Engineering Applications of Artificial Intelligence 39 (2015) 267–278

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

On the performance of parallel hybrid algorithms for the solution of the quadratic assignment problem

Umut Tosun *

Department of Computer Engineering, Baskent University, Baglica Kampusu, Eskisehir Yolu 20. km, 06530 Ankara, Turkey

ARTICLE INFO

Article history:
Received 26 March 2014
Received in revised form
20 December 2014
Accepted 22 December 2014
Available online 20 January 2015

Keywords:
Quadratic assignment problem
Parallel hybrid algorithm
Tabu search
Genetic algorithms
Ant colony algorithm

ABSTRACT

The quadratic assignment problem (QAP) is a combinatorial optimization problem, which is computationally demanding, and considered to be NP-hard. Therefore, the problem cannot be solved in polynomial time. The known sequential algorithms can solve small problem instances within long computational times; moreover, parallelization may provide only a linear speed-up. Near-optimal solutions can be obtained in feasible times using heuristics like genetic algorithms and tabu search. The QAP algorithms can be modified to solve various problems like the travelling salesman problem, the data allocation problem, and the file allocation problem. In this paper, a parallel hybrid algorithm (PHA) with three stages was proposed. In the first stage, a genetic algorithm was used to obtain a high quality seed. Later, a diversification phase was run on the initial seed. Finally, a robust tabu search was run on the intermediate solution to find a near-optimal result. Parallel computing was used to increase the seed quality, and a considerable speed-up was obtained in the diversification phase of the tabu search. The QAPLIB benchmark instances were used to conduct the experiments. The PHA is quite competitive with respect to the best-performing algorithms in the literature in terms of solution quality and execution time. It achieves results on average within 0.05% of the best solutions given in the QAPLIB. The PHA was able to solve even the largest problem instance size of 256 within 11 h, and with a higher accuracy than the best-known solutions. It was also observed that the solution quality improved considerably especially for larger instances, when the degree of parallelism increased.

1. Introduction

A fundamental class of optimization problems involves assigning assets to tasks to minimize a desired cost function. These problems are categorized as Assignment Problems (Pentico, 2007). Variations of these problems have been studied over the years with a wide range of applications in the domains of telecommunications, transportation systems and signal processing (Burkard and Cela, 1999). The classical approach to the Quadratic Assignment Problem (QAP) was first introduced by Koopmans and Beckmann (1955) as a mathematical model for the location of indivisible economic activities. Since then it has been one of the most interesting challenges for scientists having been used for modelling a great variety of problems. Typewriter keyboard design, backboard wiring (Steinberg, 1961), layout design (Rossin et al., 1999), turbine balancing (Pfister, 1998), scheduling (Lim et al., 2000), and data allocation (Adl and Rankoohi, 2009) are some of the problems that have been successfully modeled as QAP. The service allocation problem with the purpose of minimizing the container re-handling operations at a shipyard (Cordeau et al., 2007), detection of global optimality conditions for quadratic programming (Wu et al., 2012), travelling salesman, bin-packing, maximum clique, linear ordering, and the graph-partitioning problem are among the interesting applications of the QAP.

In its simplest form, the QAP is the problem of assigning n facilities to n locations with cost proportional to the amount of material flow between the facilities multiplied by the distances between locations, the initial costs for placing the facilities at their respective locations. The objective is to find an allocation such that the total cost of allocating and operating all facilities is minimized. The QAP can be formally modeled by using three $n \times n$ matrices, A, B, and C:

$$A = (a_{ik}) \tag{1}$$

where a_{ik} is the flow amount from facility i to facility k.

$$B = (b_{jl}) \tag{2}$$

where b_{jl} is the distance from location j to location l.

$$C = (c_{ij}) \tag{3}$$

where c_{ij} is the cost of placing facility i at location j.

* Tel.: +90 312 2462099; fax: +90 312 2466660.
E-mail address: utosun@baskent.edu.tr

http://dx.doi.org/10.1016/j.engappai.2014.12.011
0952-1976/

The Koopmans–Beckmann form of QAP can be written as

$$\min_{\phi \epsilon S_n}\left(\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ik}b_{\phi(i)\phi(k)} + \sum_{i=1}^{n} c_{i\phi(i)}\right) \quad (4)$$

where S_n is the set of all permutations of integers 1,2,…,n. Each individual product $a_{ik}b_{\phi(i)\phi(k)}$ is the transportation cost caused by assigning facility i to location $\phi(i)$ and facility k to location $\phi(k)$. Each term $c_{i\phi(i)} + \sum_{k=1}^{n} a_{ik}b_{\phi(i)\phi(k)}$ is the total cost given, for facility i, the cost for installing it at location $\phi(i)$, plus the transportation costs to all other facilities k, installed at locations $\phi(1), \phi(2)\ldots\phi(n)$. An instance of the QAP with input matrices A, B, and C is denoted by QAP (A, B, C). If there is no C term, it can be written as *QAP* (A, B).

The applicability of the QAP to the solution of many different problems has made it the subject of extensive research area for exhaustive and metaheuristic strategies. Small size QAP instances are appropriate for exact solutions but the larger instances cannot be solved in reasonable times due to the computational limits. Therefore, metaheuristic approaches have gained a reputation for their ability to produce high-quality solutions within the computational limitations. Simulated Annealing (Wilhelm, 1987; Connolly, 1990), Neural Networks (Calzon-Bousono, 1995), Genetic Algorithms (GAs) (Cohoon and Paris, 1987; Tate and Smith, 1995), GRASP (Li et al., 1994), Tabu Search (TS) (Battiti and Tecchiolli, 1994), and Ant Colony Optimization (ACO) (Gambardella et al., 1999) are some of the well-known metaheuristics that have been successfully applied to the QAP.

