Information Processing Letters 116 (2016) 513–520

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A Hybrid Biased Random Key Genetic Algorithm for the Quadratic Assignment Problem

Eduardo Lalla-Ruiz *, Christopher Expósito-Izquierdo, Belén Melián-Batista, J. Marcos Moreno-Vega

Department of Computer and Systems Engineering, Universidad de La Laguna, Spain

ARTICLE INFO

Article history:
Received 12 September 2014
Received in revised form 17 January 2016
Accepted 3 March 2016
Available online 18 March 2016
Communicated by X. Wu

Keywords:
Quadratic Assignment Problem
Biased Random Key Genetic Algorithm
Metaheuristic
Approximation algorithms

ABSTRACT

The Quadratic Assignment Problem (QAP) is a well-known $\mathcal{NP}$-hard combinatorial optimization problem that has received a lot of attention from the research community since it has many practical applications, such as allocation of facilities, design of electronic devices, etc. In this paper, we propose a hybrid approximate approach for the QAP based upon the framework of the Biased Random Key Genetic Algorithm. This hybrid approach includes an improvement method to be applied over the best individuals of the population in order to exploit the promising regions found in the search space. In the computational experiments, we evaluate the performance of our approach on widely known instances from the literature. In these experiments, we compare our approach against the best proposals from the related literature and we conclude that our approach is able to report high-quality solutions by means of short computational times.

1. Introduction

The Quadratic Assignment Problem (QAP) is a combinatorial optimization problem introduced by Koopmans and Beckman [14]. Input data for the QAP are a set of facilities denoted as $\mathcal{F} = \{1, 2, \ldots, n\}$ and a set of locations denoted as $\mathcal{L} = \{1, 2, \ldots, n\}$. Each pair of facilities, $(i, j) \in \mathcal{F}$, requires a certain flow, denoted as $f_{ij} \geq 0$. The distance between the locations $k, l \in \mathcal{L}$ is denoted as $d_{kl} \geq 0$. It should be mentioned that the flows and distances are symmetric (*i.e.*, $f_{ij} = f_{ji}, \forall i, j \in \mathcal{F}$ and $d_{kl} = d_{lk}, \forall k, l \in \mathcal{L}$) and the flow/distance between a given facility/location and itself is zero (*i.e.*, $f_{ii} = 0, \forall i \in \mathcal{F}$ and $d_{kk} = 0, \forall k \in \mathcal{L}$).

The objective of the QAP is to minimize the cost derived from the distance and flows among facilities. This can be formally expressed as minimizing the following expression:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\phi(i)\phi(j)}, \tag{1}$$

where ϕ is a solution belonging to the set composed of all the feasible permutations, denoted as $\mathcal{S}_n$, such that $\phi : \mathcal{F} \rightarrow \mathcal{L}$. The cost associated to assign facility i to location $\phi(i)$ and facility j to facility $\phi(j)$ is, according to Equation (1), $f_{ij} d_{\phi(i)\phi(j)}$. In addition, let us denote as $f(\phi)$ the objective function value of solution $\phi \in \mathcal{S}_n$. A comprehensive description of the QAP is provided by Burkard et al. [2].

The QAP is known to belong to the $\mathcal{NP}$-hard class (Sahni and Gonzalez [17]). In fact, there are no exact methods in the literature which can tackle the QAP in medium scenarios ($n > 25$) by means of reasonable computational times. Nowadays, its hardness and heterogeneous applications turn the QAP a challenging problem within the

* Corresponding author.
E-mail addresses: elalla@ull.es (E. Lalla-Ruiz), cexposit@ull.es (C. Expósito-Izquierdo), mbmelian@ull.es (B. Melián-Batista), jmmoreno@ull.es (J.M. Moreno-Vega).

http://dx.doi.org/10.1016/j.ipl.2016.03.002
0020-0190/

optimization field. Additionally, many well-known problems such as the Travelling Salesman Problem or Graph Partitioning can be formulated as the QAP. In this context, the QAP has served as proving ground for algorithmic proposals over the last decades. Exact methods have been proposed by Fedjki and Duffuaa [4] and Erdoğan and Tansel [10]. Drezner et al. [7] review the applicability of widespread metaheuristics from the literature to address the QAP. The interested reader is referred to the detailed survey provided by Loiola et al. [15].

There are many practical applications of the QAP in the literature. For instance, Duman and Or [8] discuss how to carry out the sequencing of placement and configuration of feeder in printed circuit boards. Cheng et al. [5] model the passenger walking distance in airports according to the passenger transfer volume between aircrafts and distance between gates. Finally, Wu et al. [20] describe an application within the field of coding theory.

The remainder of this paper is organized as follows. Section 2 describes the Hybrid Biased Random Key Genetic Algorithm proposed to address the QAP. Afterwards, Section 3 analyzes the performance of our proposal in realistic scenarios. Finally, Section 4 draws forth the main conclusions extracted from the work and suggests several directions for further research.

2. Hybrid Biased Random Key Genetic Algorithm

Genetic Algorithms (GAs) are bio-inspired algorithms based upon the concepts of biological evolution and survival of the fittest individuals (Holland [13]). One of the major drawbacks of GAs is the difficulty to maintain feasibility through successive generations. With the goal of avoiding this fact, Bean [1] introduced the concept of *random key*. A random key is a real-valued number defined in $[0, 1)$, whereas a random key vector is an element of the $[0, 1)^{\rho}$ space, where ρ depends on the dimension of the optimization problem at hand. For instance, $\rho = n$ when addressing the Quadratic Assignment Problem (QAP).

