TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA
1-AC/2013

KRZYSZTOF SCHIFF*

# ANT COLONY OPTIMIZATION ALGORITHM FOR THE SET COVERING PROBLEM

# ALGORYTM MRÓWKOWY DLA ZAGADNIENIA POKRYCIA ZBIORU

Abstract

This article describes a new hybrid ant colony optimization algorithms for the set covering problem. The problem is modeled by means of a bipartite graph. New heuristic patterns, which are used in order to choose a vertex to a created covering set have been incorporated into modified hybrid algorithms. Results of tests on investigated algorithms are discussed.

*Keywords: set covering problem, ant colony optimization, heuristic rule*

Streszczenie

W artykule przedstawiono nowy hybrydowy algorytm mrówkowy dla problemu zagadnienia pokrycia zbioru o minimalnym koszcie. Problem jest zamodelowany za pomocą grafu dwudzielnego. W modyfikowanym algorytmie wprowadzono nową heurystykę wyboru wierzchołków do podzbioru wierzchołków pokrywających. Opracowany algorytm przetestowano i porównano, a wyniki tych badań omówiono.

*Słowa kluczowe: zagadnienie pokrycia zbioru, algorytm mrówkowy, heurystyka*

* Krzysztof Schiff, Ph.D., Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

## 1. Introduction

Many practical optimization problems such as for example facility location problem, airline crew scheduling, nurse scheduling, vehicle routing and resource allocation problem can be described and modeled as the Set Covering Problem (SCP) [1, 2, 10–12] and many other combinatorial problems can be modeled in such a way. There are many kinds of computer algorithms that have been designed so far to solve SCP such as exact algorithms [13, 14], heuristic algorithms [15–18], meta-heuristics algorithms [19–21], also algorithms, which are based on ant colony optimization strategy [3, 6, 22, 23] and hybrid of Ant Colony Optimization Algorithm (ACO) with so called Constraint Programming (CP) [26, 27]. Ant algorithms were designed to solve many combinatorial problems [5, 8, 24, 25]. This paper presents new algorithms, which are based on an ant colony optimization strategy, for the set covering problem with a minimum covering cost and a new heuristic information patterns, which have been used in these algorithms. Transition probability rule and pheromone update rule and also a mechanism checking constraints consistency are used all together in order to solve the SCP and to minimize the total covering cost. The remainder of this paper is structured as follows: in section 2 the SCP is introduced, in section 3 the structure of ACO algorithm is described, in section 4 pseudo-code of ant algorithms with new heuristic patterns and transition probability rules, which is used in new elaborated ant algorithms, are discussed and in section 5 results of the conducted computational experiments on a special kind of a graph with an almost equal density and in section 6 conclusions are presented.

## 2. Set covering problem

The set covering problem can be modeled as a bipartite graph network $G(V_1 + V_2, E, w)$ with weights $w_{ij}$ assigned to edges $e_{ij}$, such that $e_{ij} = (v_i \in V_1, v_j \in V_2)$, $e_{ij} \in E$ as it is shown in Fig. 1 and in the same way as it is presented in [9]. The degree of vertex $i$ is the sum of edges adjacent to this vertex $i$. All vertices $v_{1i}$ can be grouped into subsets in such a way that all vertices from the set $V_2$ are covered by vertices from the some subset $V_s$ of vertices $V_1$. A vertex $v_{2i}$ is covered by a vertex $v_{1j}$ if an edge $e_{ij}$ exists between a vertex $i$ and a vertex $j$ in a bipartite graph, for example subset $V_s \subset V_1$ which consists of vertices $v_{11}$, $v_{13}$, $v_{14}$ and $v_{16}$ covers all vertices from the set $V_2$ and the another example of set $V_s$ is subset which consists of vertices $v_{14}$ and $v_{17}$. In general if the cardinality number of set $V_s$ is lower than the total set covering cost is higher. In the set covering problem with minimum the cardinality number of set $V_s$ and the cost the total cost of set covering, this means the sum of weights assigned to all graph edges, which are participating in the set covering, has to be minimized.

The objective function $F$ is to find a cover set with a minimum cost and is described in (1) and (2):

$$\min F = \sum_{i=1}^{n} \sum_{j=1}^{n} \left( x_{ij} w_{ij} \right), \; i \in V_1, \; j \in V_2 \tag{1}$$

and subject to constraint such that:

$$\sum_{i \in Vs}^{n} x_{ij} \geq 1, \; i \in V_1, \; j \in V_2 \tag{2}$$

where:

$x_{ij} = 1$, when a vertex $j$ is covered by a vertex $i$,
$x_{ij} = 0$, when a vertex $j$ is not covered by a vertex $i$,
$w_{ij}$ – this is a weight associated to edge $e_{ij}$ (a cost of covering a vertex $j$ by a vertex $i$),
$V_s$ – is a subset of vertices $V_1$ which cover all vertices $V_2$.

