) Wu, Michael M. 14. DATE OF REPORT (Vear, Month, Day) 15. PAGE COUNT 0 clober, 1990 Technical TBCD TO 14. DATE OF REPORT (Vear, Month, Day) 15. PAGE COUNT 10 SUPPLEMENTARY NOTATION 18. SUBJECT TERMS (Continue on reverse if necessary and Identify by block number) 10 RELD GROUP SUB-GROUP 18. SUBJECT TERMS (Continue on reverse if necessary and Identify by block number) ABSTRACT (Continue on reverse if necessary and Identify by block number) Not reverse if necessary and Identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using m-n ⁺⁴²⁰ processors and O(m+n ⁺¹⁺¹) space in O(log n) time, where e is a constant such that 0 < e ≤ 1/2.					;	
NAME OF FUNCTION OFFICE of Naval Iff spokesber N00014-85-K-0570 Research ADDRESS (City, State, and ZIP Code) ID. SOURCE OF FUNDING NUMBERS ADDRESS (City, State, and ZIP Code) ID. SOURCE OF FUNDING NUMBERS ADDRESS (City, State, and ZIP Code) ID. SOURCE OF FUNDING NUMBERS TITLE (Include Security Classification) Recearch An 0 (log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PERSONAL AUTHOR(S) Wu, Michael M. Trechnical Trechnical Image: Report Tore Technical FROM To SUPPLEMENTARY NOTATION The of Report Continue on reverse if necessary and identify by block number) Parallel algorithms, minimum spanning tree, CRCW PRAM NAME of a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ID CONSTRIBUTION / AVAILABILITY OF ABSTRACT ID DIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 21. ABSTRACT SECURITY CLASSIFICATION		ISH OFFICE SYMBOL	9. PROCUREME	NT INSTRUMENT	IDENTIFICATIO	N NUMBER
Research ADDRESS (City, State, and ZIP Code) ADDRESS (City, State, and ZIP Code) PROJECT TASK WORK UNIT ADDRESS (City, State, and ZIP Code) ATLINGTON TITLE (include Security Classification) An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PROMINE AUTHOR(S) Wu, Michael M. 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT Technical Technical 13. TIME COVERED 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT October, 1990 10 Supplementation on reverse if necessary and identify by block number) TEELD GROUP 14. Subject TERMS (Continue on reverse if necessary and identify by block number) ABSTRACT (Continue on reverse if necessary and identify by block number) ABSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using m+n ¹⁺²⁴ processors and O(m+n ^{+++*}) space in O(log n) time, where \$\epsilon is a constant such that $0 < \epsilon \leq 1/2$. 21. ABS	ORGANIZATION Office of Naval					
ADDRESS (City, State, and LP COUP) Arr Lington, VA 22217 PROGRAM PROJECT TASK WORK UNIT, Arr Lington, VA 22217 TITLE (include Security Classification) An 0(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PROJECT TASK WORK UNIT, Test (S) Wu, Michael M. Image: Court of the course of the cour	Research					
COSATI CODES 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) FIELD GROUP 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) TO 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) RESTRACT (Continue on reverse if necessary and identify by block number) No Parallel algorithms, minimum spanning tree, CRCW PRAM ABSTRACT (Continue on reverse if necessary and identify by block number) No No Ve present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using m+n ¹⁺² processors and O(m+n ⁺⁺) space in O(log n) time, where e is a constant such that $0 < e \le 1/2$. 20. DISTRIBUTION /AVAILABILITY OF ABSTRACT DITC USERS 21. ABSTRACT SECURITY CLASSIFICATION UNLIMITED SAME AS RPT.	ADDRESS (City, State, and ZIP Code)	•			TASK	WORK UNIT
An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree PERSONAL AUTHOR(S) Wu, Michael M. TME COVERED 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT Technical FROM TO October, 1990 10 SUPPLEMENTARY NOTATION COSATI CODES Image: Supplementary NOTATION COSATI CODES Image: Supplementary NOTATION Image: Supplementary NOT	Arlington, VA 22217	· · ·		NO.	NO.	ACCESSION INC
COSATI CODES 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Parallel algorithms, minimum spanning tree, CRCW PRAM Image: Sub-GROUP Parallel algorithms, minimum spanning tree, CRCW PRAM ABSTRACT (Continue on reverse if necessary and identify by block number) ABSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 21. ABSTRACT SECURITY CLASSIFICATION Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. . TYPE OF REPORT 13b. TIME	COVERED	14. DATE OF RE	PORT (Year, Mo		
Pieto GROUP Job cites ABSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS 21. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. Image: Type of Report Technical 13b. TIME FROM_	COVERED	14. DATE OF RE	PORT (Year, Mo		
Pieto GROUP Job cite ABSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS 21. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS	PERSONAL AUTHOR(S) Wu, Michael M. a. TYPE OF REPORT Technical FROM_	COVERED	14. DATE OF RE	PORT (Year, Mo		
