

Open access · Report · DOI:10.21236/ADA200520

# Transitive reduction in parallel via branchings — Source link []

Phillip B. Gibbons, Richard M. Karp, Vijaya Ramachandran, Danny Soroker ...+1 more authors

Published on: 15 Jul 1988

Topics: Parallel algorithm, Strongly connected component, Transitive reduction and Center (category theory)

# Related papers:

- · Approximate parallel scheduling. II.: applications to logarithmic-time optimal parallel graph algorithms
- The Expected Time Complexity of Parallel Graph and Digraph Algorithms.
- Efficient Sequential and Parallel Algorithms for the Negative Cycle Problem
- Parallel computation of weighted matchings in graphs
- Planar Strong Connectivity Helps in Parallel Depth-First Search



UILU-ENG-88-2254 ACT-102

# COORDINATED SCIENCE LABORATORY College of Engineering Applied Computation Theory

# TRANSITIVE REDUCTION IN PARALLEL VIA BRANCHINGS

Phillip Gibbons Richard Karp Vijaya Ramachandran Danny Soroker Robert Tarjan

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Approved for Public Release. Distribution Unlimited.

REPORT SECURITY CLASSIFICATION Unclassified       16. RESTRICTIVE MARKINGS None         SECURITY CLASSIFICATION AUTHORITY Approved for public release; distribution unlimited       9. DSTBUTION (AVAILABULTY OF REPORT Approved for public release; distribution unlimited         SECURITY CLASSIFICATION /DOWNGRADING SCHEDULE       0. DSTBUTION (AVAILABULTY OF REPORT Approved for public release; distribution unlimited         SECURITY CLASSIFICATION /DOWNGRADING SCHEDULE       0. DSTBUTION (AVAILABULTY OF REPORT NUMBER(S)         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION Coordinated Science Lab       9. MONTORING ORGANIZATION Office of Naval Research         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION Coordinated Science Lab       N/A         NAME OF FUNCTION JOINT SERVICES       PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NOO014-84-C-0149         Ilol W. Springfield Ave. Urbana, IL 61801       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NOO0014-84-C-0149         Ilor OF FUNCTION JOINT SERVICES       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NOO0014-84-C-0149         Ilor OF FUNCTION AUTHORS       10. SOURCE OF FUNCTION NUMBER NOO0014-84-C-0149         Intri (Include Security Classification)       10. SOURCE OF FUNCTION NUMBER NOO         Transitive Reduction in Parallel Via Branchings       10. SOURCE OF FUNCTION NUMBER NOC         Z PERCONAL AUTHORS       10. DIME COVERED         SUPLEMENTARY NOTA			REPORT	DOCUMENTATIO	N PAGE			OMB No. 0704-01
APORT SECONT CLASSIFICATION       None         None       None         SECURITY CLASSIFICATION AUTHORITY       2. DISTRUCTON AVAILABILITY OF REPORT         APPROVED AUTHORITY       2. DISTRUCTON AVAILABILITY OF REPORT         APPROVED AUTHORITY       3. DISTRUCTON AVAILABILITY OF REPORT         TRECLASSIFICATION / DOWNGRADING SCHEDULE       distribution unlimited         TRECLASSIFICATION / DOWNGRADING SCHEDULE       3. MONTORING ORGANIZATION NEPORT NUMBER(S)         UILU-ENC-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION       60. OFFICE SYMEDI         Coordinated Science Lab       0/f splicable)         UILU-ENC-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION       60. OFFICE SYMEDI         Coordinated Science Lab       0/f splicable)         UILU-ENC-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION       50. OFFICE SYMEDI         Coordinated Science Lab       0/f splicable)         UILU-ENC-88       10. SOURCE OF RUNDOR NUMBERS         PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER       NOO14-84-C-0149         CORAMING ORGANIZATION       10. SOURCE OF RUNDOR NUMBERS         PROCUREMENT NATURATION TO Codes       10. SOURCE OF RUNDOR NUMBERS         PROCUREMENT NATURATION VIANALABULTY OF ASIANCE         Transitive Red		LIBITY CLASSIE			16. RESTRICTIVE	MARKINGS		
OUTCLASSIFICATION AUTHORITY       1. DEFINISTING ORGANIZATION AUTHORITY         SECURITY CLASSIFICATION AUTHORITY       1. DEFINISTING ORGANIZATION AUTHORITY         DECLASSIFICATION/DOWNGRADING SCHEDULE       Approved for public release; distribution unlimited         PERFORMING ORGANIZATION REPORT NUMBER(S)       S. MONITORING ORGANIZATION GOGANIZATION (# spankcabe)         UILU-ENC-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION (Coordinated Science Lab University of Illinois       60. OFFICE SYMBOL (# spankcabe)         NAME OF PERFORMING ORGANIZATION (M spankcabe)       60. OFFICE SYMBOL (# spankcabe)       72. NAME OF MONITORING ORGANIZATION Office of Naval Research NO0014-84-C-0149         IIO W. Springfield Ave. Urbana, IL 61801       N/A       74. ADDRESS (Cry, State, and ZIP Code) Arlington, VA 22217       70. ADDRESS (Cry, State, and ZIP Code) Arlington, VA 22217         Noncal AUTHOR(S) Cibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Technical       15. TME COVERED FROM       14. DATE OF REPORT (Yew, Mont, Day)       17. FAGE COUNT Cotober 1988         Z. PERSONAL AUTHOR(S) Cibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Technical       18. SUBEC COVERED FROM       14. DATE OF REPORT (Yew, Mont, Day)       17. FAGE COUNT Cotober 1988         Z. PERSONAL AUTHOR(S) Cibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Rom       17.         SUPPLEMENTARY NOTATION       14. DATE OF REPORT (Yew, Mont,	Inclassified				None			
Approved for public release;         OBCLASSPICATION/DOWNGRADING SCHEDULE         Approved for public release;         distribution unlimited         SECONNIG ORGANIZATION REPORT NUMBER(S)         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION CONGANIZATION REPORT NUMBER(S)         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION CONGANIZATION (F # SPIKebble)       S. MONTORING ORGANIZATION OF # SPOKOGNIZATION NUMBER (F # SPIKebble)         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION (F # SPIKebble)       S. MONTORING ORGANIZATION (F # SPIKebble)         UILU-ENG-88-2254       (ACT #102)         NUMVERSIS (Chy, Stete, and ZP Code)       N/A         ADDRESS (Chy, Stete, and ZP Code)       Arlington, VA 22217         NOODIG-84-C-0149       NOODENT IDENTIFICATION NUMBER         Recentry Cassification       In science of FUNDING NUMBERS         PROGRAM       PROGRAM         PROME       PROSECT         Tasistive Reduction in Parallel Via Branchings         PROME       Is SUBJECT TEEMS (Continue on reverse M Recenses and Kentify by block number)         Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Dauny; Tarian, Robert         FRED GROUP       Is SUBJECT TEEMS (Continue on reverse M Recessary and Kentify by block number)	2a. SECURITY CLASSIFICATION AUTHORITY				3. DISTRIBUTION / AVAILABILITY OF REPORT			
DECLASSIFICATION / DOWNGRADING SCHEDULE       distribution unlimited         DERCASSIFICATION REPORT NUMBER(S)       UILU-ENG-88-2254       (ACT #102)         UILU-ENG-88-2254       (ACT #102)       5. OFFICE SYMBOL (of spokesbe)       7. NAME OF MONITORING ORGANIZATION Coordinated Science Lab University of Illinois       S. OFFICE SYMBOL (of spokesbe)       7. NAME OF MONITORING ORGANIZATION OEGANIZATION States and ZP Code)         NAME OF FUNDING SPONSORING ORGANIZATION Joint Services       8. OFFICE SYMBOL (If spokesbe)       7. NAME OF FUNDING TORTIFICATION NUMBER NO0014-84-C-0149         Record FUNDING SPONSORING ORGANIZATION Joint Services       8. OFFICE SYMBOL (If spokesbe)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO0014-84-C-0149         ADDRESS (CRY, State, and ZP Code)       10. SOURCE OF FUNDING NUMBERS NO0014-84-C-0149       NORK UNIT RELECTORING NUMBERS         Arlington, VA 22217       PROCENT NO.       NO.       NOK       WORK UNIT ACCESSION         TITLE (include Security Classification)       Transitive Reduction in Parallel Via Branchings       10. SOURCE OF FUNDING NUMBERS PROCM       NOR       NOR       NOR       NOR       NOR         Technical       TPROM       TO       10. SUBLECT TERMS (Continue on reverse if necessary and identify by block number)       17.       5. SUB-GROUP       17.         SUBJECENTION AVAILABULTY OF ABSTRACT INTE (include security Classification)       17.       5. SUB-GROUP       17.					Approved for public release;			
PERFORMING ORGANIZATION REPORT NUMBER(S)       S. MONITORING ORGANIZATION REPORT NUMBER(S)         UILU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION       66. OFFICE SYMBOL (M' applicable)       74. NAME OF MONITORING ORGANIZATION Office of Naval Research         Coordinated Science Lab University of Illinois       66. OFFICE SYMBOL (M' applicable)       74. NAME OF MONITORING ORGANIZATION Office of Naval Research         ADDRESS (CMy, State, and ZMP Code)       N/A       7b. ADDRESS (CMy, State, and ZMP Code)         1101 W. Springfield Ave. Urbana, IL 61801       N/A       7b. ADDRESS (CMy, State, and ZMP Code)         NAME OF FUNDING FONDORING ORGANIZATION Joint Services       8b. OFFICE SYMBOL (M' applicable)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-O149         Electronics Program       8b. OFFICE SYMBOL (M' applicable)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-O149         ITTLE (Include Security Classification)       10. SOURCE OF FUNDING NUMBERS       NO.         Transitive Reduction in Parallel Via Branchings       10. SOURCE OF REORT (Year, Month, Day)       15. PAGE COUNT Technical         Z. PERSONAL AUTHOR(S) Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Gibbons, Phillip; Supplementary and identify by block number)       17         Z. COSATI CODES       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       17         S. ABSTRACT (C	DECLASSIFIC	CATION / DOWN	IGRADING SCHEDU	JLE ·	distribut	ion unlim	ited	
UTLU-ENG-88-2254       (ACT #102)         NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of 1111nois       So. OFFICE SYNBOL (ff spikebbe)       7. NAME OF MONITORING ORGANIZATION Office of Naval Research         ADDRESS (Chy, Stete, and ZP Code)       N/A       7. ADDRESS (Chy, Stete, and ZP Code)         1101 W. Springfield Ave.       Arlington, VA 22217         Virbana, IL 61801       PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO001484-C-0149         ADDRESS (Chy, Stete, and ZP Code)       8. OFFICE SYNBOL (ff spikebbe)         ATINGTON, VA 22217       10. SOURCE OF FUNDING NUMBERS PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO001484-C-0149         ATINGTON, VA 22217       10. SOURCE OF FUNDING NUMBERS         ATING CONSTRUCT       8. OFFICE SYNBOL (ff spikebbe)       10. SOURCE OF FUNDING NUMBERS         Transitive Reduction in Parallel Via Branchings       10. SOURCE OF FUNDING NUMBERS PROCENT NO.       10. SOURCE OF FUNDING NUMBERS         Technical       FBOM	PERFORMING	ORGANIZATIO	ON REPORT NUMB	ER(S)	5. MONITORING	ORGANIZATION	REPORT NU	UMBER(S)
