Applied Soft Computing 11 (2011) 5360–5366

Contents lists available at ScienceDirect

Applied Soft Computing

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

An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem☆

Raka Jovanovic [a], Milan Tuba [b,*]

[a] Institute of Physics Belgrade, Pregrevica 118, Zemun, Serbia
[b] Faculty of Computer Science, Megatrend University Belgrade, Bulevar umetnosti 29, Serbia

ARTICLE INFO

Article history:
Received 12 November 2009
Received in revised form 22 August 2010
Accepted 2 May 2011
Available online 10 May 2011

Keywords:
Ant colony optimization (ACO)
Combinatorial optimization
Minimum weight vertex cover problem (MWVCP)
Pheromone correction strategy

ABSTRACT

The minimum weight vertex cover problem is an interesting and applicable NP-hard problem that has been investigated from many different aspects. The ant colony optimization metaheuristic is a relatively new technique that was successfully adjusted and applied to many hard combinatorial optimization problems, including the minimum weight vertex cover problem. Some kind of hybridization or exploitation of the knowledge about specific problem often greatly improves the performance of standard evolutionary algorithms. In this article we propose a pheromone correction heuristic strategy that uses information about the best-found solution to exclude suspicious elements from it. Elements are suspicious if they have some undesirable properties that make them unlikely members of the optimal solution. This hybridization improves pure ant colony optimization algorithm by avoiding early trapping in local convergence. We tested our algorithm on numerous test-cases that were used in the previous research of the same problem and our algorithm uniformly performed better, giving slightly better results in significantly shorter time.

1. Introduction

One of the classical graph theory problems is the minimum weight vertex cover problem (MWVCP). The problem is defined for an undirected graph $G = (V, E)$ where V is the set of vertexes, E is the set of edges and weights are assigned to each vertex in the graph. A vertex cover of graph G is a set of vertexes $V' \subseteq V$ that has the property that for every edge $e(v1, v2) \in E$ at least one of vertexes $v1$ or $v2$ is an element of V'. A minimum weight vertex cover is the vertex cover with the minimum sum of weights of the belonging vertexes. It has been shown that this problem is NP-complete even when it is restricted to a unit-weighted planar graph with the maximum vertex degree of three [1].

A large number of real life problems could be converted to this form. An example could be the optimal positioning of garbage disposal facilities. For such NP-hard problems, since the optimal solution is computationally intractable, research is concentrated on heuristic algorithms that can find good suboptimal solutions in a reasonable time. It has been shown that for the restricted version of the MWVCP, the so-called real-WVC, where a total weight is at most k and each weight is ≥ 1, the solution can be found in time $O(1.3954^k + kn)$ where $n = |V|$, or in $O(1.3788^k + kn)$ with modestly exponential memory use [2]. When weights are restricted to positive integers (integer-WVC) the problem can be solved as fast as unweighted vertex cover in $O(1.2738^k + kn)$, again with exponential memory use [3,4]. A variety of different methods have been utilized for calculating near optimal solutions. The first one is a greedy heuristic approach of collecting the vertex with the smallest ratio between its weight and degree. This approach was first applied for a generalized version of the set cover problem [5], and its variation for the vertex cover problem with a performance guarantee of 2 [6]. The MWVCP has also been investigated by application of the genetic algorithms combined with greedy heuristic [7] and recently by metaheuristic based on gravity [8].

The ant colony optimization (ACO) is another metaheuristic for solving combinatorial problems. It was first used for the traveling salesman problem (TSP) by Dorigo and Gambardella [9,10]. The ACO was applied to the MWVCP by Shyu et al. [11] and good results were obtained, better that those acquired by local search methods like tabu search or simulated annealing. Gilmour and Dras further analyzed the ACO for the MWVCP in the direction of finding the optimal values for ACO parameters [12] and improving the effectiveness of ACO by using the parameterized complexity concept and the kernelization as a tool for algorithm design [13]. Several other approaches have been introduced for improving the efficiency of the ACO. One such approach is the use of different types of hybridization, like addition of local searchers [14,15]. Although

☆ This work has been funded by the Ministry of Education and Science of Republic of Serbia, Project III 44006.

* Corresponding author.
E-mail addresses: rakabog@yahoo.com (R. Jovanovic), tubamilan@ptt.rs (M. Tuba).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.05.023

this is a good approach in the majority of cases, it has been shown that it may prevent the ACO from finding the optimal solution [16]. Combining the ACO with genetic algorithms has been applied to large number of problems and gave better results than separate use of these methods: Lee et al. [17] or Zuo and Xiong [18]. Both of these hybridization methods are rather complex due to the need for two algorithms, one for the ACO and another one for the local search or genetic algorithm.

A simpler method of improving the performance of the ACO is the use of variations of the basic algorithm like elitist ant colony, rank based ant colony system, min–max ant system (MMAS). We have previously performed a comparative assessment of different variations of the ACO for the MWVCP and shown that the MMAS outperforms other variations of the basic algorithm [19]. All these variations have the problem of becoming trapped in local optima. An interesting approach, named the minimum pheromone threshold strategy (MPTS), to solve the stagnation problem was proposed by Wong and See for the quadratic assignment problems [20].