Pieto Image: Continue on reverse if necessary and identify by block number) D. ABSTRACT (Continue on reverse if necessary and identify by block number) We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS 21. DISTRIBUTION/AVAILABILITY OF ABSTRACT DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. Type OF REPORT Technical SUPPLEMENTARY NOTATION	COVERED TO	14. DATE OF RE October	PORT (Year, Mo	nth, Day) 15.	10
We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$.	PERSONAL AUTHOR(S) Wu, Michael M. Type OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES	COVERED TO	14. DATE OF RE October	PORT (Year, Mo	nth, Day) 15.	10
We present an algorithm for finding the minimum spanning tree of a graph with n vertices and m edges on a Common CRCW PRAM using $m+n^{1+2\epsilon}$ processors and $O(m+n^{1+\epsilon})$ space in $O(\log n)$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT 21 . ABSTRACT SECURITY CLASSIFICATION Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. a. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES	COVERED TO	14. DATE OF RE October	PORT (Year, Mo	nth, Day) 15.	10
m edges on a Common CRCW PRAM using $m+n$ processors and $O(m+n')$ option in $O(e^{-1}g^{-1})$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT Unclassified Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. 13b. TIME Technical 13b. TIME FROM_ SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP	COVERED TO 18. SUBJECT TERMS Parallel a	14. DATE OF RE October S (Continue on rev algorithms, 1	PORT (Year, Mo	nth, Day) 15.	10
m edges on a Common CRCW PRAM using $m+n$ processors and $O(m+n'')$ option in $O(e^{-1}g^{-1})$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. O DISTRIBUTION / AVAILABILITY OF ABSTRACT Unclassified/UNLIMITED SAME AS RPT. \Box DTIC USERS Unclassified Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. 13b. TIME Technical 13b. TIME FROM_ SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP	COVERED TO 18. SUBJECT TERMS Parallel a	14. DATE OF RE October S (Continue on rev algorithms, 1	PORT (Year, Mo	nth, Day) 15.	10
m edges on a Common CRCW PRAM using $m+n$ processors and $O(m+n')$ option in $O(e^{-1}g^{-1})$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT Unclassified Unclassified 21. ABSTRACT SECURITY CLASSIFICATION Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. 13b. TIME Technical 13b. TIME FROM_ SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP	COVERED TO 18. SUBJECT TERMS Parallel a	14. DATE OF RE October S (Continue on rev algorithms, 1	PORT (Year, Mo	nth, Day) 15.	10
m edges on a Common CRCW PRAM using $m+n$ processors and $O(m+n')$ option in $O(e^{-1}g^{-1})$ time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT Unclassified Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. Type OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necess	COVERED TO 18. SUBJECT TERMS Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 t number)	PORT (Year, Mo , 1990 merse if necessary minimum spa	nth, Day) 15. y and identify b anning tree	10 y block number) e, CRCW PRAM
time, where ϵ is a constant such that $0 < \epsilon \le 1/2$. 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT 21. ABSTRACT SECURITY CLASSIFICATION Unclassified Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP ABSTRACT (Continue on reverse if necess	COVERED TO 18. SUBJECT TERMS Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. Type OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number)	PORT (Year, Mo , 1990 minimum spa	and identify b anning tree	10 y block number) e, CRCW PRAM
LA UNCLASSIFIED/UNLIMITED LI SAME AS NET. LI D'IL OSTIC OSTIC LISUONE (Include Area Code) 22c. OFFICE SYMBOL	PERSONAL AUTHOR(S) Wu, Michael M. TYPE OF REPORT Technical SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necessary We present an algorithmeter of a Common CH	COVERED TO TO TO Parallel a Parallel a any and identify by block	14. DATE OF RE October S (Continue on rev algorithms, 1 k number) inimum spannin n+n ^{1+2¢} process /2.	PORT (Year, Mo , 1990	(raph with n $v_{+n}^{1+\epsilon}$) space	10 y block number) e, CRCW PRAM
	PERSONAL AUTHOR(S) Wu, Michael M. Technical 13b. TIME Technical FROM_ SUPPLEMENTARY NOTATION COSATI CODES FIELD GROUP SUB-GROUP ABSTRACT (Continue on reverse if necess We present an algorithm edges on a Common CE time, where ϵ is a constant of time, where ϵ is a constant of	COVERED TO TO TO Parallel a Parallel a any and identify by block the for finding the m CCW PRAM using m such that $0 < \epsilon \le 1/2$	14. DATE OF RE October S (Continue on rev algorithms, 1 t number) tinimum spannin n+n ^{1+2¢} process 2.	PORT (Year, Mo , 1990 minimum spa minimum spa ng tree of a g sors and O(m.	and identify b anning tree traph with n v +n ^{1+c}) space	y block number) e, CRCW PRAM

