Computer Science Journal of Moldova, vol.**22**, no.1(64), 2014

Search tree-based approach for the p-median problem using the ant colony optimization algorithm

Gabriel Bodnariuc, Sergiu Cataranciuc

Abstract

In this paper we present an approximation algorithm for the p-median problem that uses the principles of ant colony optimization technique. We introduce a search tree that keeps the partial solutions during the solution process of the p-median problem. An adaptation is proposed that allows ant colony optimization algorithm to perform on this tree and obtain good results in short time.

Keywords: ant colony optimization, p-median, location theory, combinatorial optimization, search tree.

1 Introduction

Let $G = (X, U)$ be an undirected graph, with the vertex set $X = \{x_1, x_2, ..., x_n\}$ and the edge set $U = \{u_1, u_2, ..., u_m\}$. We define two functions:

a) $\upsilon : X \to N$;

b) $\omega : U \to N$,

where $N = \{0, 1, 2, ...\}$.

Values $\upsilon(x_i)$ and $\omega(u_j)$ are called weights of the vertex $x_i \in X$ and of the edge $u_j \in U$, respectively. We denote by $d(x_i, x_j)$ the distance between vertices $x_i, x_j \in X$ [16] and consider the function $f : X \to N$ such that:

$$f(x_i) = \sum_{x_j \in X} \upsilon(x_j)\, d(x_i, x_j) \tag{1}$$

for $\forall x_i \in X$.

Definition 1 ([6]). *The vertex $x^* \in X$ is called median of the graph if $f(x^*) = \min\limits_{x \in X} f(x)$.*

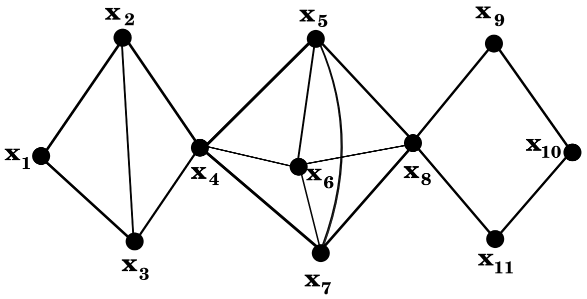

Figure 1. Vertices x_5, x_6 and x_7 are medians of the graph

According to this definition, the median of graph cannot be found univocally. For example, vertices x_5, x_6 and x_7 are the medians of the graph represented in Figure 1 (edges and vertices weights are equal to 1).

Let $A \subset X$ and an arbitrary vertex $x \in X$. We denote by $d(x, A)$ the distance between vertex x and the set A. According to [16] $d(x, A) = \min\limits_{y \in A} \{d(x, y)\}$.

We denote by $\mathcal{X}_p$ the family of all subsets of size p of the set X, $1 \leq p \leq n = |X|$, and define function $f : \mathcal{X}_p \to N$ such that:

$$f(A) = \sum_{x_j \in X} \upsilon(x_j)\, d(x_j, A) \tag{2}$$

for all $A \in \mathcal{X}_p$. Function (2) is called median function.

Definition 2 ([16])**.** *Set $A^* \subset X$, $|A^*| = p$, is called p-median of the graph $G = (X, U)$, if the following relation holds:*

$$f\left(A^*\right) = \min_{A \in \mathcal{X}_p} f\left(A\right) = \min_{A \in \mathcal{X}_p} \sum_{x_j \in X} p\left(x_j\right) d\left(x_j, A\right). \tag{3}$$

We will study the p-median problem, where vertex and edge weights have arbitrary values.

2 Methods for solving the median problem on graphs

Finding the median of an undirected graph $G = (X, U)$ is a difficult discrete optimization problem. Being a NP-complete [9], [13], this problem has stimulated the interest of many researchers for building approximation algorithms for finding graph median. In this case it is necessary to know how good is the approximation of the obtained results using these methods.

There are some well known exact algorithms for finding p-median [3], [6], [7], [18], but their efficiency is limited to a certain size of the graphs. In case of the trees, the 1-median can be found in time $O(n)$ [11] and p-median in time $O(pn^2)$ using a dynamic programming algorithm [17]. Also in polynomial time the median for d-convex simple graphs [5] can be found.

There are many other techniques for solving this problem: genetic algorithm [1], branch and cut [4], scatter search [10], variable neighborhood search [12].

In this paper we show a modification of the algorithm presented in [14] that uses the principles of the ant colony algorithms. This type of algorithm was proposed by M. Dorigo, V. Maniezzo and A. Colorni, the algorithm is described in [8]. The first problem on which the ant colony optimization algorithm was applied was the travelling salesman problem.