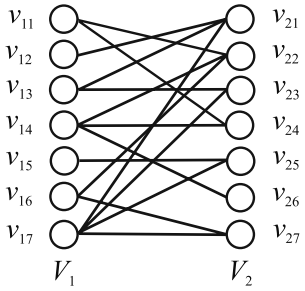


Fig. 1. The set covering problem modeled by a bipartite graph

Rys. 1. Problem pokrycia zbioru modelowany grafem dwudzielnym

## 3. Structure of ACO algorithm

In ant algorithms a colony of artificial ants is looking for a good quality solution of the investigated problem. The pseudo-code of ACO procedure is presented as algorithm 1. Each artificial ant constructs an entire solution of the problem in some number of steps, called intermediate solutions. Any of intermediate solutions are referred to as solution states. In each step $m$ of the algorithm each ant $k$ goes from a one state $i$ to an another state $j$ and thus constructs a new intermediate solution called later a partial solution of the problem since the entire solution is received in some number of steps and at each of these steps there is an intermediate solution called a partial solution or a solution under construction. At each step each ant $k$ computes a set of feasible expansions to its current state and moves to one of these in probability. This set of feasible expansions is called a neighborhood of current state. In presented algorithms concerning processing of bipartite graph, this means working on a graph model of the set covering problem at each state each ant chooses a vertex from the set $V_1$ and adds it to a partial solution in order to construct finally at the end of algorithm action the entire solution to the SCP problem. At the end of algorithm action the set of vertices $V_s$ constitute a solution to the SCP problem. Each ant $k$ starts with an empty set $V_s$ and successively adds to this set $V_s$ a vertex chosen one after the other from the set $V_1$ with probability $p^k_{ij}$ moving from a one to another state. At each state $i$ there are some vertices in the set $V_s$ and these vertices from the set $V_s$ constitute a partial solution of a problem at step $m$, this means at state $m$. Each ant in order to construct a solution uses common information which is encoded in pheromone trails, this means the trail level of the move, indicating how proficient it has been in the past to make that particular move. Each ant also deposits a pheromone on a trail when a solution has been found and a quantity of the pheromone deposited depend on a quality of this solution. The move of each ant also depends on so called the attractiveness of the move, as computed by some heuristic indicating the a priori desirability of that move. In order to avoid a very fast

convergence to a locally optimal solution an evaporation mechanism is used, this means that over the time the pheromone trail evaporates, thus reducing its attractive strength.

Algorithm 1
ACO procedure

```
begin
    while (exist cycle) do
        while (exist any ant, which has not worked) do
            while (a solution has not been completed) do
                choose a next vertex to a constructed solution with a probability p^k_ij;
                update neighborhood of current state;
            end
            update a best solution if a better solution has been found;
        end
        update a global best solution if a better solution has been found;
        use an evaporation mechanism;
        update a pheromone trails τ(i) = τ(i) + Δτ;
    end
end.
```

Each ant $k$ moves from one state $i$ to another state $j$ with a transition probability rule $p^k_{ij}(t)$, which is described by the formula:

$$p_{ij} = \left\{ \begin{array}{c} \dfrac{\left(\tau_{ij}^{\alpha}\mu_{ij}^{\beta}\right)}{\sum_{j\in N_i}\left(\tau_{ij}^{\alpha}\mu_{ij}^{\beta}\right)}, \quad \text{for} \quad j \in N_i \\ 0, \quad \text{for } j \text{ which not} \in N_1 \end{array} \right\} \tag{3}$$

using the pheromone trail $\tau_{ij}$ and the attractiveness $\mu_{ij}$ of the move. The pheromone trail $\tau_{ij}$ is the useful information, which is deposited by others ants, for each ant during its work on construction of solution, about the usage of vertex $j$ in the past by others ants. The attractiveness $\mu_{ij}$ is a desire of choosing a vertex $j$ from the neighborhood $N_i$ of current state when there is a partial solution yet constructed in state $i$ and the attractiveness $\mu_{ij}$ can by expressed by a some heuristic formula. The attractiveness $\mu_{ij}$ allows to better choose a some vertex from all vertices, from the neighborhood $N_i$ of current state, to be added to a solution under construction taking an objective function into a consideration. The neighborhood $N_i$ of state $i$ is constituted by vertices which can be added to a constructed partial solution. At the start all vertices can be added to a partial solution of the problem, this means to a solution of a problem under construction and the number of these vertices is reduced not only because of their inclusion into the solution, which is under the construction, but also because some of these vertices cannot be yet added to a solution, which is under construction, since these vertices does not satisfied solution constraints and only these vertices can be added to constructed partial solution which still satisfied solution constraints. The partial solution of the problem is a part of solution and the partial solution is a subset of vertices, which constitute a solution of the problem. Parameters $\alpha$ and $\beta$ which is used in the transition