UNCLASSIFIED

An O(log n) Time Common CRCW PRAM Algorithm for Minimum Spanning Tree

•

Michael M. Wu

Department of Electrical and Computer Engineering and Beckman Institute University of Illinois at Urbana-Champaign Urbana, Illinois 61801, U.S.A.

17 October 1989

Abstract

We present an algorithm for finding the minimum spanning tree of a graph with *n* vertices and *m* edges on a Common CRCW PRAM using $2m+n^{1+2\varepsilon}$ processors and $O(m+n^{1+\varepsilon})$ space in $O(\log n)$ time, where ε is a constant such that $0 < \varepsilon \le 1/2$.

Keywords: parallel algorithms, minimum spanning tree, CRCW PRAM

Supported by the Office of Naval Research under Contract N00014-85-K-0570.

1. INTRODUCTION

Let G = (V, E) be a connected graph with a set of vertices V and a set of edges E in which each edge e has a weight w(e). Without loss of generality, assume that the edge weights are distinct. Hence the minimum spanning tree (MST) of G is unique. Let n = |V| and m = |E|. We present an algorithm for finding the MST of G on a Common CRCW PRAM using $2m + n^{1+2\varepsilon}$ processors and $O(m+n^{1+\varepsilon})$ space in $O(\log n)$ time, where ε is a constant such that $0 < \varepsilon \le 1/2$.

1

1

I

The MST problem is an important problem of combinatorial optimization. Some practical applications of MST's include the design of computer, communication, and transportation networks. Graham and Hell [11] gave an extensive history of the MST problem.

Yao [13] and Cheriton and Tarjan [4] designed sequential MST algorithms that ran in time $O(m \log \log n)$. Fredman and Tarjan [9] gave an improved algorithm which ran in time $O(m \beta(m,n))$, where $\beta(m,n) = \min \{i \mid \log^{(i)}n \leq \frac{m}{n}\}$. If $m \geq n$, then $\beta(m,n) \leq \log^* n$. Gallager *et al.* [10] presented a distributed MST algorithm which used at most $5n \log n + 2m$ messages and $O(n \log n)$ time. Awerbuch [1] presented an optimal distributed MST algorithm that required $O(m+n \log n)$ messages and O(n) time.

