Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 119 (2017) 97–102

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Young Scientists Conference in HPC and Simulation, YSC 2017,
1-3 November 2017, Kotka, Finland

Parallelizing an exact algorithm for the traveling salesman problem

Victor Vitalyevich Burkhovetskiy[a,*], Boris Yakovlevich Steinberg[a]

[a] *Southern Federal University, Institute of Mathematics, Mechanics, and Computer Science, 8a Milchakova st., Rostov-on-Don 344090, Russia*

Abstract

We describe a project of an exact parallel algorithm for traveling salesman problems (TSPs) of large sizes. It is based on the algorithm developed by E. Balas and N. Christofides almost 40 years ago. Balas' and Christofides' algorithm uses branch-and-bound, Hungarian algorithm for assignment problem, and several heuristics to effectively eliminate unnecessary branches of the solution tree. We propose several modifications (including parallelization) of the simplified version of their algorithm which should improve its performance. The modified algorithm was implemented in C++ with OpenMP and tested on graphs with around 1000 nodes. Computational results indicate a possibility of getting exact solutions of TSPs with 10000 nodes in minutes. New heuristic algorithms could be derived from the exact algorithm to solve even bigger problems. Such algorithms could be applied to genome assembly.

Keywords: parallel computing, tree traversal, branch-and-bound, traveling salesman problem, assignment problem

1. Introduction

This article introduces an exact parallel algorithm for the traveling salesman problem (TSP) and a project of improving it. The algorithm is based on core ideas of Balas' and Christofides' algorithm [7] but doesn't use some of the heuristic procedures described in [7]. We hope, that once our project is complete, we will be able to apply the algorithm to the genome assembly problem, which is reducible to the TSP in polynomial time [8].

The traveling salesman problem is a problem of finding a minimum Hamiltonian cycle on a complete directed graph with non-negative edge costs. A Hamiltonian cycle is a graph cycle that visits each node exactly once. Many works, such as [1], [6], [11], [13], [14], are dedicated to solving the TSP. There is also a library of sample data for the TSP from various sources and of various types [16]. The TSP is NP-hard, which means that, at the moment, there is no exact polynomial-time algorithm for it. However, our computational results show that large-scale NP-hard problems can be solved with exact exponential algorithms in an acceptable amount of time.

[*] Corresponding author.
E-mail address: buvictor95@inbox.ru

10.1016/j.procs.2017.11.165

Balas' and Christofides' algorithm [7] could be considered a breakthrough among exact methods for the TSP. It uses the Hungarian algorithm [5] to solve the assignment problem associated with the TSP. Similar approach was adopted in [10], [11], where the associated assignment problem is solved heuristically. In 1979, Balas' and Christofides' program could solve TSPs whose graphs had 325 nodes with uniform random integer edge costs in the range from 1 to 1000 in less than 2 minutes, which was quite an impressive result at the time. Afterwards, more robust exact sequential algorithms [17] were created. They were able to solve TSPs with 1000 nodes and uniform random integer edge costs from 0 to 1000 in seconds.

Articles [9], [15] describe a dynamic structure for parallel branch-and-bound tree traversal on a high performance cluster with distributed memory. Descendants of each tree node are placed into the task pool and distributed among cluster nodes. Paper [12] deals with heuristic and incomplete traversal of trees, which could also be parallelized.

Analysis of a much simpler TSP algorithm [3] has inspired us to develop a library for parallel tree traversal on a single computer with shared memory. It automatically creates tasks out of branch-and-bound tree nodes and distributes them among available threads. The tasks are executed in increasing order of lower bounds (as described in section 3).

Section 2 discusses how much of an increase in performance can be expected, once a branch-and-bound algorithm is parallelized, section 3 describes the modified algorithm, section 4 compares the performance of sequential and parallel algorithm, and section 5 shows our plans of improving the algorithm.

2. Limitations of Parallel Branch-and-Bound Algorithms

Computational results of [4] show, that it's nearly impossible to make exponential branch-and-bound algorithms work as many times faster as there are cores on the processor (i.e. to achieve the perfect speedup), because during a sequential tree traversal the current best value of the cost function can (and probably will) improve several times, which helps to cut unnecessary branches closer to the root. Threads of a parallel program could explore the unneeded branches further than one thread of a sequential program would, because the current best value could not have been improved enough to eliminate them. Same conclusion can also be drawn for clusters [9], [15].

Heuristic algorithms could be used to find a good cost function value to use it as the initial best value for the exact algorithm [9], [15]. Such approach increases the effectiveness of parallelization.

3. Simplified and Modified Balas' and Christofides' Algorithm

Balas' and Christofides' algorithm [7] is an exact branch-and-bound algorithm for solving the TSP. The algorithm uses two lemmas:

Lemma 1. *If we subtract an arbitrary constant C_0 from all elements of a given row or column of the weighted adjacency matrix of a graph so that the elements remain non-negative, then the cost of minimum Hamiltonian cycle will decrease by C_0, but the cycle itself (i.e. the order of nodes in it) won't change [1], [7].*

Lemma 2. *If we interpret the weighted adjacency matrix of a complete directed graph with n nodes as an adjacency matrix of a bipartite graph with $2n$ nodes and solve the associated assignment problem (AP), then the solution forms either a Hamiltonian cycle, or a union of disjoint simple cycles on the former graph, and said union covers all nodes of the graph. Sum of all edge costs of the solution is less than or equal to the cost of the minimum Hamiltonian cycle [7].*

We used the following simplified (several optimizations are excluded) and modified (see below) version of their algorithm:

1. Input: a complete directed graph with n vertices and non-negative edge costs represented as a weighted adjacency matrix (the implementation uses a weighted adjacency list, but, for simplicity, we will use an adjacency matrix in our description of the algorithm). The initial best value is set to the cost of any Hamiltonian cycle (e.g. $1 \to 2 \to \ldots \to n \to 1$).
2. The matrix is then interpreted as an adjacency matrix of a bipartite graph with $2n$ nodes, and associated AP is solved with the Hungarian algorithm. Dual variables u_i, $i = 1, \ldots, n$, and v_j, $j = 1, \ldots, n$, calculated by the