In this paper, parallel hybrid algorithm (PHA) was proposed for the solution of the QAP. It combines the meta-heuristic of parallel algorithms (PAs), with the local search technique of TS. The QAPLIB (Burkard et al., 1991) benchmark instances were used during the tests, and extensive experiments were performed on these instances to report the results. PHA can produce high-quality solutions in shorter times than most of the best algorithms in the literature. The remainder of the paper is organized as follows: In Section 2, an overview of the exact solution methods was presented including tabu search and multi-start methods, sequential meta-heuristics, and parallel meta-heuristics. Most are hybrid approaches in which, usually, a local search technique like TS is used. Taillard's (1991) Robust TS is a very effective way of local searching, and its computational time is so short that this property makes it a very attractive tool for the researchers using the QAP. In Section 3, island parallel genetic algorithms (IPGAs) were explained. IPGAs are used to improve the solution quality of the GAs. The population is separated into nodes as islands, and a near-optimal solution is generated after the execution of the algorithm. A population exchange within 10% during the initial execution of the IPGA may increase the variation of the population resulting in a better solution. In Section 4, the proposed algorithm PHA was described. There are three stages of the PHA. The first stage benefits from the IPGAs to create a high quality seed. The second stage is a diversification phase using the high quality seed. Finally, the third stage runs a robust tabu search to find a near-optimal solution. Section 5 provides the comparison of PHA with other algorithms in the literature. Experiment results were presented for various instances of the QAPLIB. The results were compared with the best-performing algorithms in the literature. The experiments were run with 10 nodes to fairly compare the results with cooperative parallel tabu search (CPTS) algorithm. However, extensive scaling tests were also provided to understand how the degree of parallelization affects the solution quality and the execution time. The experimental results of the sequential algorithms were given as well as the parallel algorithms to better understand the impact of the parallelization on the performance. Finally, Section 6 provides some concluding remarks regarding this research.

2. Solution methods for the QAP

The QAP is a NP-hard problem, so that no polynomial time algorithm is able to exactly solve this problem for larger data sets. Several algorithms have been proposed for both exact and approximate solutions to the problem. Exact algorithms are limited to solving small data sets of the QAP with massively parallel computers whereas meta-heuristics can provide near-optimal solutions within reasonable optimization times. This property of meta-heuristics has made them prominent for solving the QAP instances, therefore many researchers have proposed different heuristics or hybrid approaches to solve this problem. In this section, a summary of the successful approaches from the literature was presented.

2.1. Exact solution methods

Branch and Bound (BB) algorithms are the most elegant approaches to solve the QAP exactly (Carraresi and Malucelli, 1992; Gilmore, 1962). Even though BB algorithms were reported to produce exact solutions up to an instance size of 36, they require much more computational power and time than the heuristic algorithms while producing the same optimal results. Besides, the QAP is a NP-hard problem resulting in an exponential increase in computational power and execution time particularly for instance sizes larger than 20. The exact solution methods for a QAP become intractable for sizes larger than 36. Therefore, the BB algorithms were not used in this study. The QAP instance sizes from 36 to 256 were of special interest in this paper, which are important for the applications like placement of circuit elements onto an electronic board (Burkard and Cela, 1999), keyboard design (Steinberg, 1961), image synthesis (Taillard, 1995), and distributed database design (Adl and Rankoohi, 2009; Tosun, 2014).

Pardalos et al. (1994) gives a detailed overview of BB algorithms. Until 1990 an exact solution of the QAP instances of size 20 was not possible. Mautor and Roucairol (1994) gave the exact solutions for nug16, els19, and the problem of the size-20 instance. Clausen and Perregaard (1997) solved the instance of nug20 with a parallel BB algorithm. Marzetta and Brungger (1999) solved the nug25 instance by parallel dynamic programming. Anstreicher and Brixius (2001) developed a convex quadratic programming relaxation within a BB algorithm and this algorithm provided an exact solution for the nug25 instance after 13 days of CPU time using sequential processing. Hahn and Krarup (2001) solved the kra30a after 99 days of work with a sequential workstation. Nystrom (1999) gave the optimal solution for the ste36b and ste36c instances after 200 days of work in a distributed environment. Anstreicher et al. (2002) reported the exact solution of the nug30, which required seven days with 650 processors roughly equal to 7 years of computation on a single CPU. Even though the powerful GPU hardware of NVIDIA and CUDA platform enabled the design of efficient parallel algorithms, there is not any known bound for the instance size that can be solved using an exact method on a GPU environment. However, there are several recent works for the solution of the QAP using GPU and TS variants. Zhu et al. (2010) designed a CUDA implementation which is called SIMD-TS. The SIMD-TS employs a non-cooperative parallelism to execute independent TS instances. Czapinski (2013) also proposed a parallel multi-start tabu search for the QAP using the CUDA platform. The algorithm showed considerable speed-up in a GPU environment using up to 30 TS instances.

2.2. Genetic algorithms

GAs were first introduced by Holland (1975) in the early 1970s and they have been widely used since then. The robustness and