The ant colony optimization algorithm is based on observation of ants, which are colony organised insects. Their activity is oriented for the benefit of the whole colony. One important aspect is the way they build short paths between colony location and food sources. Ants deposit a substance called *pheromone* while walking. Pheromone indicates the path used by other ants. Each ant usually chooses a path with high concentration of pheromone. This represents an indirect way of communication called *stigmergy.* The environment has an important role to diminish the quantity of the pheromone. This leads to changing of attractiveness of choosing different paths.

3 Tree representation of the p-median search

The problem solution is searched by starting vertex elimination from the vertex set X until p elements remain. At each step a vertex is eliminated from X according to some rules. All combinations of vertex elimination can be represented by a rooted tree, denoted by T.

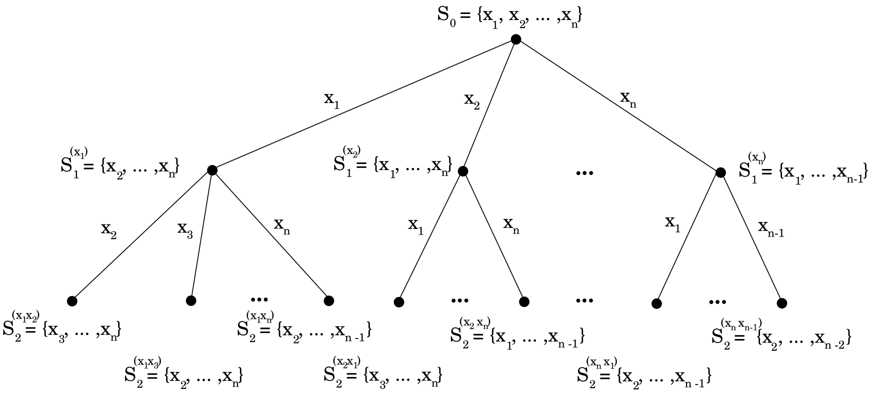
$S_0 = \{x_1, x_2, \dots, x_n\}$
x_1
x_2
x_n
$S_1^{(x_1)} = \{x_2, \dots, x_n\}$
$S_1^{(x_2)} = \{x_1, \dots, x_n\}$
...
$S_1^{(x_n)} = \{x_1, \dots, x_{n-1}\}$
x_2
x_3
x_n
x_1
x_n
x_1
x_{n-1}
...
...
...
...
$S_2^{(x_1 x_2)} = \{x_3, \dots, x_n\}$
$S_2^{(x_1 x_n)} = \{x_2, \dots, x_{n-1}\}$
$S_2^{(x_2 x_n)} = \{x_1, \dots, x_{n-1}\}$
$S_2^{(x_n x_{n-1})} = \{x_2, \dots, x_{n-2}\}$
$S_2^{(x_1 x_3)} = \{x_2, \dots, x_n\}$
$S_2^{(x_2 x_1)} = \{x_3, \dots, x_n\}$
$S_2^{(x_n x_1)} = \{x_2, \dots, x_{n-1}\}$

Figure 2. Tree representation of the 2-median search

Each tree node corresponds to a subset of X when from graph G a number of vertices was eliminated and the search of p-median is done on the remaining vertices. The tree root S_0 corresponds to the vertex set X. Each arc from S_0 corresponds to the elimination of a vertex from X and this vertex will not be a part of the solution. In this way, we pass from S_0 to S_1^j on the level 1. There are $n = C_n^1$ nodes on the first level of the tree T. The process is repeated for each node on the level $i = 1, 2, ..., (n-p+1)$. An example for the case $p = n-2$ is shown in Figure 2. The bottom level nodes represent all subsets of size p of the vertex set X. The tree T contains $1 + A_n^1 + A_n^2 + ... + A_n^{n-p}$ nodes. Each node on the level k has $n-k$ descendants, where $0 \leq k \leq n-p+1$.

In the tree T a path between the root node, which corresponds to the set S_0 and any node of the level $n-p$, is considered a branch. Value of the median is among the values of median functions for the sets that correspond to the nodes of the level $n-p$.

Ant colony algorithm is used for finding a preferential branch in the tree T that will lead to choosing a set S^*_{n-p} as an approximate solution of the problem.

There are r ants in the colony. Each ant searches the p-median traversing a branch of the tree T. An iteration corresponds to the situation when r ants participate to find a branch in the tree. Obviously, some of these branches intersect other branches. At the end of an iteration, there will be some nodes with better values of the median function. These results will be used for building new preferential path in the tree T that will help to find better solutions.