probability rule $p^k_{ij}(t)$ expressed by (3), indicate about this, how important the pheromone trail $\tau_{ij}$ and the attractiveness $\mu_{ij}$ are during transition from one to another state. Values of these parameters $\alpha$ and $\beta$ should be set by experiment and tuned to the set covering problem with minimum covering cost.

After a solution has been found each ant deposits a pheromone with a quantity $\Delta\tau$ on all vertices, which constitute the solution $V_s$, in accordance with the pattern:

$$\tau_{ij}(t) = \tau_{ij}(t) + \Delta\tau \tag{4}$$

Thus these vertices which were included into a solution have received an additional quantity of a pheromone and can be chosen to a solution that would be constructed next with a higher probability than others vertices from the set $V_1$.

An evaporation mechanism is incorporated into an ant algorithm in order to avoid a too fast convergence to a sub-optimal solution. An intensity of evaporation is controlled by a parameter $\rho$ and a quantity of a pheromone on each vertex from the set $V_1$ is update at the end of each cycle in accordance with the pattern:

$$\tau_{ij}(t) = (1-\rho)\,\tau_{ij}(t), \quad \rho \in (0, 1] \tag{5}$$

Thus a diversity of a solution is granted. Values of a parameter $\rho$ should be set by experiment.

A quantity of deposited pheromone $\Delta\tau$ depends on a quality of solution $Q$ and if the better is a solution than the more pheromone is deposited and in general can be stated as formula:

$$\Delta\tau = f(Q) \tag{6}$$

and in particular can be expressed by some specific formula, which take into account the covering cost.

## 4. Hybrid ACO algorithm

Both ACO-SCP algorithms, which are discussed in this paper, are modified versions of the hybrid algorithm described in [6] and in this paper the general pseudo-code of these algorithms is presented as algorithm 2. In the algorithm presented in this paper a new dynamic heuristic rule is proposed. The dynamic heuristic information:

$$\mu_1(i) = \frac{\nu c}{\sum_{j=1} w_{ij}}, \ \textit{if } x_{ij} = 1 \text{ and } j \in S_2 \text{ and } j \text{ is not covered yet} \tag{7}$$

is defined in the same way as in the paper [6] and the dynamic heuristic information $\mu_2(i)$ and $\mu_3(i)$ can be defined adequately:

$$\mu_2(i) = \sum_{j=1}^{n} \left( w_{\max} - w_{ij} \right) if \left[ \left( w_{ij} - w_{kj} \right), k \in V_s \right] \tag{8a}$$

$$\mu_3(i) = \sum_{j=1}^{n} \left( w_{kj} - w_{ij} \right) if \left[ \left( w_{ij} < w_{kj} \right), k \in V_s \right] \tag{8b}$$

where:

$vc$ – this is a number of additionally covered vertices from $S_2$ if vertex $i$ would be included into a solution $V_s$,
$w_{\max}$ – this is the maximal weight from weights associated to an edges $e_{ij,}$
$w_{ij}$ – this is a weight associated to an edge $e_{ij}$,
$V_s$ – this is a constructed yet subset of $V_1$ vertices,
$k$ – this is a vertex already included into set $V_s$,
$\mu_1(i)$ – desirability of vertex $i$ when not covered vertices $j$ from the set $V_2$ are taken into consideration,
$\mu_2(i)$ and $\mu_3(i)$ – desirability of vertex $i$ when edge covered vertices $j$ from the set $V_2$ are taken into consideration,

$x_{ij} = 1$ when an edge $e_{ij}$ exists between vertex $i$ and vertex $j$ and $x_{ij} = 0$ otherwise.