There have also been several parallel MST algorithms. Chin *et al.* [5] presented an efficient algorithm for the CREW PRAM which ran in time $O(\log^2 n)$ using $O\left(\frac{n^2}{\log^2 n}\right)$ processors. Hirschberg [12] gave an algorithm for the Common CRCW PRAM which ran in time $O(\log n)$ using n^3 processors. Awerbuch and Shiloach [2] designed an algorithm for the Priority CRCW PRAM which ran in time $O(\log n)$ using m+n processors and O(m+n) space.

In this paper, we employ some of the results of Fich *et al.* [7] and modify the algorithm of [2] to obtain a Common CRCW PRAM MST algorithm. A straightforward modification yields an algorithm that runs in time $O(\log n)$ using $mn + n^{1+2\varepsilon}$ processors. We then reduce the number of processors to

 $2m + n^{1+2\epsilon}$. The amount of space used by our algorithm is $O(m+n^{1+\epsilon})$. Our algorithm has the same running time as the algorithm of [12] and uses fewer processors. For mildly dense graphs, where $m = \Omega(n^{1+2\epsilon})$, our algorithm has the same performance as the algorithm of [2] and uses a weaker CRCW PRAM model. Boppana [3] and Fich, *et al.* [8] established that the time separation between

the Priority PRAM and the Common PRAM each with p processors is $\Theta\left[\frac{\log p}{\log \log p}\right]$.

In Section 2, we describe the model of computation. In Section 3, we review the MST algorithm of [2]. In Section 4, we present the results of [7] that apply to our algorithm. In Section 5, we describe the modification of the algorithm of [2] to obtain our Common CRCW PRAM algorithm.

2. MODEL OF COMPUTATION

A CRCW PRAM consists of a set of processors and a shared memory. Each step consists of three phases. In the first phase, each processor may read one shared memory cell. In the second phase, each processor may perform local computations. In the third phase, each processor may write into one shared memory cell. Any number of processors may simultaneously read from a memory cell. If more than one processor simultaneously writes into the same memory cell, then the value that is written depends on the model.

Two CRCW models are the Priority model and the Common model. In the Priority model, each processor is assigned a unique priority. If more than one processor tries to write into the same cell, then the processor with the highest priority is the one that succeeds. In the Common model, if more than one processor tries to write into the same cell, then all the processors must write the same value.

3. THE MST ALGORITHM OF AWERBUCH AND SHILOACH

The algorithm of Awerbuch and Shiloach [2] uses a Priority CRCW PRAM. The priority of each processor is determined by its index. The smaller the index, then the higher its priority. The algorithm assigns processors to edges such that the smaller the weight of an edge, the higher the priority of

2

the corresponding processor. The assignment can be made by sorting the edges by weight and then assigning processors in order. This can be done in $O(\log n)$ time using the parallel merge sort algorithm of Cole [6]. Let p(i,j) be the processor assigned to edge (i,j).

A rooted tree is a tree whose edges are directed toward the root. A star is a rooted tree with height 1. Assume the vertices of G are numbered from 1 to n. The number of a vertex is its *id*. In the algorithm, there are variables associated with each vertex i. We will use the name of a vertex to refer to a variable associated with that vertex. The processors that operate on these variables, however, correspond to edges.

Each vertex *i* has a parent P(i), which is either another vertex or itself. If a vertex is a root, then its parent is itself. The parent-child relation defines a directed graph called the parent's graph, PG. PG has the same vertices as G. Define GP(i) = P(P(i)), and call GP(i) the grandparent of *i*.

The algorithm maintains a set T of undirected edges which always forms a forest of the MST. The algorithm adds edges to T using the property that for any subset of vertices, the edge of least weight leaving the set must belong to the MST. T grows until it becomes the MST.