NAME OF PERFORMING ORGANIZATION Coordinated Science Lab       56. OFFICE SYMBOL (# spakebb9)       74. NAME OF MONITORING ORGANIZATION Office of Naval Research         University of Illinois       N/A         ADDRESS (City, Stete, and ZIP Code)       N/A         1101 V. Springfield Ave. Urbana, IL 61801       N/A         NAME OF FUNDING /SPONSORING ORGANIZATION Joint Services       80. OFFICE SYMBOL (# spakebb9)       79. ADDRESS (City, Stete, and ZIP Code)         NAME OF FUNDING /SPONSORING ORGANIZATION Joint Services       80. OFFICE SYMBOL (# spakebb9)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO0014-84-C-0149         Electronics Program       10. SOURCE OF FUNDING NUMBERS       NO0       NO         Arlington, VA 22217       PROSECT Electronics Program       10. SOURCE OF FUNDING NUMBERS         TATASITIVE (Include Security Classification) Transitive Reduction in Parallel Via Branchings       11. Space OF REPORT (rese, Month, Gay)       13. PROG COUNT Technical         Technical       13. TME (CORERD FROM       14. DATE OF REPORT (rese, Month, Gay)       15. PROG COUNT October 1988       17         7.       COSAT CODES       18. SubGROUP       14. DATE OF REPORT (researy and identify by block number) directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.       17         8. ABSTRACT (Continue on reverse if necessary and identify by block number)       14. DATE OF REPORT (researy and identify by block number)	UILU-	ENG-88-22.	54 (	ACT #102)				
Avery Difference Lab       (f # applicable)       Office of Naval Research         University of lllinois       N/A       Office of Naval Research         ADDRESS (Chy, State, and ZIP Code)       N/A       Arlington, VA 22217         Intervention       Services       PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER         ONCAMUATION Joint Services       Bb OfFice SYMBOL       PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER         NOOD1484-C-0149       Intervention       NOOD1484-C-0149         Electronics Program       ID. SOURCE OF FUNDING NUMBERS         ADDRESS (Chy, State, and ZIP Code)       ID. SOURCE OF FUNDING NUMBERS         ATINGton, VA 22217       PROGRAM         Transitive Reduction in Parallel Via Branchings         Zerstonal, AUTHOR(S)       Costant Cooles         Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert         Technical       To         Technical       To         Technical       To         Zerstonal, AUTHOR(S)         PROF       Rome         Tachical       To         Supplementation       Tachical         Transitive Reduction in Parallel Via Branchings       IF acde COUNT         Costant Cooles       Tilt.         Field       GROUP <td< td=""><td>NAME OF</td><td>ERFORMING C</td><td>RGANIZATION</td><td>66. OFFICE SYMBOL</td><td>7a. NAME OF M</td><td>ONITORING OR</td><td>GANIZATION</td><td></td></td<>	NAME OF	ERFORMING C	RGANIZATION	66. OFFICE SYMBOL	7a. NAME OF M	ONITORING OR	GANIZATION	
University of Illinois       N/A         ADDRESS (CFY, Stere, and ZIP Code)       IIII W. Springfield Ave.         IIII W. Springfield Ave.       Arlington, VA 22217         When of FUNDING/SPONSORING       Bb. OFFICE SYMBOL (If applicable)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NOO014-84-C-0149         ADDRESS (CFY, Stere, and ZIP Code)       Bb. OFFICE SYMBOL (If applicable)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NOO014-84-C-0149         ADDRESS (CFY, Stere, and ZIP Code)       III. SOURCE OF FUNDING NUMBERS       NOO14-84-C-0149         ATINGton, VA 22217       III. SOURCE OF FUNDING NUMBERS         ATINGton, VA 22217       III. SOURCE OF FUNDING NUMBERS         TITLE (include Security Classification)       Transitive Reduction in Parallel Via Branchings         TTACK Taching III; Rarp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Technical       IIII. TIME COVERED III. DATE OF REPORT (Vaer, Month, Day)         T       COSATI CODES       III. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         TELED GROUP       SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         S. ABSTRACT (Continue on reverse if necessary and identify by block number)         We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithms for the problem, which runs in polylog parallel time and uses 0 (n <sup>3</sup> ) processors on	Coordin	ated Scie	nce Lab	(If applicable)	Office of	of Naval R	esearch	
ADDRESS (City, Stete, and ZiP Code)         1101 W. Springfield Ave.         Urbana, IL 61801         NAME OF FUNDING/SPONSORING ORGANIZATION Joint Services         Electronics Program         ADDRESS (City, Stete, and ZiP Code)         Allington, VA 22217         PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NO0014-84-C-0149         Arlington, VA 22217         Intle (include Security Classification)         Transitive Reduction in Parallel Via Branchings         PROGRAM         PROMAL AUTHOR(S)         Cibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarian, Robert         Costant Codes         Technical         RFOOM_TIBS         Technical         Technical         RFOOM_TIBS         Technical         Technical         Technical         Technical         Technical         Technical         Technical         Technical         Technical         Supplementary NOTATION         Technical         Technical         Technical         Technical         Technical         Technical         Technical         Technical	Univers	ity of Il	linois	N/A				
1101 W. Springfield Ave. Urbana, IL 61801       Arlington, VA 22217         NAME OF FUNCION Joint Services Electronics Program       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-0149         Arlington, VA 22217       I. SOURCE OF FUNCING NUMBERS N00014-84-C-0149         Arlington, VA 22217       I. SOURCE OF FUNCING NUMBERS N0014-84-C-0149         Arlington, VA 22217       I. SOURCE OF FUNCING NUMBERS N0014-84-C-0149         Arlington, VA 22217       I. SOURCE OF FUNCING NUMBERS         Inte (include Security Classification)       Transitive Reduction in Parallel Via Branchings         Inte (include Security Classification)       Transitive Reduction in Parallel Via Branchings         PRESONAL AUTHOR(S)       I. DATE OF REPORT (Veer, Month, Day)       I. PAGE COUNT         Technical       IB. TIME COVERED       I. DATE OF REPORT (Veer, Month, Day)       I. PAGE COUNT         Artected graphs, graph algorithms, PRAM algorithms, directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.       I. Subject TERMS (Continue on reverse if necessary and identify by block number)         S. ABSTRACT (Continue on reverse if necessary and identify by block number)       We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also pr	ADDRESS (C	City, State, and	ZIP Code)		7b. ADDRESS (C	ty, State, and i	ZIP Code)	
NAME OF FUNDING/SPONSORING ORGANIZATION Joint Services       Bb. OFFICE SYMBOL (If applicable)       9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-0149         ADDRESS (City, State, and ZIP Code)       10. SOURCE OF FUNDING NUMBERS N00014-84-C-0149       10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.       WORK UNIT ROLE (Include Security Classification)         TITLE (Include Security Classification)       Transitive Reduction in Parallel Via Branchings       PROJECT       TASK NO.       WORK UNIT ROLE (Include Security Classification)         TITLE (Include Security Classification)       Transitive Reduction in Parallel Via Branchings       PROJECT       TASK NO.       WORK UNIT ROLE (Include Security Classification)         TTANSITIVE Reduction in Parallel Via Branchings       Its. State and Classification)       Tasis in Parallel Via Branchings         Immediate Reduction in Parallel Via Branchings       Its. Date OF REPORT (Year, Month, Day)       Its. PAGE COUNT October 1988       Its. Pace COUNT October 1988       Its.         A. SUPPLEMENTARY NOTATION       Its. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       Its. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         Me study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) pro- cessors on a PRAM. Our algorithm is simple and the major tool it uses is comput- ing a minimum-weight branching with zer	1101 W. Urbana,	Springfi IL 61801	eld Ave.		Arlingto	on, VA 22	217	
AMME OF FUNDING'S SOUNDARY SOUNDARY       B. OF PROFILENDED (If speliceble)       PROFILENDED (If speliceble)         ORGANIZATION Joint Services       B. OPPRESS(CR), State, and ZIP Code)       In. SOURCE OF FUNDING NUMBERS         All ington, VA 22217       PROFILENDED (If speliceble)       In. SOURCE OF FUNDING NUMBERS         TRILE (include Security Classification)       Transitive Reduction in Parallel Via Branchings       PROFILENT NO.       NO.       NO.       ACCESSION         TRILE (include Security Classification)       Transitive Reduction in Parallel Via Branchings       PROFILENT NO.       NO.       NO.       ACCESSION         Transitive Reduction in Parallel Via Branchings       III. DATE OF REPORT NO.       III. DATE OF REPORT (Year, Month, Day)       III. PROF.       PROF.         Technical       FROM       TO       October 1988       17         S. SUPPLEMENTARY NOTATION       III. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         T.       COSATI CODES       III. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         S. JUPLEMENTARY NOTATION       III. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         S. ABSTRACT (Continue on reverse if necessary and identify by block number)       IIII. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         S. ABSTRACT (Continue on reverse if necessary and ident			CORNE	IRE OFFICE SYMBOL	9 PROCUREMEN		IDENTIFICAT	TION NUMBER
Electronics Program       10. SOURCE OF FUNDING NUMBERS         ADDRESS (Chy, State, and ZiP Code)       Arlington, VA 22217       In SOURCE OF FUNDING NUMBERS         Arlington, VA 22217       PROGRAM       PROJECT       TASK       WORK UNI         ITTLE (include Security Classification)       Transitive Reduction in Parallel Via Branchings       Inclusion       Inclusion       Accession         ITTRE (include Security Classification)       Transitive Reduction in Parallel Via Branchings       Inclusion       Inclusion       Inclusion       Accession         IPRISONAL AUTHOR(S)       Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert       Tarian, Robert       Is Preference       Inclusion       Incl	NAME OF FUNDING / SPONSORING ORGANIZATION Loint Services (If applicable)				N000148	34-C-0149		
ADDRESS (City, State, and ZiP Code)       10. SOURCE OF FUNDING NUMBERS         Arlington, VA 22217       IO. SOURCE OF FUNDING NUMBERS         PROGRAM       ELMENT NO.       NO.       NO.         ITTLE (Include Security Classification)       Transitive Reduction in Parallel Via Branchings         ITTLE (Include Security Classification)       Transitive Reduction in Parallel Via Branchings         2. PERSONAL AUTHOR(S)       Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert         3a. TYPE OF REPORT       13. TIME COVERED       14. DATE OF REPORT (Year, Month, Day)       15. PAGE COUNT         Technical       FROM       TO       October 1988       17         8. SUPPLEMENTARY NOTATION       II. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       17         8. ABSTRACT (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       17         We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequentia algorithms for the problem that run in time	Electro	onics Prog	ram					