In both algorithms a following vertex that should be added to a partial solution is chosen with a probability that depends on a pheromone trail, heuristic information and transition rule. Main differences between elaborated algorithms and algorithm, which is presented in the paper [6] concern a transition probability rule (9), (10) and heuristic information (8a), (8b). Since the quality of a solution depends on a total weight of covering, this means depends on a sum of weights assigned to all edges between all vertices of the set $V_2$ and the set $V_s$, the attractiveness $\mu_{ij}$ of choosing vertex $j$ expressed as a function of a weight is very important. At any state only these vertices from the set $V_1$ which can improve quality of solution should be considered when any ant choose following vertex that should be added to a partial solution and to a set of vertices $V_s$. Such vertices from the set $V_1$ which can improve quality of a solution will be called available vertices and will be constitute a set $V_A$ and these vertices will be also constitute the neighborhood $N_i$ of a current state. These vertices from the set $V_1$, which cannot improve a quality of solution are excluded as a result of consistency checking from available vertices $V_A$ and such vertices constitute a set $V_{ex}$. It is obvious that if any vertex is included into a partial solution $V_s$ it cannot belongs to a set of available vertices $V_A$, so taking the above into consideration the number of available vertices $V_A$ can be computed in accordance with expression $V_A = V_1 - V_s - V_{ex}$. The attractiveness $\mu_1(i)$ and $\mu_2(i)$ and $\mu_3(i)$ of choosing vertex $i$ from available vertices $V_A$ depend on weights of its edges and not only it concerns not covered yet vertices from the set $V_2$ expressed by the attractiveness $\mu_1(i)$, but also these vertices from the set $V_2$, which have been covered up till now, this means up to this moment of choosing from the neighborhood $N_i$ of a current state the next following vertex $i$ expressed by the attractiveness $\mu_2(i)$ and $\mu_3(i)$:

a) for HACO1-SCP:

$$p(i) = \frac{\tau(i)\mu_1(i)\mu_2(i)}{\sum_{i \in V_A} \left( \tau(i)\mu_1(i)\mu_2(i) \right)}, \; V_A \in V_1 \tag{9}$$

b) for HACO2-SCP:

$$p(i) = \frac{\tau(i)\mu_1(i)\mu_3(i)}{\sum_{i \in V_A} \left( \tau(i)\mu_1(i)\mu_3(i) \right)}, \; V_A \in V_1 \tag{10}$$

where:

$V_A$ – this is a set of available vertices, $V_A = V_1 - V_s - V_{ex}$,
$V_{ex}$ – these are vertices, which are excluded as a result of consistency checking,
$\tau(i)$ – this is a pheromone trail on a vertex $i$,
$(\mu_1(i)\ \mu_2(i))$ – this is a heuristic information associated with a vertex $i$ in part a),
$(\mu_1(i)\ \mu_3(i))$ – this is a heuristic information associated with a vertex $i$ in part b).

A quantity of pheromone $\Delta\tau$ is deposited by ants during one cycle of algorithm action on all vertices of the set $V_s$, which were included into the best constructed solution, in accordance with the formula:

$$\Delta\tau = \frac{1}{1 - \frac{c_{\text{best}} - c}{c_{\text{best}}}} \tag{11}$$

where:

$c_{\text{best}}$ – this is the best cost of covering,
$c$ – this is an actual cost of covering.

Algorithm 2
Hybrid ACO procedure for SCP

```
begin
    while (exist cycle not done) do
        for (k:=1 to n Ants) do
            while (a solution (a subset V_s) is not completed ) do
                Update Available Vertices;
                Choose next vertex i with probability p(i) and consistency checking;
                Add to a Partial Solution;
                Update Partial Solution;
            end
            Save a Better Solution;
        end
        Update Optimum;
        Use an evaporation mechanism;
        Update Pheromone;
    end
Return Best Solution Founded;
end.
```

Constraint Programming based on Edge Consistency with pre and post-processing.

An edge adjacent to a vertex $i$ from the set $V_1$ and exactly from the set $V_A$ can be added to a partial solution only if a cost (weight) of this edge is lower in comparison to a cost of any edges which are already included to a partial solution $V_S$ and when both these edges, which weights are compared, cover the same vertex j from the set $V_2$. Adjacent edges of this vertex $i$, whose costs are higher, cannot be added to any partial solution, thus to a solution of the problem.

A vertex $i$ from set $V_1$, which has been already included into a partial solution $V_s$, will be excluded from this partial solution $V_S$ only when costs (weights) all of its edges are higher in comparison to costs of other edges adjacent to vertices which are too already included into a partial solution, this means that a vertex $i$ will be excluded if this vertex $i$ has no edge, this means that its edges were with higher weights and were excluded before from constructed partial solution up till now and thus a vertex $i$ has now no edge at all and a vertex $i$ can be now excluded from a partial solution $V_s$.