The algorithm maintains the invariant that after each iteration, for each directed tree in PG, there is a subtree in T spanning the same set of vertices. The algorithm finds edges of the MST by trying to hook stars to other trees in PG. Processors that correspond to edges leaving a star try to hook the star to a tree. Edges that correspond to successful processors are added to T. After the stars are hooked, the algorithm reduces the height of each tree with a shortcut operation, where each vertex takes its grandparent to be its new parent.

T(e) is a boolean variable attached to each edge e. T(e) is initially 0. During the algorithm, if edge e is added to the T, then T(e) is set to 1. WINNER(i) contains the name of the edge corresponding to the writing processor. After the initialization, the algorithm iterates three steps until all the vertices are in the same star. The algorithm is executed in parallel by each edge processor

3

p(i,j).

Priority CRCW PRAM Algorithm

Initialization:

 $T(e) := 0 \text{ for all } e \in E$ $P(i) := i \text{ for } i = 1, \dots, n$

repeat

Step 1: (Star hooking) If i belongs to a star and $P(i) \neq P(j)$ then P(P(i)) := P(j) and WINNER(P(i)) := (i,j)If WINNER(P(i)) = (i,j) then T(i,j) := 1

Step 2: (Cycle breaking) If i < P(i) and i = GP(i) then P(i) := i

Step 3: (Shortcut operation) P(i) := GP(i)

until every vertex *i* belongs to the same star

Step 1 performs the hooking operation. Processors that correspond to edges leaving a star try to hook the star to another tree. A star is hooked to a tree by assigning the root of the star a parent that is a vertex of the tree to which the star is being hooked. If more than one processor tries to hook the star, then the processor with the highest priority succeeds. WINNER(*i*) contains the name of the edge e corresponding to the writing processor. Since edge e belongs to the MST, the algorithm sets T(e) := 1. After Step 1, every star is hooked to some tree.

Step 2 eliminates any cycles that may have been formed in the parent's graph. A cycle of length two forms when an edge's endpoints belong to two different stars and the edge is the edge of least weight leaving both stars. To break a cycle, the algorithm changes the parent pointer of the vertex with the sma'ler id to point to itself.

Step 3 performs the shortcut operation. For each vertex i, the algorithm sets the grandparent of i to be the new parent of i. Note that if more than one processor updates P(i), then the processors perform a common write operation. The height of each tree that is not a star decreases a factor of at least.

A vertex determines whether it belongs to a star by using Procedure Star_Check. At the termination of Star_Check, if ST(i) is true (false), then *i* belongs (does not belong) to a star.

5

Procedure Star_Check

ST(i) := true If $P(i) \neq GP(i)$ then ST(i) := false and ST(GP(i)) := false ST(i) := ST(P(i))

Awerbuch and Shiloach [2] established the correctness of their algorithm. We briefly justify the running time. Consider each iteration of the three steps. Steps 1 and 2 ensure that every star is hooked to some tree to yield a new tree with height greater than one. Since Step 3 reduces the height of every tree with height greater than one by a factor of at least 3/2, the sum of the heights of all the trees present at the start of the iteration is reduced by a factor of at least 3/2. Thus $O(\log n)$ iterations yield a single star. Since each iteration takes O(1) time, the algorithm runs in time $O(\log n)$.

1. 11

4. r-COLOR MINIMIZATION PROBLEM

We obtain a Common CRCW PRAM MST algorithm by modifying the implementation of Step 1 of the algorithm of Awerbuch and Shiloach. Only Step 1 uses a priority write operation. In our algorithm, we avoid the priority write by determining the processor of highest priority wanting to write to each memory cell and having only those processors write. It can be seen that the values written in the memory cells are the same as those that would have been written in the Priority CRCW PRAM model.

