International Journal of Innovative
Computing, Information and Control
Volume **5**, Number **3**, March **2009**

ICIC International ©2009 ISSN 1349-4198
pp. 581–588

PARALLEL ALGORITHM FOR FINDING MINIMUM EDGES REQUIRED TO MAKE A DAG STRONGLY CONNECTED

Tsuyoshi Itokawa[1], Akio Tada[2] and Masahiro Migita[3]

[1]Computer Science and Electrical Engineering
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami Kumamoto, Japan
itokawa@cs.kumamoto-u.ac.jp

[2]Department of Computer System Technology
Faculty of Computer and Information Sciences
Sojo University
4-22-1 Ikeda Kumamoto, Japan
tada@cis.sojo-u.ac.jp

[3]Center for Multimedia and Information Technologies
Kumamoto University
2-39-1 Kurokami Kumamoto, Japan
migita@cc.kumamoto-u.ac.jp

Received February 2008; revised June 2008

Abstract. *Making a graph strongly connected is one of the most fundamental problems of graph theory. The parallel algorithm currently known* [1] *solves this problem in* $O(\log n)$ *time using* $O(n^3)$ *processors in a CRCW PRAM model. In this paper, we present a parallel algorithm to determine minimum edges required to make a disconnected directed acyclic graph (DAG) strongly connected in* $O(\log(m+n))$ *time using* $O(m+n)$ *processors in a CREW PRAM model. This algorithm is an efficient parallel algorithm because the number of processors varies according to the density of the given graph.*
Keywords: Parallel algorithm, Directed acyclic graph, Strongly connected

1. **Introduction.** A directed graph $G = (V, E)$ is strongly connected if for every ordered pair of vertices u and v in V there is a path from u to v. Determining minimum edges required to make a disconnected directed graph strongly connected is one of the fundamental problems of graph theory.

In line with increases in the size of problems being solved using computers and great advances in integrated circuit technology, parallel computation has become increasingly realistic. To avoid discussions based on a specific architecture, various parallel computation models have been proposed so far. A parallel random access machine (PRAM) is a parallel computation model consisting of a number of processors connected to a global shared memory. In regard to simultaneous access of a specific memory location by some processors, there are some different types in PRAM model. In the most powerful model called CRCW PRAM, both concurrent reading and concurrent writing are permitted. On the other hand, in the CREW PRAM model, concurrent reading is permitted, but writing is performed exclusively. In general, it is said that emulating an algorithm on a CRCW PRAM model using a CREW PRAM model requires $O(\log N)$ times computational power, where N is the size of the problem.

The currently known parallel algorithm in a disconnected directed graph with n vertices and m edges takes $O(\log n)$ time using $O(n^3)$ processors in a CRCW PRAM model.