4 Reduced tree representation of the p-median search

The size of the tree described above can be reduced if we take into account some specific features of the problem. This feature will optimize the solution search process performed by ant colony algorithm.

Let S_k^j, $1 \leq k \leq n-p$, be a node of the search tree, which is obtained from $S_0 = X$ after a successive elimination of vertices in the following order $a_{i_1}, a_{i_2}, ..., a_{i_k}$. In this case, we denote the node S_k^j by $S_k^{\left(a_{i_1}, a_{i_2}, ..., a_{i_k}\right)}$. The weight of this node is $\omega\left(S_k^{\left(a_{i_1}, a_{i_2}, ..., a_{i_k}\right)}\right)$ which is equal to the value of function (2) for vertex set of graph G that corresponds to the node $S_k^{\left(a_{i_1}, a_{i_2}, ..., a_{i_k}\right)}$ from the tree T. Formally, this can be written:

$$\omega\left(S_k^{(a_{i_1},a_{i_2},\ldots,a_{i_k})}\right)=F\left(S_k^{(a_{i_1},a_{i_2},\ldots,a_{i_k})}\right)$$

(here $F\left(S_k^{(a_{i_1},a_{i_2},\ldots,a_{i_k})}\right)$ represents the function value (2) for the set $A=X\setminus\{a_{i_1},a_{i_2},\ldots,a_{i_k}\}$).

Elements elimination order from set A does not affect the value of function (2) for a subset of vertices $A\subset X$, $A=X\setminus\{a_{i_1},a_{i_2},\ldots,a_{i_k}\}$, so:

Theorem 1. *Let the set $S_k^{(a_{i_1},\ldots,a_{i_k})}$ is obtained from the set S_0 after successive elimination of elementents $a_1,a_2,\ldots,a_k$, and the set $S_k^{(a_{i_1^*},\ldots,a_{i_k^*})}$ is obtained after a different elimination order of the same elements, then these two sets satisfy the following relation:*

$$F(S_k^{(a_{i_1},\ldots,a_{i_k})})=F(S_k^{(a_{i_1^*},\ldots,a_{i_k^*})}).$$

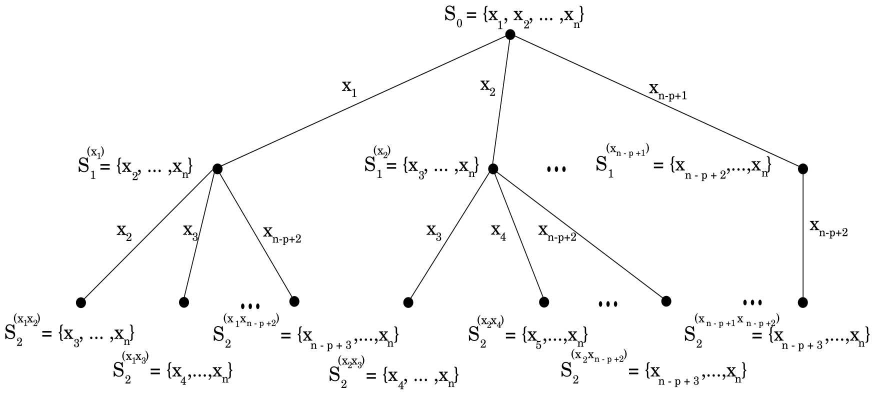
$S_0=\{x_1,x_2,\ldots,x_n\}$
x_1
x_2
x_{n-p+1}
$S_1^{(x_1)}=\{x_2,\ldots,x_n\}$
$S_1^{(x_2)}=\{x_3,\ldots,x_n\}$
...
$S_1^{(x_{n-p+1})}=\{x_{n-p+2},\ldots,x_n\}$
x_2
x_3
x_{n-p+2}
x_3
x_4
x_{n-p+2}
x_{n-p+2}
...
...
...
$S_2^{(x_1x_2)}=\{x_3,\ldots,x_n\}$
$S_2^{(x_1x_{n-p+2})}=\{x_{n-p+3},\ldots,x_n\}$
$S_2^{(x_2x_4)}=\{x_5,\ldots,x_n\}$
$S_2^{(x_{n-p+1}x_{n-p+2})}=\{x_{n-p+3},\ldots,x_n\}$
$S_2^{(x_1x_3)}=\{x_4,\ldots,x_n\}$
$S_2^{(x_2x_3)}=\{x_4,\ldots,x_n\}$
$S_2^{(x_2x_{n-p+2})}=\{x_{n-p+3},\ldots,x_n\}$

Figure 3. Reduced tree representation of the 2-median search

This gives us the possibility to build a new tree T^* which has a smaller size than T. It is obtained from T after removing some nodes that give identical results. A reduced search tree for case $p=n-2$ is represented in Figure 3.