In this article we propose a new type of hybridization for the ACO and implement it for the MWVCP. We improve the ACO with a hybridization method for leaving local optima i.e. avoiding stagnation in search for the best solution. This method is based on correcting the pheromone trail used in the ACO. We calculate this correction based on the properties of the best-found solution so far. The concept of this correction is to lower the possibility of vertexes with high level of undesirable properties to belong to the optimal solution. We show that our method improves results acquired by the ACO and that it is simple to add the method to the existing algorithms. It also gives better results that some other recent improvements [8].

This paper is organized as follows. In Section 2 we present the implementation of the ACO for the MWVCP. In Section 3 we review the other methods of avoiding the ACO algorithm stagnation. In the next section we explain our hybridization used for escaping search stagnation. In Section 5 we analyze and compare experimental results of using pure ACO and its combination with our method for the MWVCP and show the advantages of using our approach.

2. Application of the ACO to the MWVCP

In the ACO implementation for the MWVCP there are two significant differences compared to the other problems commonly solved by the ACO. These differences can be illustrated by a comparison to the ACO implementation for the TSP. In the TSP the solution is a permutation of the set of all the cities. In the case of the MWVCP the solution is a subset of the set of graph vertexes where the order is unimportant. In the TSP the heuristic function is static in the sense that it represents the distance between cities and it does not change during the calculation of the path. Contrary to this, for the MWVCP the heuristic function is a ratio between the weight and the temporary degree of a vertex (number of edges that are incident to a vertex but are not already covered by some other vertex that is already included in the solution subset), which is dynamic because the temporary degree of a vertex changes as new vertices are added to the solution subset and more edges become covered. These two differences affect the basic algorithm in two directions: the ants leave the pheromone on vertices instead of edges and the heuristic function has to be dynamically updated. Such ACO with a dynamic heuristic and a solution that consists of a subset instead of a permutation have also been used for solving the set partitioning [21], maximum independent set [22] and maximum clique [23] problems.

We adopt the ACO implementation introduced by Shyu et al. [11], and present it briefly in this section. The first step is to represent the problem in a way that makes dynamic calculation of the heuristic function simple. Since ants in their search can move from a vertex to any other vertex, it is natural to use a fully connected graph $G_c(V, E_c)$ derived from graph G. Weight 1 is assigned if an edge exists in G, or 0 if it does not appear in the original graph. As mentioned before, this graph has to be updated as new vertices are added to the result set. This is done in the following way: when vertex a is added, all edges in G_c that are connected to a are set to 0.

Let us define $G_{ck}(V, E_{ck},)$ as the state of the graph after k vertices have been added to the solution, and corresponding functions:

$$\psi_k(i,j) = \mathrm{Value}(E_{ck}(i,j)) \tag{1}$$

Now, a dynamic heuristic can be defined:

$$\eta_{jk} = \frac{\sum_{(i,j)\in E_c} \psi_k(i,j)}{w(j)} \tag{2}$$

In Eq. (2) $w(j)$ is the weight of the vertex j. Using the heuristic defined by η_{jk} in Eq. (2) the state transition rule for ants can be set:

$$p_j^k = \begin{cases} 1, & q > q_0 \,\&\, j = \mathrm{argmax}_{i\in A_k} \tau_i \eta_{ik}^{\alpha} \\ 0, & q > q_0 \,\&\, j \neq \mathrm{argmax}_{i\in A_k} \tau_i \eta_{ik}^{\alpha} \\ \dfrac{\tau_j \eta_{jk}^{\alpha}}{\sum_{i\in A_k} \tau_j \eta_{ik}^{\alpha}}, & q \le q_0 \end{cases} \tag{3}$$

In Eq. (3) q_0 is the standard parameter that specifies the exploitation/exploration rate, and q is a random variable that decides the type of selection on each step. A list of available vertices is A_k. Unlike the TSP transition rule, this one does not depend on the last selected vertex and that is why τ_i is used instead of τ_{ij}. To fully specify an ant colony system it remains to define the global (when ants finish their paths) and the local (when an ant chooses a new vertex) update rules:

$$\Delta\tau_i = \begin{cases} 0, & i \notin V' \\ \dfrac{1}{\sum_{j\in V'} w(j)}, & i \in V' \end{cases} \tag{4}$$

$$\tau_i = (1-p)\tau_i + p\Delta\tau_i \tag{5}$$

Parameter $\Delta\tau_i$ which represents a quality measure of the best global solution subset V' that contains vertex i is defined In Eq. (4) and subsequently used in the definition of the global update rule in Eq. (5). Parameter p is used to set the influence of a newly found solution on the pheromone trail. The formula for the local update rule has the standard form

$$\tau_i = (1-\varphi)\tau_i + \varphi\tau_0 \tag{6}$$

The quality measure of the solution acquired by the greedy algorithm (where the vertex with the best ratio of vertex degree and weight is selected) is selected for the value of τ_0. Parameter φ is used to specify the strength of the local update rule.

3. Stagnation avoidance in the ACO

The algorithm described in the previous section corresponds to the elitist ant system (EAS) variation of the ACO. In this version of the algorithm only the global best solution (or in some variations only the iteration best solution) deposits pheromone. It increases the efficiency of the basic ACO by intensifying the search near the global best solution, or in other words, making its search more greedy. One of the main problems of this version is the early stagnation of the algorithm. This is due to the fact that potential parts