Knowledge-Based Systems 54 (2013) 103–113

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A hybrid metaheuristic for the cyclic antibandwidth problem

Manuel Lozano [a,*], Abraham Duarte [b], Francisco Gortázar [b], Rafael Martí [c]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain
[b] Department of Computer Science, University Rey Juan Carlos, Madrid, Spain
[c] Department of Statistics and Operations Research, University of Valencia, Valencia, Spain

ARTICLE INFO

Article history:
Received 31 January 2013
Received in revised form 19 July 2013
Accepted 18 August 2013
Available online 12 September 2013

Keywords:
Artificial bee colony
Tabu search
Local search
Cyclic antibandwidth problem.

ABSTRACT

In this paper, we propose a hybrid metaheuristic algorithm to solve the cyclic antibandwidth problem. This hard optimization problem consists of embedding an n-vertex graph into the cycle C_n, such that the minimum distance (measured in the cycle) of adjacent vertices is maximized. It constitutes a natural extension of the well-known antibandwidth problem, and can be viewed as the dual problem of the cyclic bandwidth problem.

Our method hybridizes the artificial bee colony methodology with tabu search to obtain high-quality solutions in short computational times. Artificial bee colony is a recent swarm intelligence technique based on the intelligent foraging behavior of honeybees. The performance of this algorithm is basically determined by two search strategies, an initialization scheme that is employed to construct initial solutions and a method for generating neighboring solutions. On the other hand, tabu search is an adaptive memory programming methodology introduced in the eighties to solve hard combinatorial optimization problems. Our hybrid approach adapts some elements of both methodologies, artificial bee colony and tabu search, to the cyclic antibandwidth problem. In addition, it incorporates a fast local search procedure to enhance the local intensification capability. Through the analysis of experimental results, the highly effective performance of the proposed algorithm is shown with respect to the current state-of-the-art algorithm for this problem.

1. Introduction

The cyclic antibandwidth (CAB) problem consists of embedding an n-vertex graph into the cycle C_n, such that the minimum distance (measured in the cycle) of adjacent vertices is maximized [25] and is known as a NP-hard problem [32]. The CAB problem has been exactly solved for some specific classes of graphs like paths [40], cycles [40], two dimensional meshes (Cartesian product of two paths), tori (Cartesian product of two cycles), and asymptotic results are obtained for hypercube graphs [32]. Dobrev et al. [6] extended these results to the case of Hamming graphs (Cartesian product of d-complete graphs). However, because this problem is NP-hard, for most instances one must resort to metaheuristics to obtain near optimal solutions within reasonable time. To the best of our knowledge, only one of such optimization techniques has been presented for finding the cyclic antibandwidth of general graphs, the *memetic algorithm* by Bansal and Srivastava [4]. The great interest to develop more efficient optimization algorithms for solving the CAB problem led us to propose a hybrid metaheuristic that combines the effective artificial bee colony methodology (ABC) [17,18] with tabu search (TS) [13], and that also integrates a fast local search routine where the neighborhood is visited in an intelligent way.

The ABC algorithm is a new population-based metaheuristic approach inspired by the intelligent foraging behavior of honeybee swarm. In essence, it implements memory structures based on the analogy with a bee population. Inspired by the types of bees and their different behavior this methodology considers different elements in the algorithm. In particular, it consists of three essential components: food source positions, nectar-amount and three honeybee classes (employed bees, onlookers and scouts). Each food source position represents a feasible solution for the problem under consideration. The nectar-amount for a food source represents the quality of such solution (represented by an objective function value). Each bee-class symbolizes a particular operation for generating new candidate food source positions. Specifically, employed bees search the food around the food source in their memory; meanwhile they pass their food information to onlooker bees. Onlooker bees tend to select good food sources from those found by the employed bees, and then they search for food around the selected source. Scout bees are translated from a few employed bees, which abandon their food sources and search new ones. We should remark that previous studies on bee algorithms showed that they provide efficiency in solving optimization problems [43].

