Engineering Applications of Artificial Intelligence 52 (2016) 10–25

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A novel multistart hyper-heuristic algorithm on the grid for the quadratic assignment problem

Tansel Dokeroglu [a,*], Ahmet Cosar [b]

[a] Computer Engineering Department, Turkish Aeronautical Association University, Ankara, Turkey
[b] Computer Engineering Department, Middle East Technical University, Universities Street, Ankara, Turkey

A R T I C L E I N F O

Article history:
Received 28 August 2015
Received in revised form
24 December 2015
Accepted 7 February 2016
Available online 24 February 2016

Keywords:
Hyper-heuristics
Assignment
Simulated annealing
Tabu search
Ant colony
Breakout local search

A B S T R A C T

Hyper-heuristics introduce novel approaches for solving challenging combinatorial optimization problems by operating over a set of low level (meta)-heuristics. This is achieved by an evolutionary selection mechanism that controls and combines the strengths of the low level (meta)-heuristics. In this study, we propose a high-performance MultiStart Hyper-heuristic algorithm (MSH-QAP) on the grid for the solution of the Quadratic Assignment Problem (QAP). MSH-QAP algorithm makes use of state-of-the-art (meta)-heuristics, Simulated Annealing (SA), Robust Tabu Search (RTS), Ant Colony Optimization (FAnt), and Breakout Local Search (BLS) that have been reported among the best performing algorithms for the solution of difficult QAP instances in standard benchmark libraries. In the first phase of the algorithm, the most appropriate (meta)-heuristic with its near-optimal parameter settings is selected by using a genetic algorithm optimization layer that uses a self-adaptive parameter setting method for the given problem instance. In the second phase, if an optimal solution cannot be found, selected best performing (meta)-heuristic (with its finely adjusted parameter settings) is executed on the grid using parallel processing and performing several multistarts in order to increase the quality of the discovered solution. MSH-QAP algorithm is tested on 134 problem instances of the QAPLIB benchmark and is shown to be able to solve 122 of the instances exactly. The overall deviation for the problem instances is obtained as 0.013% on the average.

1. Introduction

The Quadratic Assignment Problem (QAP) is an NP-Hard combinatorial optimization problem introduced by Koopmans and Beckmann in 1957 to model the location selection problem of indivisible economic activities (Koopmans and Beckmann, 1957). Although facility location is the most popular form of the QAP, traveling salesman, bin-packing, maximum clique, scheduling, the graph-partitioning problem, statistical data analysis, minimum-inbreeding seed orchard layout, signal processing, transportation systems, typewriter keyboard design, layout design, backboard wiring, and data allocation are among the possible applications of the QAP (Lstiburek et al., 2015; Burkard et al., 1991; Steinberg, 1961; Rossin et al., 1999; Pfister, 1998; Dokeroglu, 2015).

The QAP is the problem of assigning facilities to locations with a varying installation costs for each location. The objective of the problem is to find an allocation such that the total cost is of installation and transporting required amounts of materials between the 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.

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

* Corresponding author.
E-mail address: tansel@ceng.metu.edu.tr (T. Dokeroglu).

http://dx.doi.org/10.1016/j.engappai.2016.02.004

where b_{jl} is the distance (i.e, the transportation cost) from location j to location l.

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

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

The Koopmans–Beckmann form of the QAP can be written as:

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

where S_n is permutation of integers 1,2,…, n. Each term $a_{ik}b_{\phi(i)\phi(k)}$ is the transportation cost from facility i at location ϕ (i) to facility k at location ϕ (k). Each term $c_{i\phi(i)}$ is the total cost for installing facility i, at location $\phi(i)$, plus the transportation costs to all other facilities k, installed at locations $\phi(1), \phi(2), \ldots, \phi(n)$ (the range of the indexes $i, j, k\ l$, is 1,…,n). The QAP (A, B) is an instance where A, B, and C are input matrices given with Eqs. (1), (2), (3). If there is no C term, we can write it as a QAP (A, B).

The QAP instances larger than size 35 cannot be solved with exact algorithms due to the computational limitations (Loiola et al., 2007). (Meta)-heuristic approaches produce high-quality solutions under these conditions with their high performances (Tseng and Liang, 2005; Drezner, 2005; Fescioglu-Unver and Kokar, 2011; Misevičius, 2012; Duman et al., 2012). Genetic Algorithms (GA) (Tosun et al., 2013), Simulated Annealing (SA) (Connolly, 1990), Ant Colony Optimization (Dorigo and Di Caro, 1999; Gambardella et al., 1999), Tabu Search (TS) (Glover, 1990; Taillard, 1991), and Breakout Local Search (Benlic and Hao, 2013) are some of these well-known methods that have been successfully applied to the QAP.

