Engineering Science and Technology, an International Journal 21 (2018) 843–849

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Full Length Article

A cooperative GPU-based Parallel Multistart Simulated Annealing algorithm for Quadratic Assignment Problem

Emrullah Sonuc [a,*], Baha Sen [b], Safak Bayir [a]

[a] Department of Computer Engineering, Karabuk University, Turkey
[b] Department of Computer Engineering, Yildirim Beyazit University, Turkey

ARTICLE INFO

Article history:
Received 9 May 2018
Revised 1 August 2018
Accepted 1 August 2018
Available online 14 August 2018

Keywords:
Combinatorial optimization
CUDA
GPU
Multistart Simulated Annealing
Parallel algorithms
Quadratic Assignment Problem

ABSTRACT

GPU hardware and CUDA architecture provide a powerful platform to develop parallel algorithms. Implementation of heuristic and metaheuristic algorithms on GPUs are limited in literature. Nowadays developing parallel algorithms on GPU becomes very important. In this paper, NP-Hard Quadratic Assignment Problem (QAP) that is one of the combinatorial optimization problems is discussed. Parallel Multistart Simulated Annealing (PMSA) method is developed with CUDA architecture to solve QAP. An efficient method is developed by providing multistart technique and cooperation between threads. The cooperation is occurred with threads in both the same and different blocks. This paper focuses on both acceleration and quality of solutions. Computational experiments conducted on many Quadratic Assignment Problem Library (QAPLIB) instances. The experimental results show that PMSA runs up to 29x faster than a single-core CPU and acquires best known solution in a short time in many benchmark datasets.

1. Introduction

Quadratic Assignment Problem (QAP) was mathematically modelled by Koopsman and Beckman in 1957 and applied in the field of economical activities [1]. QAP is one of the NP-hard combinatorial optimization problems. The problem aims at assigning n number of facilities to n number of locations with minimum cost. When the problem solving is achieved, each facility will be assigned to a location and any facility should not be left empty. As a permutation problem, QAP can be mathematically formulated as follows:

$$cost(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j} \cdot f_{\pi(i)\ ,\ \pi(j)} \tag{1}$$

where d is an nxn dimensional matrix at which the distances between the locations are kept and f is annxndimensional matrix in which the flow cost between the facilities are kept. The aim is to find the permutation array that may obtain the minimum value of cost function.

QAP was used in determining the number of connections among the backboard wiring components [2], scheduling problem [3], designing of a typewriter keyboard, and control panel [4] in the fields of archaeology [5] and numerical analysis [6]. Along with them, its most common usage field is seen as facility layout problem. Dickey and Hopkins used QAP in the settlement of buildings in the university campus [7], Elshafei used QAP in the layout plan of hospital [8] and Bos used QAP in a problem relevant to zoning of forest parks [9].

As can be seen in the solution methods of various NP-hard problems, the exact methods are not able to find optimum or near-optimum results within a reasonable time in QAP solution as the size of problem increases. The exact solution methods offered for QAP in the literature studies are applicable for the methods of which dataset size belonging to the problem is $n \leqslant 30$. Therefore, heuristic and metaheuristic methods are applied to QAP. Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony, Scatter Search, Particle Swarm Optimization, and Memetic Algorithm may be given as examples for the metaheuristic methods used in solution of QAP. Along with them, there are also hybrid studies at which different metaheuristic methods have been used together. According to the studies on QAP, the most popular algorithms which are applied on QAP, are Simulated Annealing, Tabu Search and Ant Colony. More detailed information is given in the surveys concerning to QAP [10,11].

* Corresponding author.
E-mail addresses: esonuc@karabuk.edu.tr (E. Sonuc), bsen@ybu.edu.tr (B. Sen), safakbayir@karabuk.edu.tr (S. Bayir).

Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2018.08.002
2215-0986/

Since metaheuristic methods are computationally expensive, parallel algorithms must be designed. Various metaheuristic methods are parallelized on CPU or GPU. In the studies concerning to GPU, CUDA (Compute Unified Device Architecture) architecture was used and implemented in different fields. CUDA is the parallel computation platform and programming model developed by NVIDIA. It was developed on GPU hardware and has high capacity computing capability [12].

Multistart techniques allow to start problem solving with different starting permutations used in the serial and parallel studies with the aim of increasing the efficiency of metaheuristic methods. The general aim for these studies is to develop a method using a multi-start technique to obtain good quality results in a reasonable time.

In this paper, Simulated Annealing algorithm was parallelized on GPU using CUDA platform for solving QAP. It was aimed to obtain an optimum faster result by ensuring that the threads started with a different permutation array (multi-start). It was aimed to develop an efficient method at which both intra-block and inter-block communication were achieved between the threads. Related work regarding GPU-based optimization algorithms and metaheuristics for QAP is considered in Section 2; the information concerning the developed method is given in Section 3; the implementation of method on GPU is given in Section 4; the results obtained in this study are shown and interpreted in Section 5; and finally, Section 6 states some conclusions and future work.