To determine the processor of highest priority writing to each cell, we solve a special case of the r-color minimization problem described in Fich *et al.* [7].

r-Color Minimization Problem

Before: Each processor p_i , $i = 1, \dots, p$, has a color x_i , $0 \le x_i \le r$, known only to itself. x_i represents the cell p_i wants to write, if any, and 0 otherwise.

After: Each processor p_i knows the value a_i , where $a_i = 1$ if and only if p_i is the processor of lowest index writing to the cell represented by x_i .

For our algorithm, we consider the case where r = 1. Fich *et al.* showed that on a Common CRCW PRAM with *k* memory cells the 1-color minimization problem can be solved in $O\left(\frac{\log p}{\log (k+1)}\right)$ steps. In our discussion, we present a simplified variation of their method and show how the problem can be solved in $O\left(\frac{\log p}{\log k}\right)$ steps.

Let M_1, \dots, M_k be the k memory cells. Assume without loss of generality that $k \le p^{\varepsilon}$, where ε is a consant such that $0 < \varepsilon \le 1/2$. If $k > p^{1/2}$, then only the first $p^{1/2}$ cells are needed to achieve O(1) steps.

1

The algorithm iterates the following steps. Processor p_i , $i = 1, \dots, k$, writes 0 into M_i . The processors are then divided into k groups of nearly equal size, where each group is a set of consecutively numbered processors. The first $p \mod k$ groups contain $\left\lfloor \frac{p}{k} \right\rfloor$ processors, and the remaining groups contain $\left\lfloor \frac{p}{k} \right\rfloor$ processors. A processor p_i in the *j*th group, $1 \le j \le k$, writes 1 into M_j if and only if $x_i = 1$:

The winner is the processor of smallest index with $x_i = 1$. Thus the winner is in the group corresponding to the M_j of smallest index containing 1. The algorithm determines the winning group by using the subroutine Leftmost One.

Leftmost One

Before: Cells M_i , $i = 1, \dots, k$, each contain 0 or 1.

After: M_i contains 1 if and only if all M_j for j < i were initially 0, and M_i was initially 1.

The Leftmost One algorithm compares all pairs of cells M_i and M_j , $1 \le i, j \le k$. If j < i and M_i and M_j both contain 1, then the algorithm writes 0 into M_i . The algorithm requires $k^2 \le p$ processors. After applying the Leftmost One subroutine, processors in group j read M_j . A group deter-

mines it is the winning group if its processors read a 1.

All processors that are not in the winning group set $a_i := 0$ and stop. The processors in the winning group then repeat the 1-color minimization algorithm. This process repeats until the winning group contains only one processor, the winner.

-

1

I

Each iteration of the 1-color minimization algorithm reduces the number of processors that may be the winner by a factor of k. Thus the winner is determined in at most $\left\lceil \log_k p \right\rceil$ iterations. Since each iteration takes O(1) steps, the winner is determined in O $\left(\frac{\log p}{\log k}\right)$ steps.

5. COMMON CRCW PRAM MST ALGORITHM

Our Common CRCW PRAM MST algorithm is the same as the Priority CRCW PRAM algorithm of Awerbuch and Shiloach except that Step 1 is modified to eliminate the priority concurrent write. Thus we describe the modified implementation of Step 1 only.

In Step 1 of the Priority algorithm, if more than one processor tries to hook a star with root i to a tree, then a priority write of the variable P(i) occurs. Since there is a P(i) for each vertex i, there are n cells into which processors may write. The P(i)'s are written by processors performing the hooking operation. Since processors performing the hooking operation correspond to edges leaving stars, as many as m processors may want to write into one P(i).

In the Common algorithm, we first determine the processor of highest priority writing to each P(i) and then have only that processor write. We begin with the direct implementation which requires solving the r-color minimization problem with m processors and n colors.