* Corresponding author. Tel.: +34 958244258; fax: +34 958 243317.
E-mail addresses: lozano@decsai.ugr.es (M. Lozano), abraham.duarte@urjc.es (A. Duarte), francisco.gortazar@urjc.es (F. Gortázar), rmarti@uv.es (R. Martí).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.08.026

From an algorithmic point of view, we can say that ABC is a population-based method with memory structures. Although this methodology has been recently proposed [17], the use of memory structures in optimization algorithms can be traced back to 1986 when Glover [12] introduced the TS methodology. Genetic algorithms (GAs) probably constitute one of the most successful memory-less methods based on semi-random sampling. It is a population based methodology proposed in the seventies in which solutions are selected from a population according to their fitness, to form a new population by means of some operators inspired by the natural evolution. From this perspective, if we focus on the algorithm elements, we can view ABC as a hybrid metaheuristic combining the elements of population based methods, such as GAs, and memory-based methods, such as TS. In this paper, we explore this hybrid perspective taking some advanced TS elements (such as the ejection chains) and integrating them in an population based procedure.

The rest of this paper is organized as follows. Section 2 introduces the CAB problem in detail. Section 3 gives a brief overview of the ABC algorithm and TS. Section 4 describes our hybrid ABC approach for the CAB problem. Section 5 provides an analysis of the performance of the proposed ABC and a comparison with the existing literature. Finally, Section 6 contains a summary of results and conclusions.

2. The cyclic antibandwidth problem

Let $G(V,E)$ be an undirected and unweighted graph, where V represents the set of vertices (with $|V| = n$) and E represents the set of edges (with $|E| = m$). A labeling φ of the vertices of G is a bijective function from V to the set of integers $\{1, \ldots, n\}$ where each vertex $v \in V$ receives a unique label $\varphi(v) \in \{1, \ldots, n\}$. A circular arrangement of a labeling, simply called circular labeling, arranges the vertices of the graph in a cycle C_n where the last vertex (the one with label n) is next to the first vertex (the one with label 1). Given a circular labeling φ, let us define the clockwise distance $d^+(u,v) = |\varphi(u) - \varphi(v)|$ with $(u,v) \in E$ and, similarly the counterclockwise distance $d^-(u,v) = n - |\varphi(u) - \varphi(v)|$ with $(u,v) \in E$. Then, for a given circular labeling φ, the cyclic antibandwidth of G, referred to as $CAB(G,\varphi)$, is computed as follows:

$$CAB(G,\varphi) = \min_{(u,v)\in E}\{d^+(u,v), d^-(u,v)\}. \quad (1)$$

The CAB problem consists of maximizing the value of $CAB(G,\varphi)$ over the set Π of all possible labelings:

$$CAB(G) = \max_{\varphi\in\Pi} CAB(G,\varphi). \quad (2)$$

Note that in optimization terms, any labeling φ of G is a solution of the CAB problem stated in (2), with an objective function value, simply called value or $CAB(G,\varphi)$, defined in (1). The optimal solution(s) is therefore the labeling, or labelings, with maximum value.

Fig. 1-left shows an example of a graph G with 8 vertices and 8 edges. Fig. 1-right shows a circular labeling φ of G, arranging the vertices in a cycle. Additionally, clockwise (d^+) and counterclockwise (d^-) distances for each edge are shown in Table 1. In particular, each row of this table reports the distance between each pair of adjacent vertices (those joined with an edge). For example, the clockwise distance between vertex A and B is $d^+(A,B) = |\varphi(A) - \varphi(B)| = |7 - 8| = 1$. Similarly, the counterclockwise distance between these two vertices is $d^-(A,B) = 8 - |\varphi(A) - \varphi(B)| = 8 - |7 - 8| = 7$. In order to compute $CAB(G,\varphi)$, we evaluate d^+ and d^- for the remaining 7 edges (shown in Table 1), reporting the minimum of all of them. Therefore, $CAB(G,\varphi) = 1$.