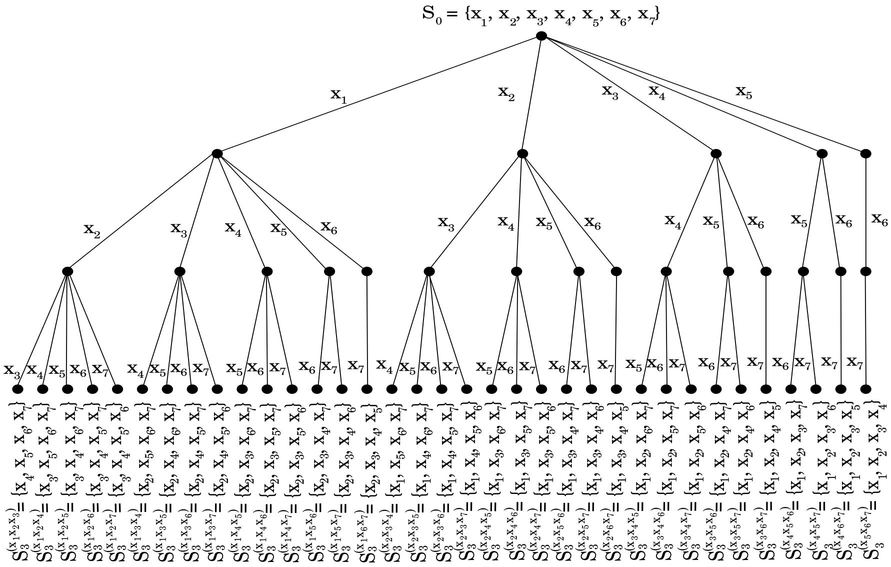
S0 = {x1, x2, x3, x4, x5, x6, x7}
x1
x2
x3
x4
x5
x2
x3
x4
x5
x6
x3
x4
x5
x6
x4
x5
x6
x5
x6
x6
x3 x4 x5 x6 x7
x4 x5 x6 x7
x5 x6 x7
x6 x7
x7
x4 x5 x6 x7
x5 x6 x7
x6 x7
x7
x5 x6 x7
x6 x7
x7
x6 x7
x7
x7
S3(x1x2x3)={x4, x5, x6, x7}
S3(x1x2x4)={x3, x5, x6, x7}
S3(x1x2x5)={x3, x4, x6, x7}
S3(x1x2x6)={x3, x4, x5, x7}
S3(x1x2x7)={x3, x4, x5, x6}
S3(x1x3x4)={x2, x5, x6, x7}
S3(x1x3x5)={x2, x4, x6, x7}
S3(x1x3x6)={x2, x4, x5, x7}
S3(x1x3x7)={x2, x4, x5, x6}
S3(x1x4x5)={x2, x3, x6, x7}
S3(x1x4x6)={x2, x3, x5, x7}
S3(x1x4x7)={x2, x3, x5, x6}
S3(x1x5x6)={x2, x3, x4, x7}
S3(x1x5x7)={x2, x3, x4, x6}
S3(x1x6x7)={x2, x3, x4, x5}
S3(x2x3x4)={x1, x5, x6, x7}
S3(x2x3x5)={x1, x4, x6, x7}
S3(x2x3x6)={x1, x4, x5, x7}
S3(x2x3x7)={x1, x4, x5, x6}
S3(x2x4x5)={x1, x3, x6, x7}
S3(x2x4x6)={x1, x3, x5, x7}
S3(x2x4x7)={x1, x3, x5, x6}
S3(x2x5x6)={x1, x3, x4, x7}
S3(x2x5x7)={x1, x3, x4, x6}
S3(x2x6x7)={x1, x3, x4, x5}
S3(x3x4x5)={x1, x2, x6, x7}
S3(x3x4x6)={x1, x2, x5, x7}
S3(x3x4x7)={x1, x2, x5, x6}
S3(x3x5x6)={x1, x2, x4, x7}
S3(x3x5x7)={x1, x2, x4, x6}
S3(x3x6x7)={x1, x2, x4, x5}
S3(x4x5x6)={x1, x2, x3, x7}
S3(x4x5x7)={x1, x2, x3, x6}
S3(x4x6x7)={x1, x2, x3, x5}
S3(x5x6x7)={x1, x2, x3, x4}