To maintain the O(log n) running time of the MST algorithm, Step 1 must run in time O(1). In Step 1, the Common PRAM algorithm simultaneously solves n 1-color minimization problems, one for each P(i), using the algorithm of Section 4. Each problem requires n^{ε} cells and $n^{2\varepsilon}$ processors to obtain an O(1) time solution. During the first iteration, m processors are divided into n^{ε} groups. During each iteration, a processor can determine the group to which it belongs since it knows its rank from the sort performed during the initialization phase. Each iteration reduces the number of contending processors by a factor of n^{ε} , and thus $O\left(\frac{\log m}{\log n^{\varepsilon}}\right) = O(1)$ iterations suffice. Since there are nproblems, Step 1 requires a total of $mn+n^{1+2\varepsilon}$ processors and $n^{1+\varepsilon}$ cells. We now show how to reduce the number of processors. In Step 1 of the Priority algorithm, each processor corresponding to an edge leaving a star writes to exactly one P(i). Thus in the Common algorithm, each processor wanting to write is a possible winner for only one of the n 1-color minimization problems.

The absence of non-writing processors from the groups of processors formed during the solution of the 1-color minimization problem does not affect the outcome since the processors would not have written even if they were present. Thus each processor that wants to write needs only to participate in the solution of the 1-color minimization problem corresponding to the P(i) it wants to write. Hence, for the *n* 1-color minimization problems, the algorithm requires a total of $m+n^{1+2\varepsilon}$ processors.

The remaining steps of the algorithm require 2m+n processors and O(m+n) space. Thus we have a Common CRCW PRAM algorithm for the MST problem that runs in time $O(\log n)$ using $2m+n^{1+2\varepsilon}$ processors and $O(m+n^{1+\varepsilon})$ space. For graphs that are not connected, the algorithm can be easily modified to find a minimum weight spanning forest, and thus the connected components.

ACKNOWLEDGMENT

The author would like to thank Michael C. Loui for his helpful comments.

8

REFERENCES

- [1] B. Awerbuch, Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election and related problems, *Proc. 19th Annual ACM Symp. on Theory of Computing*,
- [2] B. Awerbuch and Y. Shiloach, New Connectivity and MSF Algorithms for Shuffle-Exchange Network and PRAM, *IEEE Trans. Computers*, **36** (1987), 1258-1263.
- [3] R.B. Boppana, Optimal Separations Between Concurrent-Write Parallel Machines, Proc. 21st Annual ACM Symp. on Theory of Computing, (1989), 320-326.
- [4] D. Cheriton and R.E. Tarjan, Finding Minimum Spanning Trees, SIAM J. Comp., 5 (1976), 724-742.
- [5] F.Y. Chin, J. Lam, and I. Chen, Efficient Parallel Algorithms for some Graph Problems, Comm. ACM, 25 (1982), 659-665.
- [6] R. Cole, Parallel Merge Sort, SIAM J. Comp., 17 (1988), 770-785.
- [7] F.E. Fich, P. Ragde, and A. Wigderson, Relations Between Concurrent-Write Models of Parallel Computation, *SIAM J. Comp.*, **17** (1988), 606-627.
- [8] F.E. Fich, P. Ragde, and A. Wigderson, Simulations Among Concurrent-Write PRAMs Algorithmica, 3 (1988), 43-51.
- [9] M.L. Fredman, and R.E. Tarjan, Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms, J. ACM, 34 (1987), 596-615.
- [10] R. Gallager, P. Humblet, and P. Spira, A Distributed Algorithm for Minimum-Weight Spanning Trees, ACM Trans. on Programming Languages and Systems, 5 (1983), 66-77.
- [11] R.L. Graham, and P. Hell, On the History of the Minimum Spanning Tree Problem, Annals of the History of Computing, 7 (1985), 43-57.
- [12] D.S. Hirschberg, Parallel Graph Algorithms Without Memory Conflicts, Proc. 20th Annual Allerton Conf. on Communications, Control, and Computing, (1982), 257-263.
- [13] A.C. Yao, An O(|E| log log |V|) Algorithm for Finding Minimum Spanning Trees, Info. Proc. Letters, 4 (1975), 21-23.