An Adaptive and Cost-Optimal Parallel Algorithm for Minimum Spanning Trees*

S. G. Akl, Kingston, Canada

Received January 21, 1985

Abstract — Zusammenfassung

An Adaptive and Cost-Optimal Parallel Algorithm for Minimum Spanning Trees. A parallel algorithm is described for computing the minimum spanning tree of an undirected, connected and weighted graph with n vertices. We assume a shared-memory single-instruction-stream, multiple-data-stream model of computation which does not allow read or write conflicts. The algorithm is adaptive in the sense that it uses \( n^{1-e} \) processors and runs in \( O(n^{1+e}) \) time where \( e \) lies between 0 and 1 and depends on the number of available processors. In view of the obvious \( \Omega(n^2) \) lower bound on the number of operations required to compute a minimum spanning tree, the algorithm is also cost-optimal.

CR Categories and Subject Descriptors: B.3.2 [Memory Structures]: Design Styles — shared memory; C.1.2 [Computer System Organization]: Multiple Data Stream Architectures (Multi-processors) — single-instruction-stream, multiple-data-stream processors (SIMD); F.1.2 [Computation by Abstract Devices]: Modes of Computation — parallelism; F.2.2, F.2.3 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems — computations on discrete structures; Tradeoffs among Complexity Measures; G.2.2 [Discrete Mathematics]: Graph Theory — graph algorithms, trees.

General term: Algorithms.

Key words: Adaptive algorithm, cost-optimal algorithm, minimum spanning tree, parallel algorithm.

Ein adaptiver und kostenoptimaler Parallel-Algorithmus für Minimalgerüste. Wir beschreiben einen Parallel-Algorithmus, der ein Minimalgerüst für einen ungerichteten, zusammenhängenden bewerteten Graphen mit \( n \)-Knoten ermittelt. Das zugrunde liegende Rechnermodell ist ein Mehrprozessorsystem mit gemeinsamem Speicher, das vielfachen Datentransfer erlaubt und von einer einzigen Zentraleinheit kontrolliert wird. Les- oder Schreibkonflikte des Systems sollen ausgeschlossen sein. — Der Algorithmus paßt sich an die Zahl der verfügbaren Prozessoren an. Mit \( n^{1-e} \)-Prozessoren läuft er in der Zeit \( O(n^{1+e}) \), wobei \( e \) zwischen null und eins liegt. Der Algorithmus ist kostenoptimal, da die Berechnung des Minimalgerüsts \( \Omega(n^2) \) Operationen benötigt.

1. Introduction

An undirected and connected graph \( G \) with a sequence \( V = \{v_1, v_2, \ldots, v_n\} \) of \( n \) vertices and a sequence \( E \) of \( m \) edges is given together with a function length (or weight) which associates an arbitrary real number to each edge \( (v_i, v_j) \) representing

* This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant NSERC – A 3336 (Technical Report No. 85-164).
its length (or weight). A minimum spanning tree (MST) of $G$ is a tree connecting all the vertices of $V$ with edges of $E$ such that the sum of its edge lengths is a minimum [1]. When all the edges in $E$ have distinct lengths, the MST is unique. Since the MST consists of $n-1$ edges chosen among potentially $n(n-1)/2$ candidates, a lower bound on the number of operations to compute it is $\Omega(n^2)$.

The problem of devising algorithms for computing the MST on various parallel models of computation has received considerable attention over the past twelve years as testified by the list of references [2–26]. As usual with parallel algorithms, existing MST algorithms vary with respect to the parallel model of computation on which they are designed to run and, consequently, differ in their performances. An overview and comparison of several parallel MST algorithms is provided in [26].

This paper describes a parallel MST algorithm for the shared-memory single-instruction-stream, multiple-data-stream (SM-SIMD) model of computation. The algorithm uses $n^{1-\epsilon}$ processors and runs in $O(n^{1+\epsilon})$ time where $\epsilon$ lies between 0 and 1 and depends on the number of available processors. The algorithm is therefore adaptive. It is also cost-optimal in the sense that the product of its running time $t(n)$ and the number of processors it uses $p(n)$ is $O(n^2)$ and hence matches the lower bound stated above. This is the first adaptive and cost-optimal parallel MST algorithm in which no restrictions are placed on the length function and the number of processors is allowed to vary between 1 and $n$.

It should be noted that another parallel MST algorithm exists which is also adaptive and cost-optimal [6]. However, that algorithm differs from the algorithm in this paper in two respects:

1. the algorithm in [6] deals with the special case in which the vertices of $G$ are points in Euclidean $k$-space and the length of an edge is the distance in space between its endpoints; and
2. the cost-optimality of the algorithm in [6] depends on the number of processors lying within a given range of values.

The remainder of this paper is organized as follows. We start by defining our model of parallel computation in section 2. Sections 3 and 4 are devoted to a formal description of the algorithm and its analysis, respectively. Some concluding thoughts are offered in section 5.

2. Computational Model

As mentioned above, our model of a parallel computer is the SM-SIMD machine [27, 28]. In this model, $N$ processors numbered 1 to $N$ share a common memory and operate under the control of a single instruction stream issued by a central control unit. In addition to the shared memory, each processor possesses a local memory in which programs and data are stored. The processors operate synchronously: during a given time unit a selected number of processors are active and execute the same instruction each on a different data set; the remaining processors are inactive. When two processors wish to communicate they do so through the shared memory: one

---

1 In the remainder of this paper we assume that $n^{1-\epsilon}$ and $n^{1+\epsilon}$ stand for $\lceil n^{1-\epsilon} \rceil$ and $\lfloor n^{1+\epsilon} \rfloor$, respectively.