Figure 4. Reduced tree representation for the case $n = 7$ and $p = 4$

$p+1$ arcs start from the root node $x^{T^*}_{S_0}$. These arcs correspond to exclusion of elements a_1, a_2, ..., a_{p+1} from S_0 and make connection with nodes of the level 1: $x^{T^*}_{S_1^{(a_1)}}$, $x^{T^*}_{S_1^{(a_2)}}$,..., $x^{T^*}_{S_1^{\left(a_{p+1}\right)}}$ that correspond to the sets $S_1^{(a_1)} = \{a_2, a_3, ..., a_n\}$, $S_1^{(a_2)} = \{a_3, ..., a_n\}$,..., $S_1^{(a_{p+1})} = \{a_{p+2}, ..., a_n\}$, respectively.

At the tree level 1, $p+1$ arcs start from the node $x^{T^*}_{S_1^{(a_1)}}$, and they indicate the possibility of removing of the following elements from $S_1^{(a_1)}$: a_2, a_3, ..., a_{p+2}.

It is the node $x^{T^*}_{S_1^{(a_2)}}$ from which p arcs start, and they indicate the possibility of removing of the following elements from $S_1^{(a_2)}$: a_3, ..., a_{p+2}.

It is the node $x^{T^*}_{S_1^{(a_3)}}$ from which $p-1$ arcs start, and they indicate the possibility of removing of the following elements from $S_1^{(a_3)}$:

a_4, ..., a_{p+2}.

...

It is the node $x^{T^*}_{S_1^{(a_{p+1})}}$ from which one arc starts, and it indicates the possibility of removing of the element a_{p+2} from $S_1^{(a_{p+1})}$.

The process continues until the level $n - p$ of the tree T. There are C_n^p nodes on the last level.

There is an example in Figure 4, where $n = 7$ and $p = 4$, and in Figure 5 it is shown what happens when element x_1 is removed.