Arlington, VA 22217       PROGRAM or NO.       PROJECT NO.       NO.       WORK UN WORK UN NO.         TITLE (include Security Classification)       Transitive Reduction in Parallel Via Branchings         Intrastrive Reduction in Parallel Via Branchings         Image: Comparison of the problem is the	c ADDRESS (City, State, and ZIP Code)				10. SOURCE OF	FUNDING NUM	BERS	luces u
TITLE (include Security Classification)         Transitive Reduction in Parallel Via Branchings         2. FERSONAL AUTHOR(S)         Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert         Technical         FROM	Arlingt	TOD VA 22	217		PROGRAM	PROJECT	NO.	ACCESSIO
Transitive Reduction in Parallel Via Branchings         PERSONAL AUTHOR(S) Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert Technical 13b. TIME COVERED 14. DATE OF REPORT (Year, Month, Day) 15. PAGE COUNT Technical 16. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         V.       COSATI CODES FIELD GROUP SUB-GROUP       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         D. ABSTRACT (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         Me study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) pro- cessors on a PRAM. Our algorithm is simple and the major tool it uses is comput- ing a minimum-weight branching with zero-one weights. We also present sequen- tial algorithms for the problem that run in time O (m+n·logn).         10. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION UnclassificED/UNLIMITED						1		
Transitive Reduction in Parallel Via Branchings         I. PERSONAL AUTHOR(5) Glibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert.         Sa. TYPE OF REPORT         Technical         FROMTOOLDES         I. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         October 1988         I. COSATI CODES         I. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         I. COSATI CODES         I. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         I. COSATI CODES         I. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         9. ABSTRACT (Continue on reverse if necessary and identify by block number)         We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified	TITLE Hade	ida Familia C	accification					
2. PERSONAL AUTHOR(S) Gibbons, Phillip; Karp, Richard; Ramachandran, Vijaya; Soroker, Danny; Tarjan, Robert 33. TYPE OF REPORT       13b. TIME COVERED 13b. TIME COVERED FROM	. TITLE (Inclu	de Security C	assification)	11-1 Wie Buonah	1000	•		
Gibbons, Phillip; Karp, Richard; Ramachandran, Vijag; Soroker, Danny; Tarjah, Robert         3a. TYPE OF REPORT         Technical         13b. TIME COVERED         FROM	1. TITLE (Inclu Transit	<b>ide Security C</b> ive Reduc	<b>assification)</b> tion in Para	allel Via Branch	ings	•		
3a. TYPE OF REPORT       13b. TIME COVERED       14. DATE OF REPORT       17. Notice Continue on reverse if necessary and identify by block number)         7.       COSATI CODES       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       17         7.       COSATI CODES       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       17         8. ABSTRACT (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       19. ABSTRACT (Continue on reverse if necessary and identify by block number)         We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .         20. DISTRIBUTION / AVAILABILIT	1. TITLE (Inclu Transit 2. PERSONAL	ive Reduc	assification) tion in Para	allel Via Branch	ings			
Technical       FROM       10       Occoder 1900       11         6. SUPPLEMENTARY NOTATION       7.       COSATI CODES       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified	I. TITLE (Inclu Transit 2. PERSONAL Gibbons	ude Security C ive Reduc AUTHOR(S) s, Phillip	ssification) tion in Para ; Karp, Rich	allel Via Branch	ings an, Vijaya;	Soroker, D	anny; Ta	rjan, Robert
S. SUPPLEMENTARY NOTATION         7.       COSATI CODES         FIELD       GROUP         SUBJECT TERMS (Continue on reverse if necessary and identify by block number)         directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         9. ABSTRACT (Continue on reverse if necessary and identify by block number)         We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified	. TITLE (Inclu Transit . PERSONAL Gibbons . TYPE OF	ude Security C ive Reduc AUTHOR(S) s, Phillip REPORT	ssification) tion in Para ; Karp, Rich 13b. TIME	allel Via Branch: nard; Ramachandra COVERED	ings an, Vijaya; 14. DATE OF REP October	Soroker, D ORT (Year, Mon 1988	anny; Ta nth, Dey)	rjan, Robert 5. PAGE COUNT 17
7.       COSATI CODES       18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) directed graphs, graph algorithms, PRAM algorithms, directed spanning trees.         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n logn).         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION/AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified	Transit PERSONAL Gibbons 3. TYPE OF Technic	AUTHOR(S) s, Phillip REPORT cal	ssification) tion in Para ; Karp, Rich 13b. TIME FROM	allel Via Branch nard; Ramachandra COVERED TO	ings an, Vijaya; 14. DATE OF REP October	Soroker, D ORT (Year, Mo 1988	anny; Ta nth, Day) 1	rjan, Robert 5. PAGE COUNT 17
7.       COSATI CODES         FIELD       GROUP         SUB-GROUP       SUB-GROUP         Image: Sub-GROUP       Sub-GROUP	. TITLE (Incl. Transit . PERSONAL Gibbons 3a. TYPE OF Technic 5. SUPPLEME	AUTHOR(S) AUTHOR(S) S, Phillip REPORT Cal NTARY NOTAT	tion in Para ; Karp, Rich 13b. TIME FROM	allel Via Branch nard; Ramachandra COVERED TO	ings an, Vijaya; 14. DATE OF REP October	Soroker, D ORT (Year, Mon 1988	anny; Ta nth, Day) 1	rjan, Robert 5. PAGE COUNT 17
FIELD       GROUP       SUB-GROUP       directed spanning trees.         9. ABSTRACT (Continue on reverse if necessary and identify by block number)       We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n <sup>3</sup> ) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n·logn).         20. DISTRIBUTION / AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified         20. DISTRIBUTION / AVAILABILITY OF ABSTRACT       21. ABSTRACT SECURITY CLASSIFICATION Unclassified	. TITLE (Inclu Transit . PERSONAL Gibbons 3a. TYPE OF Technic 5. SUPPLEME	ade Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal INTARY NOTAT	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON	allel Via Branch nard; Ramachandra COVERED TO	ings an, Vijaya; 14. DATE OF REP October	Soroker, D ORT (Year, Mon 1988	anny; Ta nth, Oay)	rjan, Robert 5. PAGE COUNT 17
<ul> <li>9. ABSTRACT (Continue on reverse if necessary and identify by block number)</li> <li>9. ABSTRACT (Continue on reverse if necessary and identify by block number)</li> <li>We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n<sup>3</sup>) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n·logn).</li> <li>20. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION Unclassified</li> <li>22. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>23. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION Unclassified</li> </ul>	. TITLE (Incl. Transit . PERSONAL Gibbons 3a. TYPE OF Technic 5. SUPPLEME 7.	AUTHOR(S) AUTHOR(S) S, Phillip REPORT Cal INTARY NOTAT	assification) tion in Para ; Karp, Rich 13b. TIME FROM TION	allel Via Branch nard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gi	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm	anny; Ta nth, Day) 1 and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
<ul> <li>B. ABSTRACT (Continue on reverse if necessary and identify by block number)</li> <li>We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n<sup>3</sup>) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n·logn).</li> <li>20. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION Unclassified</li> <li>22. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>23. DTIC USERS</li> </ul>	Transit . PERSONAL Gibbons . TYPE OF Technic . SUPPLEME	ade Security Cl ive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP	allel Via Branch: nard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gi directed gi	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees	Soroker, D ORT (Year, Mon 1988 nse if necessary algorithm 5.	anny; Ta nth, Day) and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
<ul> <li>We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses O (n<sup>3</sup>) processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n logn).</li> <li>10. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION Unclassified</li> </ul>	. TITLE (Incl. Transit Gibbons La. TYPE OF Technic S. SUPPLEME 7. FIELD	AUTHOR(S) a, Phillip REPORT cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TION CODES SUB-GROUP	allel Via Branch: nard; Ramachandra COVERED TO	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees	Soroker, D ORT (Year, Mon 1988 1988 rse if necessary algorithm s.	anny; Ta nth, Oay) 1 and identify s, PRAM	s. PAGE COUNT 17 y by block number) algorithms,
We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .	TITLE (Inclu Transit Gibbons I. TYPE OF Technic S. SUPPLEME FIELD	AUTHOR(S) AUTHOR(S) S, Phillip REPORT Cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TION CODES SUB-GROUP	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gi directed gi directed gi	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number)	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm S.	anny; Ta nth, Day) 1 and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ .	. TITLE (Incl. Transit . PERSONAL Gibbons Ia. TYPE OF Technic 5. SUPPLEME 7. FIELD ). ABSTRACT	ude Security Cl ive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TION CODES SUB-GROUP	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed g: directed g: directed si directed si	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number)	Soroker, D ORT (Year, Mon 1988 nse if necessary algorithm s.	anny; Ta nth, Day) 1 and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ . 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [] DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22b TELEPHONE (Incluster Area Code) [22c, OFFICE SYMBOL]	TITLE (Inclu Transit PERSONAL Gibbons I. TYPE OF Technic S. SUPPLEME FIELD	ude Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed g: directed si directed si ry and identify by block	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number)	Soroker, D ORT (Year, Mod 1988 1988 rse if necessary algorithm 5.	anny; Ta nth, Day) and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