A vertex $i$ from set $V_1$, which can be chosen to a partial solution $V_s$, is excluded from these available vertices to be chosen to a partial solution $V_A$ if costs of all its all edges are higher in comparison to costs of edges from a constructed yet partial solution, this means from the set $V_s$ covering these some vertices $j$ from the set $V_2$ and any of its edges cover any vertex $j$ from the set $V_2$, which is not yet included into a partial solution $V_s$.

This new modified heuristic pattern lets to receive a better solution than a solution which is received by the hybrid ACO algorithm, which is presented in [6] for a bipartite graph with almost equal degree of all vertices. The constraint programming technique used in this paper is based on the edge consistency with pre and post processing [4, 6, 7]. Thus a number of available vertices which can be potentially included into a partial solution is minimized and any no longer needed vertices are eliminated from a partial solution and also only these edges with lower weights are added to a solution under construction and these with higher weights are eliminated from a partial yet constructed solution.

## 5. Experiments

There are three algorithms which were studied during experiments. The first is the HACO algorithm, which was described in [6], the second is the HACO1-SCP algorithm with desirability $\mu_2(i)$ and the third is the HACO2-SCP with desirability $\mu_3(i)$, which are described in this paper. Two parameters were under observation during conducted experiments: an average minimum cost of set covering and an average cardinality number of the set $V_s$, which were received as a result of 10 measures. All algorithms were studied for a bipartite graph with 100 x 100 vertices and for a different graph densities $q$, which were generated in random. Later all algorithms were studied for a bipartite graph with random generated edges for each its vertex, this means with random generated vertex degree and for measure cases with different number of vertices, this means for 50 x 50, 100 x 100, 150 x 150, 200 x 200 and 250 x 250 vertices in a bipartite graph. These bipartite graphs belong to the particular kind of a bipartite graph since each edge in each of these graphs exists with a probability $q$ and thus each vertex of graph has almost equal degree, this means has almost equal number of adjacent edges. An average minimum set covering costs for 10 measures were presented in Table 1 and Fig. 2 and an average minimum cardinality numbers of the set $V_s$ for 10 measures were presented in Table 2 and Fig. 3. These all three algorithms ran with the following common parameters setting for each measure cases: the evaporation rate was set to 0.995, the number of ants was set to 200 and the number of cycles was set to 300.

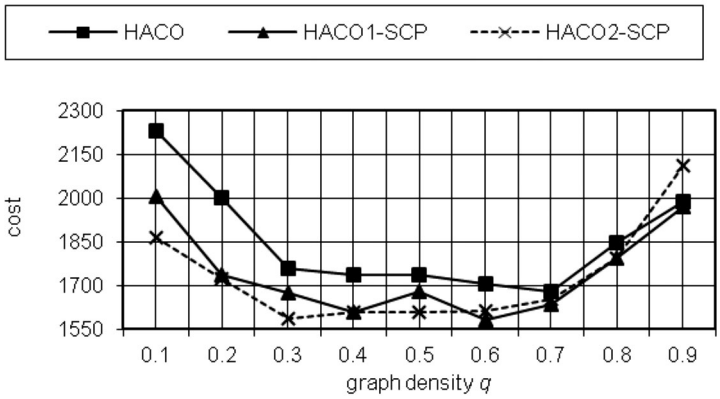


Fig. 2. An average cost in dependency on a graph density $q$

Rys. 2. Średni koszt pokrycia w zależności od gęstości grafu

Table 1

**An average cost in dependency on a graph density $q$**

| $q$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| HACO | 2233.0 | 2000.4 | 1757.8 | 1738.5 | 1736.6 | 1706.8 | 1678.9 | 1845.6 | 1988.3 |
| HACO1-SCP | 2006.9 | 1736.5 | 1676.0 | 1608.3 | 1677.4 | 1583.5 | 1633.1 | 1794.4 | 1969.8 |
| HACO2-SCP | 1865.5 | 1721.7 | 1587.7 | 1606.3 | 1606.4 | 1610.8 | 1651.3 | 1792.3 | 2111.1 |

Table 2

**An average cardinality number of $V_s$ in dependency on a graph density $q$**

| $q$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| HACO | 25.9 | 15.6 | 11.6 | 8.7 | 6.9 | 5.7 | 5,0 | 4,0 | 3.2 |
| HACO1-SCP | 25.3 | 15,0 | 10.3 | 7.9 | 6.1 | 5.0 | 4,0 | 3.1 | 2.2 |
| HACO2-SCP | 26.4 | 15,0 | 10.8 | 7.9 | 6.3 | 4.9 | 4,0 | 3,0 | 2,0 |

There is an improvement in quality of the solution when the HACO1-SCP or HACO2--SCP algorithm is used instead of the HACO algorithm since there are lower cardinality numbers of the set $V_s$ and there are lower covering costs for all investigated graph densities $q$ = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. The HACO2-SCP algorithm is better than the HACO1-SCP algorithm when average costs are taken into consideration for rare graphs $q \leq 0.5$ and there is not a difference between both algorithms for dense graphs $0.5 < q$. As concern average cardinality numbers HACO1-SCP and HACO2-SCP algorithm do not differ from one another.