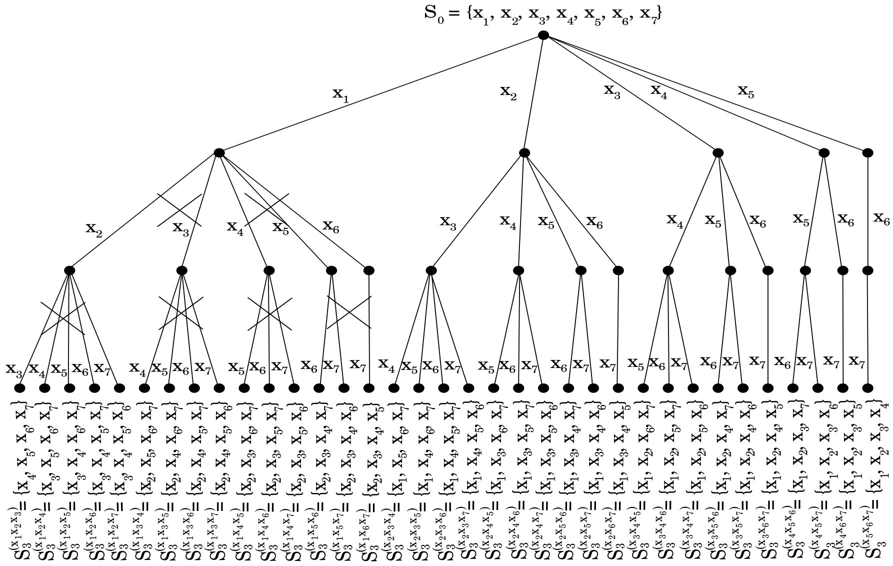
S₀ = {x₁, x₂, x₃, x₄, x₅, x₆, x₇}
x₁
x₂
x₃
x₄
x₅
x₂
x₃
x₄
x₅
x₆
x₃
x₄
x₅
x₆
x₄
x₅
x₆
x₅
x₆
x₆
x₃ x₄ x₅ x₆ x₇
x₄ x₅ x₆ x₇
x₅ x₆ x₇
x₆ x₇
x₇
x₄ x₅ x₆ x₇
x₅ x₆ x₇
x₆ x₇
x₇
x₅ x₆ x₇
x₆ x₇
x₇
x₆ x₇
x₇
x₇
S₃^(x₁x₂x₃)={x₄, x₅, x₆, x₇}
S₃^(x₁x₂x₄)={x₃, x₅, x₆, x₇}
S₃^(x₁x₂x₅)={x₃, x₄, x₆, x₇}
S₃^(x₁x₂x₆)={x₃, x₄, x₅, x₇}
S₃^(x₁x₂x₇)={x₃, x₄, x₅, x₆}
S₃^(x₁x₃x₄)={x₂, x₅, x₆, x₇}
S₃^(x₁x₃x₅)={x₂, x₄, x₆, x₇}
S₃^(x₁x₃x₆)={x₂, x₄, x₅, x₇}
S₃^(x₁x₃x₇)={x₂, x₄, x₅, x₆}
S₃^(x₁x₄x₅)={x₂, x₃, x₆, x₇}
S₃^(x₁x₄x₆)={x₂, x₃, x₅, x₇}
S₃^(x₁x₄x₇)={x₂, x₃, x₅, x₆}
S₃^(x₁x₅x₆)={x₂, x₃, x₄, x₇}
S₃^(x₁x₅x₇)={x₂, x₃, x₄, x₆}
S₃^(x₁x₆x₇)={x₂, x₃, x₄, x₅}
S₃^(x₂x₃x₄)={x₁, x₅, x₆, x₇}
S₃^(x₂x₃x₅)={x₁, x₄, x₆, x₇}
S₃^(x₂x₃x₆)={x₁, x₄, x₅, x₇}
S₃^(x₂x₃x₇)={x₁, x₄, x₅, x₆}
S₃^(x₂x₄x₅)={x₁, x₃, x₆, x₇}
S₃^(x₂x₄x₆)={x₁, x₃, x₅, x₇}
S₃^(x₂x₄x₇)={x₁, x₃, x₅, x₆}
S₃^(x₂x₅x₆)={x₁, x₃, x₄, x₇}
S₃^(x₂x₅x₇)={x₁, x₃, x₄, x₆}
S₃^(x₂x₆x₇)={x₁, x₃, x₄, x₅}
S₃^(x₃x₄x₅)={x₁, x₂, x₆, x₇}
S₃^(x₃x₄x₆)={x₁, x₂, x₅, x₇}
S₃^(x₃x₄x₇)={x₁, x₂, x₅, x₆}
S₃^(x₃x₅x₆)={x₁, x₂, x₄, x₇}
S₃^(x₃x₅x₇)={x₁, x₂, x₄, x₆}
S₃^(x₃x₆x₇)={x₁, x₂, x₄, x₅}
S₃^(x₄x₅x₆)={x₁, x₂, x₃, x₇}
S₃^(x₄x₅x₇)={x₁, x₂, x₃, x₆}
S₃^(x₄x₆x₇)={x₁, x₂, x₃, x₅}
S₃^(x₅x₆x₇)={x₁, x₂, x₃, x₄}

Figure 5. Prunning vertex x_1 from the search process

Theorem 2. *Let the graph $G = (X, U)$, where $|X| = n$, has the subsets $Y_k \in X$, ($|Y_k| = k, k > 1$) and $Y_{k-1} = Y_k \setminus \{x_i^*\}$, then:*

$$F(Y_k) \leq F(Y_{k-1})$$

Proof. There are two possible cases:

a) eliminated vertex x_i^* from the subset Y_k has in its neighbourhood only elements from Y_k. So, the shortest path that connects each element of the set $X \setminus Y_k$ to one of the elements of the subset Y_k remains unchanged and the following sum does not change: $\sum\limits_{x_i \in X \setminus Y_k} d(x_i, Y_k)$, and $F(Y_{k-1}) = F(Y_k) + \min\limits_{x_i \in \Gamma(x_i^*)} d(x_i, x_i^*)$;

b) one of the following two relations is true: (i) $\Gamma(x_i^*) \subset X \setminus Y_k$ or (ii) $\Gamma(x_i^*) \cap \{X \setminus Y_k\} \neq \emptyset$ and $\Gamma(x_i^*) \cap \{Y_k\} \neq \emptyset$. Then paths of minimal length between some vertices from $X \setminus Y_k$ and x_i^* disappear and there is a necessity to establish new paths of minimal length, that make connections with vertices from Y_{k-1}. New paths are longer than the initial paths. A new connection between vertex x_i^* and $y_i \in Y_{k-1}$ and $y_i \in Y_{k-1}$ and $y_i \in Y_{k-1}$ is built.

5 Implementation of ant colony algorithm for the p-median problem

The role of ants is to use the reduced tree described above for finding a good approximate solution. The size of tree used in the solving process should be as small as possible.