we study the following processing subgraph of it. Our main result is a parallel minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ . 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT [] DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified [] 22c. OFFICE SYMBOL	TITLE (Inclu Transit PERSONAL Gibbons a. TYPE OF Technic S. SUPPLEME FIELD	ude Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal INTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed gr directed sp ry and identify by block	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number)	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s.	anny; Ta nth, Day) 1 and identify s, PRAM	s. PAGE COUNT 17 y by block number) algorithms,
minimal strongly connected spanning subgraph of the structure algorithm for this problem, which runs in polylog parallel time and uses $O(n^3)$ processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time $O(m+n \cdot \log n)$ . The construction / Availability of ABSTRACT [] DTIC USERS [] ABSTRACT SECURITY CLASSIFICATION Unclassified [] 22c. OFFICE SYMBOL	TITLE (Incl. Transit Gibbons a. TYPE OF Technic SUPPLEME FIELD	ude Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal INTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TION CODES SUB-GROUP reverse if necessar	Allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gi directed gi directed gi ry and identify by block	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number) a strongly comm	Soroker, D ORT (Year, Mon 1988 ree if necessary algorithm s.	anny; Ta nth, Day) 1 and identify s, PRAM	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
algorithm for this problem, which runs in polylog parallel time and uses o (a) y pro- cessors on a PRAM. Our algorithm is simple and the major tool it uses is comput- ing a minimum-weight branching with zero-one weights. We also present sequen- tial algorithms for the problem that run in time O (m+n·logn). 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT [] DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22b TELEPHONE (Include Area Code) [ 22c, OFFICE SYMBOL	TITLE (Incl. Transit PERSONAL Gibbons Technic SUPPLEME FIELD	ude Security Cl ive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP reverse if necessar	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed gr directed sp ry and identify by block	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number) a strongly comm	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s. ected digrap nain result is	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
<ul> <li>cessors on a PRAM. Our algorithm is simple and the major tool it uses is comparing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n·logn).</li> <li>DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION Unclassified</li> <li>22. TELEBHOME (include Area Code)   22c. OFFICE SYMBOL</li> </ul>	TITLE (Inclu Transit PERSONAL Gibbons Technic SUPPLEME FIELD	we stu minima	assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP reverse if necessar dy the followin 1 strongly conner	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed sp ry and identify by block ng problem: given a ected spanning subgra	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number) a strongly comm aph of it. Our m	Soroker, D ORT (Year, Mod 1988 rse if necessary algorithm s. ected digrap nain result is he and uses (	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro-	rjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
<ul> <li>ing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time O (m+n·logn).</li> <li>20. DISTRIBUTION / AVAILABILITY OF ABSTRACT</li> <li>21. ABSTRACT SECURITY CLASSIFICATION</li> <li>Unclassified</li> <li>22. DTIC USERS</li> <li>23. TELEPHONE (include Area Code)   22c. OFFICE SYMBOL</li> </ul>	TITLE (Incl. Transit Gibbons A. TYPE OF Technic S. SUPPLEME FIELD	ude Security Cl ive Reduct AUTHOR(S) s, Phillip REPORT cal INTARY NOTAT GROUP (Continue on We stu minima algorith	assification) tion in Para ; Karp, Rich 13b. TIME FROM FROM TON CODES SUB-GROUP reverse if necessa dy the followin 1 strongly conne m for this probl	Allel Via Branch: hard; Ramachandra COVERED TO TO 18. SUBJECT TERMS directed gr directed gr directed sp ry and identify by block ng problem: given a ected spanning subgra lem, which runs in po	ings an, Vijaya; 14. DATE OF REPOCTOBER October (Continue on rever raphs, graph panning trees number) a strongly conn aph of it. Our m lylog parallel tim	Soroker, D ORT (Year, Mod 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro-	y by block number) algorithms,
tial algorithms for the problem that run in time $O(m+n \cdot \log n)$ . 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22. DTIC USERS	TITLE (Inclu Transit Gibbons La. TYPE OF Technic S. SUPPLEME FIELD	ude Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal INTARY NOTAT COSATI GROUP (Continue on We stu minima algorith cessors	assification) tion in Para ; Karp, Rich 13b. TIME FROM	Allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed gr directed gr ry and identify by block ng problem: given a ected spanning subgra lem, which runs in po pur algorithm is simpl	ings an, Vijaya; 14. DATE OF REP October (Continue on reve raphs, graph panning trees number) a strongly comm aph of it. Our m lylog parallel tim e and the major	Soroker, D ORT (Year, Mon 1988 realignerithm algorithm s. ected digrap nain result is ne and uses C tool it uses i	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput-	y by block number) algorithms,
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22. DTIC USERS 23. DTIC USERS 23. DTIC USERS 24. ABSTRACT SECURITY CLASSIFICATION Unclassified	. TITLE (Inclu Transit Gibbons Ba. TYPE OF Technic 5. SUPPLEME 7. FIELD 9. ABSTRACT	ude Security Cl ive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP (Continue on We stu minima algorith cessors ing a m	assification) tion in Para ; Karp, Rich 13b. TIME FROM	allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed gr directed spanning subgra lected spanning subgra lem, which runs in po bur algorithm is simple branching with zero	ings an, Vijaya; 14. DATE OF REP October (Continue on rever raphs, graph panning trees number) a strongly comm aph of it. Our m lylog parallel tim e and the major -one weights. W	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen-	y by block number) algorithms,
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS SAME AS RPT. DTIC USERS SAME AS RPT. DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22. DEFICE SYMBOL	TITLE (Inclu Transit Gibbons La TYPE OF Technic S. SUPPLEME FIELD	we stuminima algorith cessors ing a m tial algore	Assification) tion in Para ; Karp, Rich 13b. TIME FROM TON CODES SUB-GROUP reverse if necessar dy the followin 1 strongly conner m for this probl on a PRAM. O inimum-weight prithms for the p	Allel Via Branch: hard; Ramachandra COVERED 	ings an, Vijaya; 14. DATE OF REP October (Continue on rever raphs, graph panning trees number) a strongly conn aph of it. Our m lylog parallel tim e and the major -one weights. We ne O (m+n logn	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen ).	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen-	trjan, Robert 5. PAGE COUNT 17 y by block number) algorithms,
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION Unclassified 22b TELEBHONE (Include Area Code) 22c. OFFICE SYMBOL	TITLE (Inclu Transit Gibbons Ja. TYPE OF Technic 5. SUPPLEME 7. FIELD	we stuminima algorith cessors ing a m tial algo	Assification) tion in Para ; Karp, Rich 13b. TIME FROM FROM TON CODES SUB-GROUP reverse if necessan dy the following reverse if necessan dy the following in strongly connect am for this proble on a PRAM. Out inimum-weight prithms for the p	Allel Via Branch: hard; Ramachandra COVERED TO TO 18. SUBJECT TERMS directed gr directed gr directed sp ry and identify by block ng problem: given a ected spanning subgra lem, which runs in po bur algorithm is simple branching with zero- problem that run in time	ings an, Vijaya; 14. DATE OF REPORT October (Continue on rever raphs, graph panning trees number) a strongly conn aph of it. Our m lylog parallel time e and the major one weights. We ne O (m+n logn	Soroker, D ORT (Year, Mod 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen ).	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen-	y by block number) algorithms,
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS Unclassified 21. ABSTRACT SECONTY CLASSIFICATION Unclassified 22. DFFICE SYMBOL	TITLE (Inclu Transit Gibbons La. TYPE OF Technic S. SUPPLEME 7. FIELD	ude Security Cl tive Reduct AUTHOR(S) s, Phillip REPORT cal NTARY NOTAT COSATI GROUP (Continue on We stu minima algorith cessors ing a m tial algo	Assification) tion in Para ; Karp, Rich 13b. TIME FROM FROM TON CODES SUB-GROUP reverse if necessar dy the following n for this probl on a PRAM. O tinimum-weight prithms for the p	allel Via Branch: hard; Ramachandra COVERED TO	ings an, Vijaya; 14. DATE OF REPOCTOBET (Continue on rever raphs, graph panning trees number) a strongly conn aph of it. Our m lylog parallel tim e and the major -one weights. Wane O (m+n logn	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen ).	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen-	y by block number) algorithms,
INCLASSIFIED/UNLIMITED I SAME AS RPT. DTIC USERS UNCLASSIFIED (Declassified Area Code) 22c. OFFICE SYMBOL	TITLE (Inclu Transit Gibbons 3a. TYPE OF Technic 5. SUPPLEME 7. FIELD 9. ABSTRACT	AUTHOR(S) a, Phillip REPORT cal NTARY NOTAT COSATI GROUP (Continue on We stu minima algorith cessors ing a m tial algo	assification) tion in Para ; Karp, Rich 13b. TIME FROM ION CODES SUB-GROUP reverse if necessa dy the followin 1 strongly conne m for this probl on a PRAM. O inimum-weight orithms for the p	Allel Via Branch: hard; Ramachandra COVERED TO TO TO TO TO TO TO TO TO TO	ings an, Vijaya; 14. DATE OF REPORT October (Continue on rever raphs, graph panning trees number) a strongly connu- aph of it. Our m lylog parallel tim e and the major -one weights. We ne O (m+n·logn	Soroker, D ORT (Year, Mod 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen ).	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen-	y by block number) algorithms,
	TITLE (Inclu Transit Gibbons Ja. TYPE OF Technic 5. SUPPLEME 7. FIELD 9. ABSTRACT	AUTHOR(S) a, Phillip REPORT cal NTARY NOTAT COSATI GROUP (Continue on We stu minima algorith cessors ing a m tial algo	Assification) tion in Para ; Karp, Rich 13b. TIME FROM ISD. TIME FRO	Allel Via Branch: hard; Ramachandra COVERED TO 18. SUBJECT TERMS directed gr directed gr directed gr directed sp ry and identify by block ng problem: given a ected spanning subgra lem, which runs in po bur algorithm is simple branching with zero problem that run in time	ings an, Vijaya; 14. DATE OF REPOCTOBER (Continue on rever raphs, graph panning trees number) a strongly connu aph of it. Our m lylog parallel tim e and the major -one weights. We ne O (m+n·logn	Soroker, D ORT (Year, Mon 1988 rse if necessary algorithm s. ected digrap nain result is ne and uses C tool it uses i /e also presen ).	anny; Ta nth, Day) 1 and identify s, PRAM h, find a a parallel (n <sup>3</sup> ) pro- s comput- nt sequen- SIFICATION	y by block number) algorithms,