2. Related work

Dokeroglu and Cosar [13] developed a method called Multistart hyper-heuristic algorithm that operates parallel on CPU and applied the method on QAP. In their study, Genetic Algorithm is run on the grid at first phase and then several heuristic mechanisms are used for improving solution quality. They highlighted the performance on the results by emphasizing the multistart method. According to their experiments, improvement in the average deviation is %27 when the multistart mechanism is used. Ferreiro et al. [14] presented a parallel algorithm on CUDA with Simulated Annealing method. They have presented both asynchronous and synchronous parallel version in their study. According to their experiments on the mathematical test functions, the synchronous version has better performance in terms of accuracy and computational cost. Sonuc et al. have presented a parallel Simulated Annealing algorithm for solving 0–1 Knapsack Problem [15] and Weapon-Target Assignment Problem [16]. Both studies are developed on GPU and according to the results, speedup is achieved 7x to 16x for 0/1 Knapsack Problem and 92x to 250x for Weapon-Target Assignment Problem. Tsutsui and Fujimoto [17] applied Ant Colony and Tabu Search methods on CUDA and presented a hybrid algorithm called Move-Cost Adjusted Thread Assignment. Acceleration has been used for comparison between GPU and CPU in their study. Paul [18] presented a parallel version of Simulated Annealing using CUDA on GPU and applied on QAP formula. In this study, comparison is performed only in terms of acceleration. According to their experiments, they found up to 100 times better than non-parallel version for acceleration. James et al. [19] proposed a Cooperative Parallel Tabu Search (CPTS) for QAP and focused on average percentage deviation. According to their experiments, CPTS is shown to meet or exceed the average solution quality of many of approaches in the literature. Czapiński [20] presented a study by using Parallel Multistart Tabu Search (PMTS) method on CUDA for QAP and used acceleration and relative percentage deviation as the comparison criterion. According to their experiments, PMTS runs up to 420x faster than a single-core CPU. Chaparala et al. [21] solved the QAP using parallel algorithm that employs a 2-opt heuristic on GPU and used the well-known QAPLIB datasets for performance comparison. Their solution provides effective speedups and gives a small penalty in terms of accuracy. Novoa et al. [22] developed a parallel tabu search algorithm to solve the QAP on GPU and compared the results with their studies at past using runtime and gap. A survey about QAP optimization algorithms applied on GPU can be found in [23].

3. The proposed method

In this section, the stages relevant to implementation of Simulated Annealing method on GPU are mentioned. Sequential Simulated Annealing method is shortly mentioned in Section 3.1; the parallel implementation of the method is presented and schematically described in Section 3.2.

3.1. Sequential Simulated Annealing

Simulated Annealing has been developed by Kirkpatrick and Vecchi [24] in 1983 with the aim of finding global minimum or maximum point of function belonging to the problems that have more than one local minimum or maximum points. The status of a physical system is used to solve a method problem. A solid substance is melted and heated up to a specific heat temperature. Then, it is performed for cooling slowly. Within the process of this annealing procedure, the aim is to ensure that the substance reaches its the best form with a smooth crystallization. The energy of the substance represents the cost function of the problem. Metropolis [25] criterion is used to accept or reject for neighbor solution. If the difference between two energies is considered as ΔE, the acceptance criterion is defined as below:

$$P = \exp\left(-\frac{\Delta E}{T}\right) \tag{2}$$

where T is potential temperature belonging to current iteration of the method, and P is the acceptance probability of each neighbor solution the annealing process. The temperature is decreased to a target temperature by cooling factor. When it reaches a target temperature, the method is stopped. Except for that, an iteration number or time dependent termination methods for Simulated Annealing can be used.

Simulated Annealing method was used for the first time in the solution of QAP by Burkard and Rendl [26] in 1984. Along with that, various studies have been done in which Simulated Annealing method is used for solution of QAP [27–29]. The pseudocode of Simulated Annealing algorithm is as follows (See Fig. 1).

3.2. Parallel multistart Simulated Annealing (PMSA)

Use of the multistart technique on metaheuristic methods provides good quality results for solution of combinatorial problems [30]. As in metaheuristic methods, multistart technique is also used for Simulated Annealing algorithm in [20,31]. Besides, multistart, shown in Fig. 2, is a very efficient technique for parallel algorithms and it is commonly used in the literature [20]. Each thread starts with own sequence for solving the problem and communicates to share results in periodically during this process. After sharing results, the threads begin a new iteration with new sequences taken from the thread having the best quality result.

In PMSA, the threads in the blocks on GPU communicate with the threads in its blocks at first. The synchronization mechanism between the threads in CUDA platform occurs only between the threads located in the same block using shared memory. According