ELSEVIER

# Parallel dynamic programming and automata theory

D.G. Morales, F. Almeida\*, C. Rodríguez, J.L. Roda, I. Coloma,
A. Delgado

*Dpto. Estadística, Investigación Operativa y Computación, Universidad de La Laguna, 38071 Tenerife, Spain*

**Abstract**

Following Karp's discrete Dynamic Programming (DP) approach, this work extends the sequential model for monadic DP to the parallel case. We propose general parallel DP algorithms for pipeline and ring networks. The study of the optimality of these algorithms leads us to the introduction of new classes of multistage automata. However, the important class of Polyadic Dynamic Problems is out of the scope of this theory. Based on the concept of tree automaton, a new theory covering both Sequential and Parallel Dynamic Programming Polyadic Programs is presented. Monadic Programs appear as a particular case of this new theory. A large number of experiments have proved that the schemes proposed in this work lead to efficient implementations. © 2000 Published by Elsevier Science B.V. All rights reserved.

*Keywords:* Automata theory; Dynamic programming; Pipeline parallelism; Tree automata

## 1. Introduction

Dynamic Programming is an important problem-solving technique that has been widely used in various fields such as operations research [18,31], compute science [1,17], control theory [17], control theory [15,28,30], and biology [13]. Li and Wah [24,36] classify Dynamic Programming (DP) programs into monadic and polyadic according to the form of the functional equation. When the functional equation contains a single recursive term it yields to a monadic DP formulation. DP formu-

---

\* Corresponding author.
  *E-mail address:* falmeida@ull.es (F. Almeida).

lations whose cost function contains multiple recursive terms are called polyadic formulations. Sequential computation for DP has been extensively studied. A theory for General Discrete Sequential Monadic Dynamic programs viewed as multistage finite automaton with a certain cost structure superimposed was presented in [18,21]. Unfortunately, the important class of Polyadic Dynamic Problems was out of the scope of this theory.

Many authors describe parallel algorithms for specific DP problems. Parallel DP algorithms for 0/1, two-dimensional and integer knapsack problems have been proposed for different architectures in [2,4,6,8,20–23,27,32,35]. Specific recurrence relations such as shortest paths, optimal binary search tree, optimal chained matrix multiplication parathesizations, etc. have been approached for systolic arrays [25,27,29], general shared-memory multiprocessors [12] and for PRAM models [14,20]. However, few efforts have been made for general parallel DP formal model. The common practice is to introduce the parallelization of the functional equation in an intuitive and non-formal way. For example, in the previously quoted work [24,36] Li and Wah suggest parallelizations of certain classes of DP formulations using parallel matrix product algorithms. It is a well-known fact that the availability of a formal model for a given paradigm constitutes the core for the development of automatic tools and skeletons for the parallelization of such paradigm. In this paper, we develop formal models that allow to broach DP problems in a methodical fashion. The proposed models may constitute the kernel of high level tools to parallelize the technique.

Following Karp's [21] discrete DP approach, this work extends the sequential model for monadic DP to the parallel case. We propose general parallel Dynamic Programming algorithms for pipeline and ring networks. The study of the optimality of these algorithms leads us to the introduction of new classes of multistage automata. A problem corresponding to these type of automata can be easily parallelized. So the form of the automata is a guideline for the programmers, who should try to reformulate their problems in such a way that they corresponds to one of these automata. Based in the concept of tree automata [10,11,33,34], a new theory covering both Sequential and Parallel DP Polyadic Programs is presented. Monadic Programs appear as a particular case of this new theory.

Four representative platforms of the current state of parallel computing hardware were used for the experiences. The IBM SP2 is a distributed memory computer with a omega-like multistage interconnection network. The CRAY T3E is a three-dimensional torus with a shared address space but no shared memory. The Silicon Origin 2000 (O2000) computer has a shared distributed memory and a hypercube interconnection network. Completing the group, the Digital Alpha Server 8400 (DEC 8400) is a single bus shared memory computer. The native versions of MPI were used for all the implementations.

In Sections 2 and 3 we present the extension to the parallel case of the theory for the monadic case, we deduce the algorithms and study the conditions for their optimality. Following an analogous scheme, Sections 4 and 5 are devoted to the new theory for the polyadic case. Computational results for both examples of Monadic and Polyadic Problems (the Single Resource Allocation Problem the Matrix