UNCLASSIFIED

# TRANSITIVE REDUCTION IN PARALLEL VIA BRANCHINGS

Phillip Gibbons<sup>\*</sup> Richard Karp<sup>\*,†</sup> Computer Science Division, University of California, Berkeley, CA Vijaya Ramachandran<sup>\*,‡</sup> Coordinated Science Lab., University of Illinois, Urbana, IL Danny Soroker<sup>\*</sup> IBM Almaden Research Center, San Jose, CA Robert Tarjan<sup>§</sup> Computer Science Dept., Princeton University, Princeton, NJ and AT&T Bell Labs., Murray Hill, NJ

July 15, 1988

#### ABSTRACT

We study the following problem: given a strongly connected digraph, find a minimal strongly connected spanning subgraph of it. Our main result is a parallel algorithm for this problem, which runs in polylog parallel time and uses  $O(n^3)$  processors on a PRAM. Our algorithm is simple and the major tool it uses is computing a minimum-weight branching with zero-one weights. We also present sequential algorithms for the problem that run in time  $O(m+n \cdot \log n)$ .

Supported in part by the International Computer Science Institute, Berkeley, California.

Also supported by NSF grant CCR-8411954.

Also supported by Joint Services Electronics Program under N00014-84-C-0149.

Supported in part by NSF grant DCR-8605962 and ONR contract N00014-87-K-0467.

# 1. Introduction

The transitive reduction problem for strongly connected digraphs is: given a strongly connected digraph G, find a minimal strongly connected spanning subgraph of it, i.e., a strongly connected spanning subgraph for which the removal of any arc destroys strong connectivity. We are looking for a minimal subgraph because the problem of finding a *minimum* subgraph with the same transitive closure is NP-hard [GJ].

There is an obvious sequential algorithm for solving this problem: scan the arcs one by one; at step *i* test if the *i*-th arc can be removed without destroying strong connectivity. If so, remove it and update the digraph. This algorithm has complexity  $O((n+m)^2)$ , where *n* is the number of vertices of the input graph and *m* is the number of arcs. A simple modification is to initially reduce the number of arcs to at most 2n-2 by taking the union of a forward and an inverse branching (defined below). This reduces the running time to  $O(n^2)$ .

The problem studied here is reminiscent of the well-studied problem of finding a maximal independent set of vertices in a graph, for which several parallel algorithms have appeared in the literature ([KW],[Lu],[ABI],[GS]). Two common features are that there is a simple sequential algorithm for it that seems hard to parallelize and that the related optimization problem (minimum vs. minimal) is *NP*-hard.

We can define the following independence relation on the arcs of a strongly connected digraph, G: a set of arcs is independent if it can be removed without destroying strong connectivity of G. Using this definition, finding a transitive reduction of G is equivalent to removing a maximal independent sets of arcs from G. A property that sets our problem apart from the maximal independent set problem is that in our case independence of a set is not guaranteed when every pair of elements in it is independent.

Our problem can be expressed as the determination of a maximal independent set in an independence system as defined by Karp, Upfal and Wigderson ([KUW]). The problem computed by a "rank oracle" in this case is NP -hard, but an "independence oracle" is easy to compute in NC. Following the method described in [KUW] this automatically yields a randomized parallel algorithm that uses a polynomial number of processors and runs in time  $O(\sqrt{n} \cdot \log^c n)$  (for some constant c).

In this paper we present parallel and sequential algorithms for this problem. Our first parallel algorithm runs in time  $O(\log^5 n)$  and uses  $O(n^3)$  processors on a CREW PRAM. We then present an improved implementation of one of the steps in the algorithm that leads to a parallel algorithm that runs in  $O(\log^4 n)$  time with the same processor bound. Both of these algorithms can be speeded up by a log n factor if we use a CRCW PRAM; we assume here the COMMON concurrent-write model in which all processors participating in a concurrent write must write the same value [KR]. The processor bound of  $O(n^3)$  represents the number of processors needed to multiply two *n* by *n* matrices in  $O(\log n)$  time on a CREW PRAM by the straightforward parallel matrix multiplication algorithm. It is possible that the processor bound can be improved by using sophisticated techniques for multiplying *n* by *n* matrices (see e.g., [CW]); we do not elaborate on this.

The major tool that our algorithms use is computing a minimum-weight branching with zero-one weights. Central to our algorithms is a proof that two suitable applications of this tool are guaranteed to reduce by half the number of arcs still to be removed. We also present two sequential algorithms for the problem, each of which runs in time  $O(m+n \cdot \log n)$ . This is an improvement over the straightforward algorithm mentioned above.

The transitive reduction problem is, in some sense, a dual of the minimum strong augmentation problem - add a minimum set of arcs to a digraph to make it strongly connected. A linear time sequential algorithm was given for this problem by Eswaran and Tarjan ([ET]), and a parallel algorithm running in  $O(\log n)$  time with  $O(n^3)$  processors on a CRCW PRAM was given by Soroker ([So]).

Our problem extends naturally to general digraphs: given a digraph G, find a minimal spanning subgraph of it whose transitive closure is the same as that of G. A sequential algorithm for this problem in the case that G is acyclic is given in [AGU] and can be parallelized in a straightforward manner. Combining it with our algorithms we obtain parallel algorithms (with the same complexities as stated above) for the transitive reduction problem on general digraphs. We point out that these parallel algorithms are good with respect to the state of the art, since the problem solved is at least as hard as testing reachability from one vertex to another in a digraph, and the best NC algorithm currently known for this requires on the order of M(n) processors, where M(n) is the number of processors needed to multiply two n by n Boolean matrices in  $O(\log n)$  time.

We note that the name "transitive reduction" was given to a different problem by Aho, Garey and Ullman ([AGU]), and should not be confused with our definition. Given a digraph G, they ask for a digraph with a minimum number of arcs (not necessarily a subgraph of G) whose transitive closure is the same as that of G. This definition agrees with ours when G is acyclic.

#### Definitions

Let G be a strongly connected digraph. A <u>forward (inverse) branching</u> rooted at x is a spanning tree of G in which x has in-degree (out-degree) zero and all other vertices have indegree (out-degree) one. A <u>branching</u> is either a forward or an inverse branching. Throughout this paper the root, x, will be some (arbitrarily) fixed vertex of the input digraph, and the set of all

branchings will be taken to be only those rooted at x.

An arc, e, is <u>G-redundant</u> (or simply <u>redundant</u> when the graph is clear) if  $G-\{e\}$  is strongly connected. Arc e is <u>G-essential</u> (or <u>essential</u>) if it is not redundant. Let H be a subgraph of G. Let  $\underline{r_G(H)}$  denote the number of G-redundant arcs in H. When H=G we will use the shorthand  $\underline{r(G)}$ .

An <u>*H*-philic</u> (<u>*H*-phobic</u>) branching in G is one that has the greatest (smallest) number of arcs in common with H over all branchings (rooted at x) in G.

Our model of parallel computation is the Parallel Random Access Machine (PRAM), which consists of a collection of independent processing elements communicating through a shared memory. For a survey on the PRAM model and PRAM algorithms see [KR].

#### 2. The Transitive Reduction Algorithm

Our basic algorithm is based solely on computing philic and phobic branchings. The following lemma explains how these branchings are computed:

Lemma 0: An H-philic (H-phobic) branching can be computed by a minimum-weight branching computation with zero-one weights.

**Proof:** Assign weight 0(1) to every arc in H and weight 1(0) to all other arcs. []

Such a minimum-weight branching can be computed in time  $O(\log^2 n)$  using  $O(n^3)$  processors on a CRCW PRAM by Lovasz's method ([Lo]). On a CREW PRAM, this algorithm runs in  $O(\log^3 n)$  time.

**Proposition 1:** An arc of G is essential if and only if it is the unique arc crossing some directed cut of G.

**Proposition 2:** The union of a forward branching and an inverse branching of G is a strongly connected spanning subgraph of G.

**Proposition 3:** Let G' be a strongly connected spanning subgraph of G. Then e is G'-redundant only if it is G-redundant.

Lemma 1: Let F be a forward branching in G and let I be an F-philic inverse branching in G. Let  $G' = F \cup I$ . Then the arcs of I - F are all G'-essential. **Proof:** Let  $e \in I - F$ . Assume  $G' - \{e\}$  contains some inverse branching, I'. Then I' has one more arc in common with F than I does (since all branchings have the same number of arcs). But this contradicts the fact that I is F-philic. Thus  $G' - \{e\}$  contains no inverse branching and is therefore not strongly connected. []

A <u>cut leaving S</u> is the set of arcs extending from S to V(G)-S in a digraph, G, and its cardinality is denoted by  $\underline{\delta_G(S)}$ .

Theorem 1 (Edmonds' Branching Theorem ([Ed])): Let

 $k = \min \{ \delta_G(S) \mid x \in S, S \neq V(G) \}.$ 

Then G contains k arc-disjoint forward branchings (rooted at x).

Lemma 2: For every strongly connected digraph, G, there exists a forward branching, F, of G such that  $r_G(F) \leq \frac{1}{2}r(G)$ .

**Proof:** Let G' be obtained from G by duplicating all essential arcs. Let S be a proper subset of V(G) containing x. We claim that  $\delta_{G'}(S) \ge 2$ . This is because the cut leaving S must contain at least one duplicated essential arc of G or at least two redundant arcs (by proposition 1). Therefore, by theorem 1, there are two arc-disjoint forward branchings in G' (each corresponding to a branching in G), one of which must contain at most half of the (unduplicated) G-redundant arcs.[]

**Theorem 2:** Let R be the set of redundant arcs in G. Let F be an R-phobic forward branching and let I be an F-philic inverse branching. Let  $G' = F \bigcup I$ . Then  $r(G') \leq \frac{1}{2}r(G)$ .