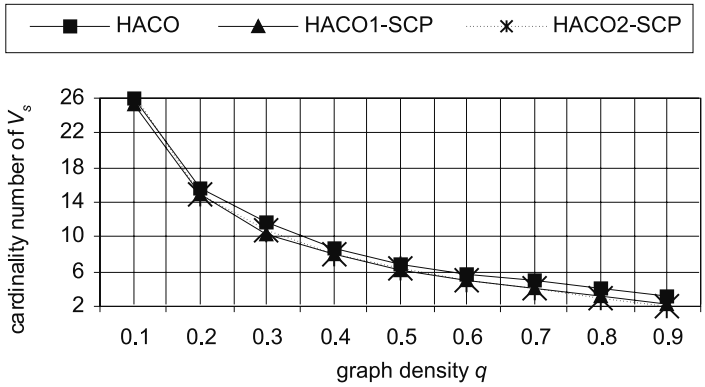


Fig. 3. An average cardinality number of $V_s$ in dependency on a graph density $q$

Rys. 3. Średnia liczba kardynalna zbioru $V_s$ w zależności od gęstości grafu $q$

The HACO1-SCP and the HACO2-SCP algorithm are also better than the HACO-SCP algorithm when average costs of set covering and average cardinality numbers of the set $V_s$ are taken into consideration in function of a number of bipartite graph vertices $n$ with different vertices degree. These two above parameters has been observed during conducted tests when a graph density was differentiated for different number of graph vertices and received values of two above parameters have been shown in the Table 3 and in the Table 4 or in the Fig. 4 and in the Fig. 5. In order to get a bipartite graph with a different density for each graph vertex each edge was generated with any probability so degree of each graph vertex has different values, this means each graph vertex has a different number of adjacent edges and thus a bipartite graph has vertices with different degree.

All experiments have shown that both elaborated algorithms give a better quality of solution than the HACO algorithm which has been presented in the paper [9].

Table 3

**An average cost in dependency on a number of vertices**

| $n$ | 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| HACO | 856.1 | 1696.1 | 2538.6 | 3314.4 | 4026.3 |
| HACO1-SCP | 839.8 | 1636.5 | 2433.8 | 3215.3 | 3933.9 |
| HACO2-SCP | 829.9 | 1571.7 | 2374.4 | 3106.9 | 3927.9 |

All algorithms, the HACO algorithm and these both elaborated algorithms HACO1--SCP and HACO2-SCP, which are presented in this paper, are implemented in Microsoft Visual C++ under Microsoft Windows XP on Intel Celeron CPU 1.7GHz, 256 Mb RAM and the running time of these algorithms are proportional to the time complexity expressed by a multiplication of a quadratic number of vertices $n^2$ existing in a bipartite graph, a number of cycles, a number of ants and a cardinal number of a set $V_s$.

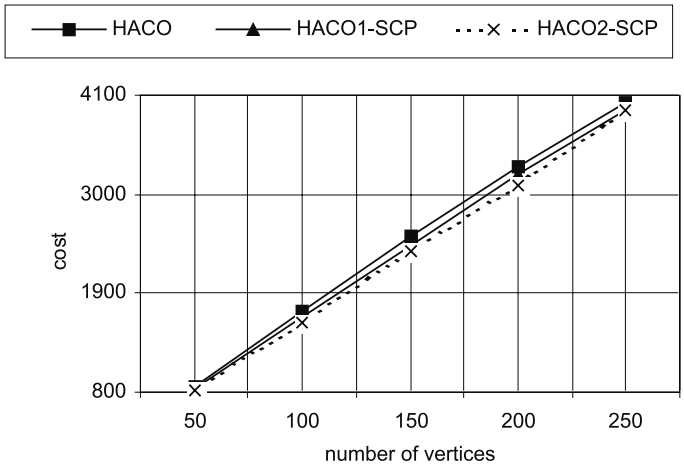


Fig. 4. An average cost in dependency on a number of vertices

Rys. 4. Średni koszt w zależności od liczby wierzchołków

Table 4

**An average cardinality number of $V_s$ in dependency on a number of vertices**

| $n$ | 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| HACO | 6.1 | 7.0 | 7.8 | 8.4 | 9.0 |
| HACO1-SCP | 5.0 | 6.1 | 7.0 | 7.6 | 8.0 |
| HACO2-SCP | 5.0 | 6.1 | 7.0 | 7.8 | 8.0 |