CAB is a natural extension of the antibandwidth problem [3,7]. Although these two optimization problems are related, we should not expect a method developed for the former problem to perform well on the latter. We illustrate this fact by considering the example shown in Fig. 2, which corresponds to a caterpillar $P_{5,4}$ graph. A caterpillar P_{n_1,n_2} is constructed using the path P_{n_1} and n_1 copies of the path P_{n_2}, where each vertex i in P_{n_1} is connected to the first vertex of the i-th copy of the path P_{n_2}. For such instance we apply the RBFS constructive procedure by Bansal and Srivastava [4] to generate 10^6 labelings (solutions) and compute for each one the objective function value for the cyclic antibandwidth (CAB) according to (1), and the objective function value of the antibandwidth problem (AB) according to Duarte et al. [7]. The correlation between both values computed over the 10^6 solutions is rather small ($r = 0.13$). In addition, the labeling with maximum AB value, out of the 10^6 generated, is

$$(0, 12, 3, 16, 6, 10, 1, 11, 2, 13, 4, 14, 5, 15, 7, 19, 9, 17, 8, 18),$$

while the labeling with maximum CAB value is

$$(16, 5, 14, 2, 11, 7, 19, 9, 17, 8, 18, 4, 15, 6, 12, 3, 13, 1, 10, 0).$$

We can see that both labelings are very different.

The CAB problem was proved to be NP-hard in Raspaud et al. [32]. This problem was originally introduced in Leung et al. [25] in connection with multiprocessor scheduling problems. It has been found to be relevant in allocating time slots for different sensors in a network such that two sensors that interfere each other have a large time interval between their periods of operation

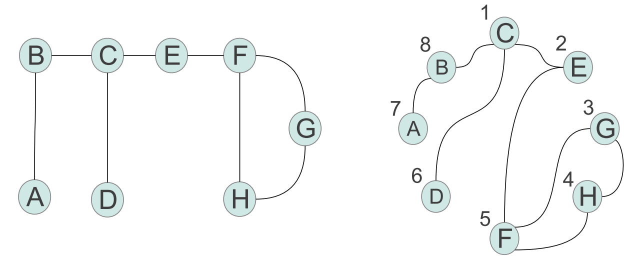

Fig. 1. A graph and a circular labeling layout.

Table 1
Clockwise and counterclockwise distances for the graph depicted in Fig. 1.

d^+	d^-
$d^+(A,B) = \|7-8\| = 1$	$d^-(A,B) = 8 - \|7-8\| = 7$
$d^+(B,C) = \|8-1\| = 7$	$d^-(B,C) = 8 - \|8-1\| = 1$
$d^+(C,D) = \|1-6\| = 5$	$d^-(C,D) = 8 - \|1-6\| = 3$
$d^+(C,E) = \|1-2\| = 1$	$d^-(C,E) = 8 - \|1-2\| = 7$
$d^+(E,F) = \|2-5\| = 3$	$d^-(E,F) = 8 - \|2-5\| = 5$
$d^+(F,G) = \|5-3\| = 2$	$d^-(F,G) = 8 - \|5-3\| = 6$
$d^+(F,H) = \|5-4\| = 1$	$d^-(F,H) = 8 - \|5-4\| = 7$
$d^+(G,H) = \|3-4\| = 1$	$d^-(G,H) = 8 - \|3-4\| = 7$

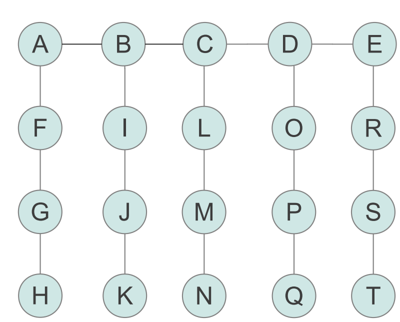

Fig. 2. A caterpillar $P_{5,4}$ graph.