**Proof:** First note that by proposition 2, G' is strongly connected. By lemma 2 and proposition 3,  $r_{G'}(F) \leq r_G(F) \leq \frac{1}{2}r(G)$ . By lemma 1,  $r(G') = r_{G'}(F)$ . Therefore  $r(G') \leq \frac{1}{2}r(G)$ . []

It is an immediate consequence of theorem 2 that the following NC algorithm gives a transitive reduction of G:

Repeat

(1)  $R \leftarrow$  set of redundant arcs in G

(2)  $F \leftarrow R$ -phobic forward branching in G

(3)  $I \leftarrow F$ -philic inverse branching in G

$$(4) G \leftarrow F \cup I$$

until  $R = \phi$ 

(5) output G (it is a transitive reduction of the input digraph)

By Theorem 2 the repeat loop runs  $O(\log n)$  times, where n is the number of vertices in G. Steps (2) and (3) are implemented with Lovasz's minimum-weight branching algorithm (lemma 0). The straightforward implementation of step (1) is to perform a strong connectivity test (transitive closure) with each vertex of the graph deleted in turn, which requires  $n \cdot M(n)$  processors. In the next section we shall show how to perform this step more efficiently.

# 3. Efficient Classification of Arcs

In this section we give parallel algorithms to classify the arcs of G as essential or redundant in poly-log time using only  $O(n^3)$  processors. In section 3.1 we provide a simple polylog time parallel algorithm using  $O(n^3)$  processors. In section 3.2 we provide a faster algorithm using tree contraction [MR].

# 3.1. Finding Redundant Arcs Using Minimum Weight Branchings

Let  $\underline{E_f}$  ( $\underline{E_i}$ ) be the set of essential arcs contained in all forward (inverse) branchings. It follows from proposition 2 that:

**Proposition 4:** An arc is essential if it is either in  $E_f$  or in  $E_i$  (or both).

Lemma 3: Let H be a set of arcs containing  $E_f$  and let F be an H-phobic forward branching in G. Then  $|(F \cap H) - E_f| \le \frac{1}{2} |H - E_f|$ .

**Proof:** Let G' be obtained from G by duplicating all the arcs in  $E_f$ . As in lemma 2, there exist two arc-disjoint forward branchings in G' (corresponding to branchings in G), one of which contains at most half the arcs of  $H-E_f$ . []

Therefore  $E_f$  (and similarly  $E_i$ ) can be computed by the following algorithm:

(1)  $H \leftarrow G$ 

repeat steps (2) and (3)  $\lceil lg m \rceil$  times

(2)  $F \leftarrow H$ -phobic forward branching in G

(3)  $H \leftarrow H \cap F$ 

(4) output H (this is the set  $E_f$ )

This algorithm requires  $\log n$  applications of Lovasz's minimum weight branching algorithm, which runs in  $O(\log^2 n)$  parallel time on a CRCW PRAM with  $O(n^3)$  processors. Thus we can use this algorithm to find all redundant arcs in  $O(\log^3 n)$  parallel time on a CRCW PRAM with  $O(n^3)$  processors. This in turn leads to a transitive reduction algorithm that runs in  $O(\log^4 n)$  parallel time on a CRCW PRAM with  $O(n^3)$  processors.

## 3.2. Finding Redundant Arcs Using Tree Contraction

Let r be a fixed root of a directed graph G = (V, E). We call arc (v, w) an <u>out-bridge</u> if (v, w) is on every path from r to w, and an <u>in-bridge</u> if (v, w) is on every path from v to r. Let O be the set of out-bridges of G, and I the set of in-bridges of G. Then the set of redundant arcs is the set  $E - (I \cup O)$ .

Let B be a forward branching rooted at r. Then every out-bridge of G lies in B. We can view B as a rooted directed tree B = (V, E', r). For a vertex v in  $V - \{r\}$ , we denote by parent (v), the parent of v in B. A vertex v is <u>active</u> if there is a path from r to v that avoids arc (parent (v), v). Similarly, a non-tree arc (w, v) is <u>active</u> if it lies on a path from r to v that avoids arc (parent (v), v).

Lemma 4: Let B = (V, E', r) be a forward branching in a directed graph G. A tree arc e = (parent(v), v) in B is an out-bridge of G if and only if v is not active.

**Proof:** If e is an out-bridge of G then every path from r to v passes through e. Thus v cannot be active. Conversely, if e is not an out-bridge, then there exists a path from r to v that avoids e and hence v must be active.[]

We now give an algorithm to identify all active vertices, and hence all out-bridges, using <u>tree contraction</u> [MR]. An analogous computation on an inverse branching rooted at r gives the in-bridges, from which we can compute the redundant arcs in G.

We shall use a variant of tree contraction proposed in [Ra] in which the basic operation is <u>shrink</u>, which we now define. A <u>leaf chain</u> in a rooted tree T=(V,E,r) is a path  $\langle v_1, \dots, v_l \rangle$  such that each  $v_i, i > 1$  has exactly one incoming arc and one outgoing arc in T,  $v_1$  has either no

incoming arc or more than one outgoing arc in T, and  $v_l$  is a leaf in T. We will call  $v_1$  the <u>root</u>, and  $v_l$  the <u>leaf</u> of the leaf chain. Note that every leaf in T is part of a leaf chain, possibly a degenerate one (if l=2).

The shrink operation applied to a rooted tree T = (V, E, r) removes all vertices in each leaf chain in T except the root of the leaf chain. It can be shown that  $O(\log n)$  applications of the shrink operation suffice to reduce any *n*-node tree to a single node [Ra].

We now develop an algorithm <u>Shrink(P)</u> for identifying out-bridges for the case when the forward branching is a simple path. We shall then use this to find the out-bridges in leaf chains while implementing the shrink operation in a tree contraction algorithm to find out-bridges in G given an arbitrary forward branching.

The input to algorithm Shrink(P) is a directed graph P = (V, E) consisting of a directed path  $p = <1,2, \dots, t>$ , together with a collection of <u>forward arcs</u> of the form (i,j), i < j, and a collection of <u>back arcs</u> of the form (i,j), i > j. The algorithm Shrink(P) will identify all active vertices, thereby giving the out-bridges in p. Note that P is allowed to have two arcs of the form (i,i+1), one of which is a forward arc and the other lies in p. We will need this when we apply algorithm Shrink(P) to the general problem of finding out-bridges in a graph with an arbitrary forward branching.

We now make a series of observations.

Observation 1: Every forward arc is active.

Let p(u) be the subgraph of G induced by vertices u through t. For each vertex v in p(u), let  $v \rightarrow u$  if u is reachable from v in p(u). Let reach(u) be the set of vertices v in p(u) with  $v \rightarrow u$ .

**Observation 2:** Reach (u) is a single interval of the form [u, u']. Further a vertex  $v \neq u$  is in reach (u) if and only if there exists a sequence of back arcs  $b_i = (u_i, v_i), i = 1, \dots, k$  such that  $v_1 = u$ ,  $u_k \geq v$ , and  $u_i \geq v_{i+1}, i = 1, \dots, k-1$ .

Lemma 5: A vertex u is active if and only if there is a forward arc (k,l) with k < u and l in reach (u).

**Proof:** Let u be an active vertex. Then there is a path q from the root to u that avoids arc (u-1,u). This in turn implies that q must contain a forward arc f = (k,l) with  $k < u, l \ge u$  and with u reachable from l using only arcs in p(u). Hence l must be in reach (u).

Conversely suppose there is a forward arc f = (k, l) with k < u and l in reach(u). Hence there is a path q from l to u using only arcs in p(u). Then the path consisting of arcs in p from the root to k, followed by arc f and then the path q is a path from 1 to u that avoids arc (u-1,u). Hence u must be an active vertex.[] Observations 1 and 2 and Lemma 5 together give us the following algorithm to find all outbridges when the forward branching is a simple path.

#### Shrink(P);

1. Find reach (u) for each vertex u as follows:

a) For each back arc b = (i, j) find a back arc next(b) = (i'j') with j' in [j, i] and maximum i'. If  $i' \le i$  then set  $next(b) = \phi$ .

b) Form an auxiliary graph with a vertex for each back arc b and an arc from b to next (b), if next (b) exists. This auxiliary graph is a forest of trees.

c) For each vertex u, pick some back arc b = (v, u) incident on u, and find the root b' of the tree it belongs to. Let b' be the back arc (x, y). Set reach (u) = [u, x].

If there is no back arc incident on u set reach (u)=[u,u].

2. For each vertex u, find a forward arc f = (k, l) with l in reach(u) and with minimum k. If k < u mark u as active.

3. For each vertex u that is not active, mark (p(u),u) as an out-bridge.

We now show how to implement each of the steps in the algorithm efficiently in parallel. Step 3 can be implemented trivially in constant time with t processors. The following method implements step 2 in  $O(\log t)$  time with a number of processors linear in the size of P: Initially we determine, for each vertex u, the forward arc (v, u) with minimum v (if such an arc exists). It is straightforward to compute this in  $O(\log t)$  time with a linear number of processors. Then by a doubling computation we compute, for each interval  $[u, u+2^j], 1 \le j \le \lceil \log t \rceil, 1 \le u \le t-2^j$ , the forward arc (v, x) with minimum v such that x is in the interval  $[u, u+2^j]$ . This computation can be done in  $O(\log t)$  time with a linear number of processors on a CREW PRAM. Any interval  $[i, j], 1 \le i < j \le t$  can be written as the overlapping union of two of the previously computed intervals, and hence each vertex can now find a forward arc as required in step 2 in constant time.

Step 1 can be implemented in  $O(\log t)$  time with a linear number of processors on a CREW PRAM as follows. Step 1a can be performed in a manner analogous to step 2. Step 1b can be implemented in constant time with a linear number of processors. Step 1c can be implemented by pointer jumping in  $O(\log t)$  time with a linear number of processors. Thus we have a parallel algorithm for Shrink(P) that runs in  $O(\log t)$  time with a linear number of processors on a CREW PRAM.

We now incorporate the Shrink algorithm in the following tree contraction algorithm that finds the out-bridges in an arbitrary forward branching of a directed graph G rooted at r. The algorithm constructs a sequence of pairs  $(G_k, T_k)$ , where  $G_k$  is a digraph and  $T_k$  is a forward branching;  $G_1$  is the input digraph and  $T_1$  is a forward branching of G rooted at a fixed vertex r. Iteration k identifies the leaf chains of  $T_k$ , determines the out-bridges of  $G_k$  within those leaf chains, deletes all the vertices of the leaf chains except their roots, and then performs a transitive closure computation and adds appropriate arcs to ensure that the out-bridges in  $G_{k+1}$  are precisely the out-bridges of G not yet identified.