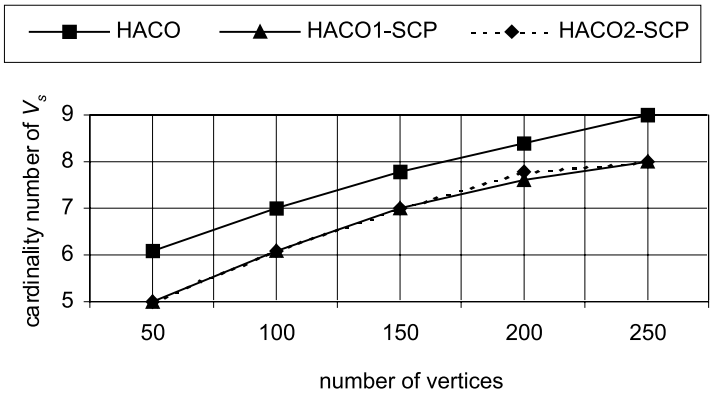


Fig. 5. An average number of cardinality number $V_s$ in dependency on a number of vertices

Rys. 5. Średnia liczba kardynalna zbioru $V_s$ w zależności od liczby wierzchołków

## 6. Conclusions

In this article the minimum cost set covering problem was solved by using the ACO algorithm with Constraint Programming and with new heuristic patterns. These new proposed heuristics, which have been used in the HACO1-SCP and the HACO2-SCP algorithm, lets to match in a better way an available vertex $i$ from the neighborhood $N_i$ of state to an already constructed partial solution $V_s$ in case of graphs with an almost equal degree of vertices and with edges, which have been generated in random with a determined probability $q$ and in case of graphs with different number of vertices and different degree of vertices and with edges, which are generated in random with any probability $q$ for each graph vertex. Both the HACO1-SCP and the HACO2-SCP algorithm look for a new vertex form the set $V_A$ to be added to a partial solution $V_s$ with the highest number of additional edges and with the lowest corresponding overall cost of partial solution. The HACO algorithm is taking into account only these available vertices from the set $V_A$ which are outside of a partial solution $V_S$ and which can be added to a partial solution with a minimum additional average cost, this means with an average minimum sum of edge weights and thus omits these available vertices from the set $V_A$ which can be added to a partial solution $V_s$ with a higher number of additional edges and a little higher cost than an average minimum sum of edge weights and which can minimize the overall cost of a constructed partial solution because of lower weights of its edges covering already covered yet vertices from the set $V_2$.

## References


[1] Valenzuela C., Crawford B., Soto R., Monfroy E., Paredes F., *A 2-level Metaheuristic for the Set Covering Problem*, International Journal of Computers, Communications & Control, vol. 7/2, 2012, 377-387.

[2] Ren Z.G., Feng Z.R., Ke L.J., Zhang Z.J., *New ideas for applying ant colony optimization to the set covering problem*, Journal Computers and Industrial Engineering, vol. 58/4, 2010, 774-784.

[3] Lessing L., Dumitrescu I., Stutze T., *A comparision between ACO algorithms for the set covering problem*, LNCS, vol. 3172, 2004, 1-12.

[4] Ren Z.G., Ke L.J., Chang H., *A fast and efficient Ant colony Optimization Approach for the Set covering problem*, Proc. of IEEE World Congress on Computational Intelligence, Hong Kong 2008, 1839-1844.

[5] Hadji R., Rahoual M., Talbi E., Bachelet V., *Ant colonies for the set covering problem*, [in:] Abstract Proc. of From Ant Colonies to Artificial Ants: Second International Workshop on Ant Colony Optimization, Brussels 2000, 63-66.

[6] Crawford B., Soto R., Monfroy E., Paredes F., Palma W., *A hybrid Ant algorithm for the set covering problem*, International Journal of the Physical Sciences, vol. 6/19, 2011, 4667-4673.

[7] Dechter R, Frost D., *Backjump-based backtracking for constraint satisfaction problems*, Artificial Intelligence, vol. 136/2, 2002, 147-188.