In this study, we propose a novel two-phase high-performance Multistart Hyper-heuristic Algorithm (MSH-QAP) on the grid for the QAP. MSH-QAP makes use of an emerging approach, hyper-heuristics, a selection mechanism that controls and combines the strengths of several heuristics to find the best solution for an optimization problem (Burke et al., 2013). Due to the No Free Lunch (NFL) theorem, heuristics do not demonstrate the same performances when the domain and/or the structure of the problem is changed (Wolpert and Macready, 1997). We think that applying more than a single heuristic to the same problem with parallel computation is a good idea where most probably one of the proposed heuristics will have a better performance than the others. MSH-QAP algorithm uses the state-of-the-art (meta)-heuristics, SA, Robust Tabu Search (RTS), Fast Ant System (FAnt), and Breakout Local Search (BLS) that have been reported among the best performing algorithms for large problem instances of the QAP (Taillard, 1998; Benlic and Hao, 2013). MSH-QAP is a two phase algorithm. In the first phase of the algorithm, a GA layer selects the best heuristic and tunes the parameters of the selected heuristic adaptively while trying to find the best solution for the given QAP. If an optimal solution is not found in the first phase, then in the second phase, the selected best heuristic is run on several processors by applying a multistart technique. This phase of the algorithm behaves as a stagnation prevention mechanism and restarts the exploration of the search space from different starting points and attempts to further improve the quality of the solutions. MSH-QAP also benefits from the high performance capabilities of a parallel computing environment by running the time-consuming calculations of each heuristic on a different processor. It explores the search space and uses the delta calculation approach for the fitness evaluation of the neighbors, which is a very efficient way of reducing the computation time (Misevičius, 2012).

We can summarize the contributions of our study as follows. A novel parallel multistart hyper-heuristic algorithm is proposed for the intractable QAP. The proposed MSH-QAP algorithm significantly reduces the total execution time of the optimization while improving the solution quality of the problems. The reduction is obtained through parallel execution of the heuristics on the grid. State-of-the-art heuristics SA, RTA, FAnt, and BLS are used as low-level heuristics for the solution of the QAP. The MSH-QAP algorithm has an adaptive parameter setting mechanism for each heuristic with respect to the given problem instance. MSH-QAP algorithm obtains 122 of the benchmark problems optimally while producing only 0.013% deviation from the best known results for the remaining 12 problems.

In Section 2, related studies for the state-of-the-art hyper-heuristic algorithms and the QAP are given. Our proposed algorithm, MSH-QAP, new genetic operators, the low-level heuristics SA, RTS, FAnt, and BLS that are used in the proposed algorithm are briefly explained in Section 3. The setup of the experimental environment, obtained results, and comparison with state-of-the-art (meta)-heuristics are presented in Section 4. Concluding remarks are provided in the last section.

2. Related work

Several optimization problems have been reported to be successfully solved by using hyper-heuristics. In this section, we give information about the most important ones that are related with combinatorial optimization problems. We also give short information about Genetic Algorithms (GA) and the state-of-the-art QAP solution algorithms. Finally we make a comparison between the existing solutions and our proposed algorithm.

The term hyper-heuristic was first described as a technique to combine different artificial intelligence techniques for improving the performance of automated theorem proving systems in 1997 (Cowling et al., 2000). Contemporary use of hyper-heuristics involves a set of (meta)-heuristics that are used for solving NP-hard search problems. Heuristics are selected and adapted for each problem instance, automatically, by using a selection algorithm. By using hyper-heuristic techniques, general applicability of heuristic search methods is improved without requiring intervention by a human expert to adjust the parameters of employed search heuristics (Burke et al., 2013, 2003, 2010; Ochoa et al., 2012). The hyper-heuristics differ from (meta)-heuristics by performing a search within the search space of heuristics, while (meta)-heuristics search within the space of problem solutions. Thus, a hyper-heuristic works by selecting the best search method or sequence of heuristics for a given problem instance (Burke et al., 2013; Ochoa et al., 2012; Ryser-Welch and Miller, 2014).

Burke et al. report on two distinct timetabling and rostering problems using three randomly prepared sets of problem instances and investigate the performance of a tabu-search based hyper-heuristic (Burke et al., 2003). This study uses reinforcement learning principles to come up with rules that will competitively evaluate and choose the most preferable heuristic. In order to prevent some heuristics from being chosen for a certain period a tabu list of heuristics is maintained. The work successfully shows that for various problem instances acceptable quality solutions can be obtained by using a tabu-search hyper-heuristic. Burke et al. propose a study of a simple generic