An Efficient CGM-Based Parallel Algorithm Solving the Matrix Chain Ordering Problem

Jean Frédéric Myoupo, UFR Sciences, University of Picardie Jules Verne, Amiens, France
Vianney Kengne Tchendji, Department of Computer Science, University of Yaounde 1, Yaounde, Cameroon

ABSTRACT
This study focuses on the parallel resolution of the matrix chain ordering problem and the optimal convex polygon triangulation problem on the Coarse grain multicomputer model (CGM for short). There has been intensive work on the parallelization of these dynamic programming problems in PRAM, including the use of systolic arrays, but a BSP/CGM solution is necessary for ease of implementation and portability. Our CGM algorithm is based on Yao’s sequential solution running in \(O(n^2)\) time and \(O(n^2)\) space. This CGM algorithm uses \(p\) processors, each with \(O(n/p)\) local memory. It requires at most \(O(S/p \times n^2)\) running time with \(S\) communication rounds and with \(S/p < 1\). Our algorithm performs better than the algorithm proposed in 2012 by Dilson and Marco when \(S\) is less than \(n/p\). We offer several ways of partitioning the problem to solve and study the impact of each partitioning algorithm performance. A CGM solution exists based on Yao’s algorithm, but the subdivision of tasks is defined according to the BSP cost model. In this paper, we propose a solution based only on the CGM model specifications. Note that \(S\) is the number of super-steps of the CGM algorithm.

Keywords: Bulk Synchronous Parallel, Coarse Grain Multicomputer, Dynamic Programming, Parallel Processing, PRAM

1. INTRODUCTION
Dynamic Programming (DP) is a paradigm used to solve optimization problems that is applied to a large number of areas including optimal control, industrial engineering, economics and artificial intelligence (Dehne, Ferreira, Caceres, Song, & Roncato, 2002; Gupta & Tang, 1995). Many practical problems involving a sequence of interrelated decisions can be efficiently solved by DP. The essence of many DP algorithms lies in computing solutions of the smallest sub-problems and sorting the results for later use in computing larger sub-problems. Thus the solution to the original problem is constructed in a bottom-up fashion, from the smallest sub-problems to the largest.

ADP formulation of a problem is expressed as a recursive functional equation whose left-hand side is an expression involving the

DOI: 10.4018/ijghpc.2014040105
maximization (or minimization) of some cost 
functions (Equation (1)). Guo and Benjamin 
(1985) have developed a classification of DP 
schemes according to the form of the functional 
equations and the nature of the recursion. A DP 
formulation is monadic if the inherent cost func-
tion involves only one recursive term, otherwise 
it is polyadic. It is serial if the sub-problems 
can be grouped in levels and the solution to 
any sub-problem in a certain level can be found 
using sub-problems that belong only to the 
levels immediately preceding, otherwise it is 
non-serial. We are interested in a CGM (coarse 
grain multicomputer)-based parallel solution 
for a typical polyadic non-serial dynamic 
programming problem, such as the optimal 
string parenthesizing (OSP) problem, the op-
timal binary search tree (OBST) problem, the 
optimal convex polygon triangulation (OCPT) 
problem and all problems that can be modeled 
by a recurrence equation similar to (1).

2. RELATED WORK

The classical sequential algorithm, or Godbole’s 
algorithm (Godbole, 1973), for these problems 
is based on a dynamic programming technique. 
It requires \( O(n^3) \) calculation operations and 
\( O(n^2) \) memory space. By using the monotonic-
ity property of OBST, Knuth (1973) derived an 
\( O(n^2) \) algorithm in the same space. Yao (1982) 
obtained the same result with the help of the 
quadrange inequalities. Eppstein, Galil and 
Giancarlo (1988) developed an \( O(n \log n) \) 
algorithm using the restrictive assumption of 
convexity. Whereas the parallelization of the 
classical version has been extensively studied 
by the community of parallel processing re-
searchers for the different parallel computing 
models (Bradford, 1994; Fotso, Kengne, & 
Myoupo, 2010; Guibas, Kung, & Thompson, 
1979; Gupta & Tang, 1995; Karypis & Kumar, 
1993; Kengne & Myoupo, 2012; Rytter, 1988), 
few works have been produced on the paral-
lelization of the Knuth approach (Kechid & 
Myoupo, 2008a) or the Yao approach (Kechid & 
Myoupo, 2008b).

In this study, we parallelize the Yao algo-


Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Related Content

A Mobile Cloud Computing System for Mathematical Computation
[www.irma-international.org/article/a-mobile-cloud-computing-system-for-mathematical-computation/141299/](www.irma-international.org/article/a-mobile-cloud-computing-system-for-mathematical-computation/141299/)

Implementing Geospatial Web Services for Cloud Computing
[www.irma-international.org/chapter/implementing-geospatial-web-services-cloud/64506/](www.irma-international.org/chapter/implementing-geospatial-web-services-cloud/64506/)

Methodology for Information Management and Data Assessment in Cloud Environments

Experimental Error Measurement in Monte Carlo Simulation
[www.irma-international.org/chapter/experimental-error-measurement-monte-carlo/38259/](www.irma-international.org/chapter/experimental-error-measurement-monte-carlo/38259/)

A Performance Study of Secure Data Mining on the Cell Processor
[www.irma-international.org/article/performance-study-secure-data-mining/3964/](www.irma-international.org/article/performance-study-secure-data-mining/3964/)