[8] Dorigo M., Di Caro G., *The Ant Colony Optimization meta-heuristics. New Ideas in Optimization*, McGraw Hill, New York 1999.

[9] Mihelic J., Robic B., *Facility Location and Covering Problems*, Proc. of the 7th International Multiconference Information Society, Volume D – Theoretical Computer Science, Ljubljana, Slovenia 2004.
[10] Housos E., Elmoth T., *Automatic optimization of subproblems in scheduling airlines crews*, Interfaces, vol. 27/5, 1997, 68-77.
[11] Vasko F.J., Wilson G.R., *Using a facility location algorithm to solve large set covering problems*, Operations Research Letters, vol. 3/2, 1984, 85-90.
[12] Vasko F.J., Wolf F.E., *Optimal selection of ingot sizes via set covering*, Operations Research, vol. 35/3, 1987, 346-353.
[13] Balas E., Carrera M.C., *A dynamic subgradient-based branch-and-bound procedure for set covering*, Operations Research, vol. 44/6, 1996, 875-890.
[14] Fisher M.L., Kedia P., *Optimal solution of set covering/partitioning problems using dual heuristics*, Management Science, vol. 36/6, 1990, 674-688.
[15] Chvatal V., *A greedy heuristic for the set-covering problem*, Mathematics of Operations Research, vol. 4/3, 1979, 233-235.
[16] Lan G., DePuy G.W., *On the effectiveness of incorporating randomness and memory into a multi-start meta-heuristic with application to the set covering problem*, Computer & Industrial Engineering, vol. 51/3, 2006, 362-374.
[17] Ceria S., Nobili P., Sassano A., *A Lagrangian-based heuristic for large-scale set covering problems*, Mathematical Programming, vol. 81, 1998, 215-228.
[18] Caprara A., Fischetti M., Toth P., *A heuristic method for the set covering problem*, Operations Research, vol. 47/5, 1999, 730-743.
[19] Beasley J.E., Chu P.C., *A genetic algorithm for the set covering problem*, European Journal of Operational Research, vol. 94/2, 1996, 392-404.
[20] Brusco M.J., Jacobs L.W., Thompson G.M., *A morphing procedure to supplement a simulated annealing heuristic for cost- and coverage-correlated set-covering problems*, Annals of Operations Research, vol. 86, 1999, 611-627.
[21] Caserta M., *Tabu search-based meta-heuristic algorithm for large-scale set covering problems*, [in:] Doerner K.F., Gendreau M., Greistorfer P., Gutjahr W., Hartl R., Reimann M. (Eds.), *Metaheuristics: Progress in complex systems optimization*, Springer, New York 2007, 43-63.
[22] Lessing L., Dumitrescu I., Stützle T., *A comparison between ACO algorithms for the set covering problem*, [in:] Dorigo M., Mauro Birattari M., Blum C., Gambardella L.M., Mondada F., Stützle T. (Eds.), *Ant Colony Optimization and Swarm Intelligence*, LNCS, vol. 3172, Springer-Verlag, Berlin, Heidelberg, 2004, 1-12.
[23] Crawford B., Castro C., *Integrating look ahead and post processing procedures with ACO for solving set partitioning and covering problems*, [in:] Rutkowski L., Tadeusiewicz R., Zadeh L.A., Żurada J.M. (Eds.), *Proc. of the 8th international conference on Artificial Intelligence and Soft Computing*, Springer-Verlag, Berlin, Heidelberg 2006, 1082-1090.
[24] Finger M., Stützle T., Ramalhinho H., *Exploiting fitness distance correlation of set covering problems*, [in:] Cagnoni S., Gottlieb J., Hart E., Middendorf M.R., Raidl G.R. (Eds.), *Proc. of the Applications of Evolutionary Computing on EvoWorkshops*, LNCS, vol. 2279, Springer, Berlin, Heidelberg 2002, 61-71.
[25] Dorigo M., Gambardella L.M., *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Transactions on Evolutionary Computation, vol. 1/1, 1997, 53-66.

[26] Gagne C., Gravel M., Price W., *A look-ahead addition to the ant colony optimization metaheuristic and its application to an industrial scheduling problem*, Proc. of the fourth Metaheuristics International Conference, vol. 1, Proto 2001, 79-84.

[27] Michel R., Middendorf M., *An island model based ant system with lookahead for the shortest super sequence problem*, [in:] Eiben A.E., Bäck T., Schoenauer M., Schwefel H.P. (Eds.), *Parallel Problem Solving from Nature*, LNCS, Springer Verlag, vol. 1498, 1998, 692-701.