Outbridges(G=(V,E,r),T);

Input: A directed graph G = (V, E) with a forward branching T rooted at r; |V| = n.

#### Repeat

1. Find out-bridges in the leaf chains of T:

For each leaf chain l in T pardo

Let t be the root of l and t' the leaf of l. Let L' be the subgraph of G induced by vertices in l.

a) Form L from L' by introducing a forward arc (t,y) for each non-tree arc (x,y) in G with y in  $V(l) - \{t\}$  and x not in V(l).

b) Apply Shrink(L) to find the out-bridges in L and label these as out-bridges of G.

2. Remove leaf chains from T:

a) Form the graph H with vertex set V and arc set the arcs in all leaf chains and all non-tree arcs of G.

b) Form M, the adjacency matrix of H, and form the transitive closure  $M^*$  of M.

c) For each vertex v, determine, using  $M^*$ , the set of vertices from which v is reachable in H. For each such vertex w, introduce an arc (w, v) in G.

d) For each vertex t that is the head of some leaf chain, delete all incoming non-tree arcs to proper descendants of t. Collapse all of these proper descendants into t. Delete any self-loops in this graph.

until  $T = \phi$ 

Generalizing our earlier notation for the case when the forward branching is a simple path, we now let p(u) be the subgraph of G induced by those vertices that lie in the subtree of T rooted at u. For each vertex v in p(u), let  $v \rightarrow u$  if u is reachable from v in p(u). Let reach(u) be the set of vertices v in p(u) with  $v \rightarrow u$ . The following lemma is a straightforward generalization of Observation 2 and Lemma 5 (here a vertex v is a *descendant* of a vertex u if u=v or if there is a directed path from u to v in T; otherwise v is a *non-descendant* of u).

Lemma 6: A vertex u in G is active if and only if there is an arc (x,y) with x a non-descendant of u and with y in reach (u).

Let G be a directed graph with a forward branching T rooted at r, and let v be a vertex in G. An <u>active path</u> to v is a path p from r to v consisting of an initial path p' using tree arcs from r to a non-descendant x of v followed by an intermediate path consisting of a single non-tree arc a from x to a descendant y of v followed by a final path p'' from y to v using only arcs connecting descendants of v.

Observation 3: Vertex v is active if and only if there is an active path to v.

We now prove some lemmas that will allow us to establish the correctness of algorithm Outbridges. As before let  $G_i$  and  $T_i$  be the graph and forward branching present at the start of the *i* th iteration of the repeat loop in the algorithm; hence  $G_1$  and  $T_1$  are the input graph together with its forward branching, and  $G_k$  and  $T_k$  are the current graph and forward branching at the start of the *k* th iteration. Similarly let  $H_i$  be the graph *H* of step 2a of algorithm Outbridges constructed in the *i* th iteration of the repeat loop.

We first note that Observation 2 remains valid in each  $G_k$  when u is a vertex in a leaf chain of  $T_k$ . We state this in the following observation.

Observation 4: Let u be a vertex in a leaf chain l of forward branching T, where for convenience we assume that the vertices in the leaf chain are numbered from 1 to s, with 1 the root of the leaf chain and s the leaf of the leaf chain. Then reach(u) is a single interval of the form [u, u']. Further, a vertex  $v \neq u$  is in reach(u) if and only if there exists a sequence of back arcs  $b_i = (u_i, v_i), i = 1, \dots, k$  in L (where L is the subgraph of G induced by vertices in l) such that  $v_1 = u, u_k \geq v$ , and  $u_i \geq v_{i+1}, i = 1, \dots, k-1$ .

Lemma 7: For each  $k \ge 1$ , algorithm Outbridges correctly finds the out-bridges in the leaf chains of  $G_k$ .

**Proof:** By Observation 4, for a vertex u in a leaf chain l of  $T_k$ , reach(u) in  $G_k$  is the same as reach(u) in the subgraph of  $G_k$  induced by l. Hence the reach value of each vertex in the leaf chain is correctly computed in the Shrink computation of step 1b in algorithm Outbridges.

By Lemma 6, a vertex u in a leaf chain is active if and only if there is an arc e = (x, y) in  $G_k$ with x a non-descendant of u and with y in reach (u). Such an arc e is either a forward arc in the leaf chain or is an arc with x not in the leaf chain and y in the leaf chain. The former case is the same as that used in the Shrink algorithm. In the latter case, (x, y) will cause any vertex u in the leaf chain with y in reach (u) to be active. Hence for the purpose of the Shrink algorithm this is equivalent to having an arc from the root, t, of the leaf chain to y. Thus the computation in steps 1a and 1b of algorithm Outbridges correctly finds the outbridges in the leaf chains of  $G_k$ .[]

Lemma 8 Let e = (u, v) be an out-bridge in  $G_k, k > 1$ . Then e is an out-bridge in  $G_{k-1}$ .

*Proof:* First note that if e is an out-bridge in  $G_k$ , then e lies in  $T_k$ . Hence e lies in  $T_{k-1}$ , since every tree arc in  $T_k$  is present as a tree arc in  $T_{k-1}$ .

Suppose e is not an out-bridge in  $G_{k-1}$ . Hence v is an active vertex in  $G_{k-1}$ . Let p be an active path to v in  $G_{k-1}$ , and let p consist of an initial tree path p' to a vertex x that is a nondescendant of v, followed by a non-tree arc a=(x,y), where y is a descendant of v, followed by a final path p" from y to v using only arcs connecting vertices that are descendants of v. We now establish that there must be an active path to v in  $G_k$ , contradicting the assumption that e is an out-bridge of  $G_k$ , and thereby establishing the lemma.

If p contains no vertex in  $G_{k-1}-G_k$  then p is an active path to v in  $G_k$  as well. If p contains some vertices in  $G_{k-1}-G_k$  then consider the last vertex z on p such that z is in  $G_{k-1}-G_k$ .

Case 1: z is a non-descendant of v. Then z must be x and all vertices in p'' lie in  $G_k$ . Let t be the root of the leaf chain of  $G_{k-1}$  to which z belongs. Then by step 2d of algorithm Outbridges, z is collapsed into t and hence the path in  $G_k$  consisting of the tree path to t, followed by non-tree arc (t, y), followed by path p'' is an active path to v in  $G_k$ .

Case 2: z is a descendant of v. Let b = (z, a) be the outgoing arc from z in p, and let t' be the root of the leaf chain in  $G_{k-1}$  to which z belongs. Hence t' is a descendant of a and z is a proper descendant of t'. Let p''' be the portion of p'' from a to v. The path p''' is a path in  $G_k$  as well.

Case 2a: The vertex z is reachable from some non-descendant w of v in  $H_k$ . Then an arc (w, z) is introduced in step 2c of the algorithm. If w is in  $G_k$  then the path from r to w followed by arc (w,a) followed by path p''' is an active path to v in  $G_k$ . If w is in  $G_{k-1}-G_k$  then the analysis of Case 1 gives an active path to v in  $G_k$ .

Case 2b: The vertex z is not reachable from any non-descendant of v in  $H_k$ . Now consider p''. This is a path of the form  $\langle u_{1,1}, \cdots, u_{1,k_1}, v_{1,1}, \cdots, v_{1,l_1}, \cdots, u_{c,1}, \cdots, u_{c,k_e}, v_{c,1}, \cdots, v_{c,l_e} \rangle$ , where the  $u_{i,j}$  are in  $G_{k-1}-G_k$  and the  $v_{i,j}$  are in  $G_k$ , and if y is in  $G_k$  the initial sequence of  $u_{1,j}$ 's is empty. All of the  $u_{i,j}$  and  $v_{i,j}$  are descendants of v. Each  $v_{i,1}$  is reachable from  $v_{i-1,l_{i-1}}, i > 1$  in  $H_k$ . Hence by step 2c of algorithm Outbridges, there is an arc from  $v_{i-1,l_{i-1}}$  to  $v_{i,1}, i > 1$  in  $G_k$ . The vertex  $v_{1,1}$  has an incoming arc from x in  $G_k$ . The remaining arcs in p'' remain in  $G_k$ . Hence there is a path from x to v in  $G_k$  that contains only vertices that are descendants of v in  $G_k$ . Hence v is an active vertex in  $G_k$ .[] Lemma 9: Let e = (u, v) be a tree arc in  $G_k, k > 1$  that is not an out-bridge in  $G_k$ . Then e is not an out-bridge in  $G_{k-1}$ .

*Proof:* Since e is not an out-bridge in  $G_k$  there is an active path p to v in  $G_k$ . Consider any arc f = (x, y) in p that is not present in  $G_{k-1}$ . If f was introduced in step 2c of algorithm Outbridges then there is a path from x to y in  $G_{k-1}$  that avoids all tree arcs in  $G_k$  and hence arc e. If f was introduced in step 2d then there is a path from a descendant of x to y in  $G_{k-1}$  that avoids all tree arcs in  $G_k$ . Hence there is a path from x to y in  $G_{k-1}$  that avoids arc e. Hence from p we can obtain an active path p' to v in  $G_{k-1}$ . Thus e is not an out-bridge in  $G_{k-1}$ .[]

Lemma 10: Algorithm Outbridges correctly finds the out-bridges of G.

Proof: We show that at the start of each iteration of the repeat loop,

1) The out-bridges identified so far are exactly the out-bridges in the portion of the input graph G that has been collapsed by the algorithm.

2) An arc e in the current graph G is an out-bridge in this graph if and only if it is an out-bridge in the original input graph.

The proof is by induction on k, the number of iterations of the repeat loop.

Base: k=1. The claim is vacuously true since no out-bridges have been identified and the input graph is the same as the current graph.

Induction step: Assume that the two claims are true until the start of iteration k-1 and now consider the start of iteration k. Claim 1) follows by the induction hypothesis and Lemma 7. Claim 2) follows by the induction hypothesis and Lemmas 8 and 9.[]

Finally we note that algorithm Outbridges runs in  $O(\log^2 n)$  with  $O(n^3)$  processors on a CRCW PRAM. To see the processor and time bounds let us analyze the time complexity of each iteration of the repeat loop. By the previous analysis for the time complexity of algorithm Shrink, step 1 runs in  $O(\log n)$  time with  $O(n^2)$  processors on a CREW PRAM. Steps 2a, 2c, and 2d run in  $O(\log n)$  time with  $O(n^2)$  processors on a CREW PRAM. Step 2b runs in  $O(\log n)$  time with  $O(n^3)$  processors on a CREW PRAM. Step 2b runs in  $O(\log n)$  time with  $O(n^3)$  processors on a CREW PRAM. Step 2b runs in  $O(\log n)$  time with  $O(n^3)$  processors on a CREW PRAM. Step 2b runs in  $O(\log n)$  time with  $O(n^3)$  processors on a CREW PRAM, and is the most expensive step in the repeat loop. Since the repeat loop is executed  $O(\log n)$  times we obtain the stated time and processor bounds for algorithm Outbridges. On a CREW PRAM this algorithm runs in  $O(\log^3 n)$  time with M(n) processors.