This reduced tree is denoted by T^* and it is built iteratively. At first, the tree T^* has only the root node that corresponds to the set S_0. Each ant chooses, with some probability, an element a_i, $1 \leq i \leq n-p+1$ for removing from this set. The chosen element will not be examined further by the ant as part of its solution. Eliminations are done until there are $n-p$ elements from the original set. The eliminated elements form a branch in the tree T^*. To each element $x_i \in S_0$ it is attributed a value τ_i that represents quantity of pheromone. Initially, the quantity of the pheromone is equal for every element. The pheromone influences the way how elements are chosen for removing from the set S_k.

The probability to choose element i at stage k that represents elimination of element $x_i \in S_k$, is [8]:

$$p_i\left(k\right) = \frac{[\tau_i]^\alpha [\eta_i\left(k+1\right)]^\beta}{\sum\limits_{l=1}^{|S_k|} [\tau_l]^\alpha [\eta_l\left(k+1\right)]^\beta},$$

where $\eta_i\,(k+1) = 1/F\,(S_k \setminus \{a_i\})$.

The quantity of pheromone deposited by each ant at the end of one iteration is:

$$\Delta\tau_i^j = \begin{cases} 1/F^j\,(S_{n-p}) & \text{, if } x_i \in S_{n-p} \\ 0 & \text{, otherwise} \end{cases}$$

where $F^j\,(S_{n-p})$ is the solution obtained by the ant j.

Evaporation and addition of pheromone is realized in the following way:

$$\tau_i \leftarrow (1-\rho)\,\tau_i + \Delta\tau_i,$$

where $\Delta\tau_i = \sum_{j=1}^{r} \Delta\tau_i^j$, r is the number of ants and ρ is the evaporation coefficient.

According to Theorem 2, the elimination of one element from S_k leads to rising of value $F(S_{k-1})$. This helps us to build a *branch-and-bound* algorithm.

5.1 Improving the obtained solution

The solution can be improved if the vertex set X of the graph $G = (X, U)$ is partitioned into p subsets: X_i, $i = \overline{1,p}$. Let $S = \{x_{i_1}, x_{i_2}, ..., x_{i_p}\}$ be an approximate solution.

Each subset X_j, $i = \overline{1,p}$, consists of one vertex x_{i_j} contained in the solution S and the closest to it vertexes from the set $G \setminus S$. The 1-median of each subgraph $G_i = (X_i, U_i)$ is found. The set S_1 of obtained 1-medians could be considered a new approximate solution if $F(S) > F(S_1)$.

5.2 Ant colony optimization algorithm on the reduced search algorithm (ACORST)

All operations are done on the tree T^*.

1. Let $nrIterations$ be the maximal number of iterations, $nrAnts$ – the number of ants, $Rec = \infty$ – the initial value of the searched‘ solution.

2. $iteration := 0$;

3. $idAnt := 0$;

4. $k := 0$;

5. Ant $idAnt$ starts from the root node x_0;

6. If for the current node x_k there are no child nodes of the level $k+1$, then the neigbourhood set $\Gamma^+_{x_k}$ is built;

7. For each node $x_{k+1} \in \Gamma^+_{x_k}$ the value $F(S_{k+1})$ is calculated;

8. If there are nodes x_{k+1} for which $Rec < F(S_{k+1})$, then the node x_{k+1} and its subtree are removed and will not be examined further;

9. A node $x_{k+1} \in \Gamma^+_{x_k}$ is chosen with probability p described above;

10. If $\Gamma^+_{x_k} = \emptyset$, then we pass to the step 13;

11. $k := k+1$;

12. If $k < n-p-1$, then we pass to the step 6;

13. If $k = n-p-1$, then improve the solution using the algorithm described in the section 5.1 and then $rec = \min\limits_{x_{k+1} \in \Gamma^+_{x_k}} F(S_{k+1})$. If $rec < Rec$, then $Rec := rec$;

14. If there are nodes x_{k+1}, for which $Rec < F(S_{k+1})$, then the node x_{k+1} and its subtree is removed and will not be examined further;

15. $k := k-1$;

16. If $k > 0$, then we pass to the step 14;

17. $idAnt := idAnt + 1$;

18. If $idAnt < nrAnts$, then we pass to the step 4;

19. $iteration := iteration + 1$;

20. If the tree T^* has only the root node x_0, then STOP;

21. If $iteration < nrIterations$, then we pass to the step 3, else STOP.

6 Experimental results

The OR Library was chosen for tests [2]. Here the results of ACORST algorithm and the results of the implementation of ACO algorithm [14] are produced. The tests were performed on a Pentium Dual Core 2.2 GHz PC with 3 GB memory. The algorithms were implemented in C++ and the codes were compiled with gcc 4.5.0 compiler with optimization flag -O2. For both algorithms the ant colony consists of 30 ants and the number of iterations is limited to 40. In the Table 1 the best results obtained after 50 runs of the algorithm for each instance and the mean running time in seconds are produced.