A Random Key Genetic Algorithm (RKGA) is a variant of GA in which the chromosomes are random key vectors. The reproduction is performed by copying a subset of elite individuals (*i.e.*, those individuals with the lowest objective function value) from the current population to the next one. In this case, the parameterized uniform crossover suggested by Spears and De Jong [18] is used as crossover strategy. This strategy involves tossing a biased coin for each gene in order to determine which parent contributes to the corresponding gene of the relevant offspring solution. Finally, a set of random individuals is included into the current population during the mutation phase.

A variation of RKGA was presented by Ericsson et al. [11], in which one parent is selected from the set of elite individuals and the other one from the rest of the population when applying the crossover operator. In this case, a biased coin favouring the elite parent is tossed during the crossover. Although this specialized version of the RKGA was proposed as a heuristic to solve a particular problem (*i.e.*, the Weight Setting Problem), it contained the germ of what in the subsequent paper by Gonçalves and Resende [12] would be identified as a general-purpose metaheuristic: the Biased Random Key Genetic Algorithm (BRKGA).

Algorithm 1: Hybrid Biased Random Key Genetic Algorithm.

Require: $\mathcal{G}$, number of generations
Require: t, size of the population
Require: e, number of elite individuals in the population
Require: m, number of mutant individuals in the population
Require: α, crossover rate
Require: n, number of facilities in the QAP
Ensure: Best solution found for the QAP
1: Create $\mathcal{P}(1)$ with random key vectors composed of n random keys by means of Solution Generator Procedure
2: Evaluate the fitness of each individual in $\mathcal{P}(1)$
3: **for** $(g = 2 \ldots \mathcal{G})$ **do**
4: $\quad \mathcal{P}(g) = \mathcal{P}_e(g-1)$
5: $\quad$ Apply improvement method over each individual included into $\mathcal{P}(g)$
6: $\quad$ Include m mutant individuals in $\mathcal{P}(g)$ with Solution Generator Procedure
7: $\quad$ **while** $(|\mathcal{P}(g)| \leq t)$ **do**
8: $\quad\quad rk_1 \leftarrow$ Select an individual at random from $\mathcal{P}_e(g-1)$
9: $\quad\quad rk_2 \leftarrow$ Select an individual at random from $\mathcal{P}(g-1) \setminus \mathcal{P}_e(g-1)$
10: $\quad\quad rk \leftarrow \text{Crossover}(rk_1, rk_2, \alpha)$
11: $\quad\quad \mathcal{P}(g) = \mathcal{P}(g) \cup \{rk\}$;
12: $\quad$ **end while**
13: $\quad$ Evaluate the fitness of each individual in $\mathcal{P}(g)$
14: **end for**
15: **return** Best solution in $\mathcal{P}(\mathcal{G})$

A BRKGA evolves a fixed-size population, denoted as $\mathcal{P}(g) = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_t\}$, composed of t random key vectors for each generation $g = 1, 2, \ldots, \mathcal{G}$. The objective function values of the individuals included into the population determine a partition: $\mathcal{P}(g) = \mathcal{P}_e(g) \cup \mathcal{P}_c(g)$, $g = 1, 2, \ldots, \mathcal{G}$ (where $t = e + c$). In this regard, $\mathcal{P}_e(g) \subset \mathcal{P}(g)$ is termed *elite population* and composed of the elite individuals, whereas $\mathcal{P}_c(g) \subset \mathcal{P}(g)$ is termed *non-elite population* and contains the remaining individuals. Each random key vector, $\mathcal{P} \in \mathcal{P}(g)$, is mapped at the solution space of the optimization problem by means of a deterministic procedure termed *decoder*, denoted as $d : \mathcal{P} \to \phi$ (see Section 2.2). This way, a random key vector, $\mathcal{P}$, is decoded to a feasible solution of the optimization problem, $\phi \in \mathcal{S}_n$. Once the solution is decoded into the problem space, its fitness value, $f(\phi)$, is computed. The evolutionary dynamics of a BRKGA are as follows. At each generation g, all the elite individuals are copied from the current population $\mathcal{P}(g)$ (without any change) to the population of the next generation, $\mathcal{P}(g+1)$. Afterwards, a set $\mathcal{P}_m(g+1)$ of mutant individuals is inserted into $\mathcal{P}(g+1)$ with the goal of diversifying the search.

In this work, we propose a Hybrid Biased Random Key Genetic Algorithm (HBRKGA) approach in order to solve the QAP. Its pseudocode is depicted in Algorithm 1. It takes as parameters the number of generations, $\mathcal{G}$, the size of the population, t, the number of elite individuals, e, the number of mutant individuals, m (where $e + m \leq t$ and $2 \times e \leq t$), the crossover rate, α, and the number of facilities involved in the QAP to be solved, n. The first step of the HBRKGA is to obtain the initial population, $\mathcal{P}(1)$ (line 1) generated by a Solution Generation Procedure consisting of generating n random keys at random.