Whether we use a CREW model or a CRCW model the time and processor bounds for finding a minimum weight branching using the algorithm in [Lo] dominate the time and processor bounds of algorithm Outbridges. Hence we can find redundant arcs within the time and processor bounds for minimum weight branchings, and thus the parallel transitive reduction algorithm runs in  $O(\log^3 n)$  parallel time with  $O(n^3)$  processors on a CRCW PRAM and in  $O(\log^4 n)$ 

parallel time with the same processor bound on a CREW PRAM.

# 4. Sequential Algorithms for Transitive Reduction

As in section 3.2, let r be a fixed root of a directed graph G = (V, E), where |V| = n and |E| = m. An algorithm for finding the in- and out-bridges is given in [Ta2]. This algorithm actually does more: It computes two forward branchings  $T_1$  and  $T_2$  having only the out-bridges in common, and two inverse branchings,  $T_3$  and  $T_4$ , having only the in-bridges in common. This algorithm can be implemented to run in linear time by using linear-time algorithms for computing nearest common ancestors [HT] and maintaining disjoint sets [GT].

Let R be the set of redundant arcs, I the set of in-bridges and O the set of out-bridges. Hence  $R = E - (I \cup O)$ . The following algorithm finds a transitive reduction of G.

1. Pick a root vertex r in G. Find a forward branching B and an inverse branching B' in G and replace G by  $B \cup B'$ .

#### 2. Repeat

a) Construct two forward branchings  $T_1$  and  $T_2$  having only the out-bridges in common; identify the set of out-bridges as O.

b) Construct two inverse branchings  $T_3$  and  $T_4$  having only the in-bridges in common; identify the set of in-bridges as I.

c) Form the set of redundant arcs R as  $R = E - (O \cup I)$ .

d) For i = 1, 2, 3, 4 form  $S_i = T_i \cap R$ .

e) Choose  $T_i$  and  $T_j$  such that  $1 \le i \le 2$ ,  $3 \le j \le 4$  and  $S_i \cup S_j$  has minimum cardinality among  $S_1 \cup S_3, S_2 \cup S_3, S_1 \cup S_4, S_2 \cup S_4$ .

f) Replace G by  $T_i \cup T_j$ .

until  $R = \phi$ .

The following claim establishes that the repeat loop is executed only  $O(\log n)$  times.

Lemma 11: In step 2e of the algorithm the chosen  $S_i$  and  $S_j$  satisfy  $|S_i \cup S_j| \le (3/4) \cdot |R|$ .

**Proof:** For i=1,2,3,4, let  $F_i$  be the set of those arcs in  $T_i$  that are not present in any other  $T_j$ , and let  $P_{1,3}=(S_1 \cap S_3) \cup F_1$ ,  $P_{2,4}=(S_2 \cap S_4) \cup F_2, P_{2,3}=(S_2 \cap S_3) \cup F_3$  and  $P_{1,4}=(S_1 \cap S_4) \cup F_4$ . Note that  $P_{1,3}, P_{2,4}, P_{2,3}$  and  $P_{1,4}$  are disjoint. Let one, say  $P_{1,3}$  be the one of maximum cardinality. Then we must have  $|P_{2,4}|+|P_{2,3}|+|P_{1,4}| \le (3/4) \cdot |R|$ . But  $S_2 \cup S_4 \subseteq P_{2,4} \cup P_{2,3} \cup P_{1,4}$ , which implies  $|S_2 \cup S_4| \le (3/4) \cdot |R|$ . Since we also have  $S_1 \cup S_3 \subseteq P_{1,3} \cup P_{1,4} \cup P_{2,3}$ ,  $S_1 \cup S_4 \subseteq P_{1,3} \cup P_{1,4} \cup P_{2,4}$  and  $S_2 \cup S_3 \subseteq P_{2,3} \cup P_{2,4} \cup P_{1,3}$ , we have  $|S_1 \cup S_3| \le (3/4) \cdot |R|$  if  $P_{2,4}$  is of maximum cardinality,  $|S_1 \cup S_4| \le (3/4) \cdot |R|$  if  $P_{2,3}$  is of maximum cardinality and  $|S_2 \cup S_3| \le (3/4) \cdot |R|$  if  $P_{1,4}$  is of maximum cardinality. Hence the chosen  $S_i$  and  $S_j$  in step 2e of the algorithm satisfy  $|S_i \cup S_j| \le (3/4) \cdot |R|$ .[]

Step 1 of the algorithm takes O(n+m) time and renders G sparse  $(O(n) \operatorname{arcs})$ . As mentioned above, steps 2a and 2b can be implemented to run in O(n) time using the algorithm in [Ta2], in conjunction with the algorithms in [HT] and [GT]. Each of steps 2c through f takes O(n) time. Hence each execution of the repeat loop takes linear time. Since by Lemma 11 the repeat loop is executed  $O(\log n)$  times, the entire transitive reduction algorithm runs in  $O(m+n\log n)$  time.

The algorithm of section 2 can also be implemented to run in  $O(m+n\log n)$  time. This is because the minimum-weight branching algorithm of Edmonds [Ed2] can be implemented to run in linear time for 0-1 edge weights by using the algorithm in [GGST], with the heaps replaced by two buckets. As before, the redundant arcs can be found in linear time and hence each execution of the repeat loop takes linear time, leading to an  $O(m+n\log n)$  time sequential algorithm for transitive reduction.

We have obtained sequential and parallel algorithms with similar complexities for analogous problems on undirected graphs, i.e., for finding a minimal bridge-connected spanning subgraph and a minimal biconnected spanning subgraph in an undirected graph, if such subgraphs exist. These results will appear in a companion paper.

We conclude by noting that it is conceivable that one (or both) of our sequential algorithms runs in linear time, since it is possible that the repeat loop needs to be executed only a constant number of times. We leave this question for further investigation. For the same reason it is possible that our parallel algorithms run faster than the stated time bounds by an  $O(\log n)$  factor.

### References

- [ABI] Alon, N., Babai, L. and Itai, A., "A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem", J. of Algorithms, 7, pp. 567-583, 1986.
- [AGU] Aho, A.V., Garey, M.R. and Ullman, J.D., "The Transitive Reduction of a Directed Graph", SIAM J. Comput., 1, pp. 131-137, 1972.
- [CW] Coppersmith, D. and Winograd, S., "Matrix multiplication via arithmetic progressions," Proc. 19th Ann. ACM Symp. on Theory of Computing, pp. 1-6, 1987.

- [Ed] Edmonds, J., "Edge-Disjoint Branchings", in *Combinatorial Algorithms*, Algorithmic Press, pp. 91-96, 1973.
- [Ed2] Edmonds, J., "Optimum branchings," J. of Res. of the Nat. Bureau of Standards, 71B, 1967, pp. 233-240.
- [ET] Eswaran, K.P. and Tarjan, R.E., "Augmentation Problems", SIAM J. Comput., 5, pp. 653-665, 1976.
- [GGST] Gabow, H.N., Galil, Z., Spencer T. and Tarjan, R.E., "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs," *Combinatorica*, 6, pp. 106-122, 1986.
- [GJ] Garey, M.R. and Johnson, D.S., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Fransisco, CA, 1979.
- [GT] Gabow, H.N. and Tarjan, R.E., "A linear-time algorithm for a special case of disjoint set union," Journal of Comput. Sys. Sci., 30, pp. 209-221, 1985.
- [GS] Goldberg, M. and Spencer, T., "A New Parallel Algorithm for the Maximal Independent Set Problem", Proc. 28th IEEE Symp. on Foundations of Computer Science, pp. 161-165, 1987.
- [HT] Harel, D. and Tarjan, R.E., "Fast algorithms for finding nearest common ancestors," SIAM J. Comput., 13, pp. 338-355, 1984.
- [KR] Karp, R.M. and Ramachandran, V. "Parallel algorithms for shared memory machines," *Handbook of Theoretical Computer Science*, J. van Leeuwen, ed., North Holland, to appear; also Report No. UCB/CSD 88/408, Computer Science Div., Univ. of California, Berkeley, CA, 1988.
- [KUW] Karp, R.M., Upfal, E. and Wigderson, A., "Are Search and Decision Problems Computationally Equivalent?", Proc. 17th ACM Symp. on Theory of Computing, pp. 464-475, 1985.
- [KW] Karp, R.M. and Wigderson, A., "A Fast Parallel Algorithm for the Maximal Independent Set Problem", JACM, pp. 762-773, 1985.
- [Lo] Lovasz, L. "Computing Ears and Branchings in Parallel", Proc. 26th IEEE Symp. on Foundations of Computer Science, pp. 464-467, 1985.
- [Lu] Luby, M. "A Simple Parallel Algorithm for the Maximal Independent Set Problem", Proc. 17th ACM Symp. on Theory of Computing, pp. 1-10, 1985.
- [MR] Miller, G.L. and Reif, J.H., "Parallel tree contraction and its applications," *Proc. 26th Ann.* Symp. on Foundations of Comp. Sci., pp. 478-489, 1985.

- [Ra] Ramachandran, V. "Fast parallel algorithms for reducible flow graphs," Concurrent Computations: Algorithms, Architecture and Technology, S.K.Tewksbury, B.W.Dickinson and S.C.Schwartz, ed., Plenum Press, New York, NY, 1988, in press.
- [So] Soroker, D., "Fast Parallel Strong Orientation of Mixed Graphs and Related Augmentation Problems", J. of Algorithms, 9, pp. 205-223, 1988.
- [Ta] Tarjan, R.E., "Finding Optimum Branchings", Networks, 7, pp. 25-35, 1977.
- [Ta2] Tarjan, R.E., "Edge-disjoint spanning trees and depth-first search," Acta Informatica, 7, pp. 171-185, 1976.