Table 1. Results of ACORST and ACO algorithms

Test	n	p	Optimal	ACORST	Time	ACO	Time
pmed1	100	5	5819	5819	0.427	5819	3.214
pmed2	100	10	4093	4247	0.597	4093	3.097
pmed3	100	10	4250	4279	0.646	4273	3.11
pmed4	100	20	3034	3312	1.105	3050	2.461
pmed5	100	33	1355	1463	1.354	1357	2.211
pmed6	200	5	7824	7877	2.047	7824	24.554
pmed7	200	10	5631	5854	2.434	5645	25.641
pmed8	200	20	4445	4824	4.087	4479	23.601
pmed9	200	40	2734	3036	7.168	2797	21.064
pmed10	200	67	1255	1331	14.466	1288	16.483
pmed11	300	5	7696	7721	6.117	7696	93.177
pmed12	300	10	6634	6948	8.075	6657	86.424
pmed13	300	30	4374	4900	16.89	4449	83.907
pmed14	300	60	2968	3341	26.247	3057	74.346
pmed15	300	100	1729	1987	35.965	1773	52.85
pmed16	400	5	8162	8314	12.494	8162	245.656
pmed17	400	10	6999	7262	12.956	7010	226.541
pmed18	400	40	4809	5157	30.153	4906	213.587
pmed19	400	67	2845	3713	40.772	3319	193.144
pmed20	400	133	1789	1866	76.547	1820	132.204

The ACO algorithm has slightly better solutions than ACORST, but the running time is worse. The running time of our algorithm is proportional to p for fixed n.

7 Conclusion

We proposed an algorithm for the p-median problem using ant colony optimization technique. The algorithm is based on using a tree for keeping track of vertex removals and for pruning bad solutions. The algorithm gives good results for $p < \frac{n}{2}$ and there is a comparison table with results obtained by ACO algorithm.

References

[1] O. Alp, E. Erkut, D. Drezner. *An efficient genetic algorithm for the p-median problem*, Annals of Operations Research, vol. 122, issue 1-4, pp. 21–42, 2003.

[2] J.E. Beasley. *OR-Library: Distributing test problems by electronic mail*, Journal of the Operational Research Society, vol. 41, no. 11, pp. 1069–1072, 1990.

[3] C. Beltran, C. Tadonki, J.-Ph. Vial. *Solving the p-median problem with a semi-Lagrangian relaxation*, Computational Optimization and Applications, vol. 35, issue 2, pp.239–260, 2006.

[4] M. Boccia, A. Sforza, C. Sterle, I. Vasilyev. *A cut and branch approach for the capacitated p-median problem based on Fenchel cutting planes*, J. Math. Model. Algor., vol. 7, pp 43–58, 2008.

[5] S. Cataranciuc, N. Sur. *d-convex simple and quasi-simple graphs*, CECMI Textbook and Monograph Series, vol. 7, State University of Moldova, Chişinău, 200 p., 2009. (in Romanian)

[6] N. Christofides. *Graph Theory: an algorithmic approach*, Academic Press, 400 p., 1975.

[7] N. Christofides, J.E. Beasley. *Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem*, European Journal of Operational Research, Vol. 12, no. 1, pp. 19–28, 1983.

[8] M. Dorigo, G. Di Caro, M. Gambardella. *Ant algorithm for Discrete Optimization*, Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.

[9] M.R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 338 p., 1979.

[10] F. Garcia-Lopez, B. Melian-Batista, J.A. Moreno-Perez, J. Moreno-Vega. *Parallelization of the scatter search for the p-median problem*, Parallel Computing, vol. 29, pp 575–589, 2003.

[11] A.J. Goldman. *Optimal center location in simple networks*, Transportation Science, vol. 5, issue 2, pp. 212–221, 1971.

[12] K. Fleszar, K.S. Hindi. *An effective VNS for the capacitated p-median problem*, European Journal of Operational Research, no. 191, pp. 612–622, 2008.

[13] O. Kariv, S.L. Hakimi. *An Algorithmic Approach To Network Location Problems. II: The p-medians*, SIAM J. appl. math., vol. 37, no. 3, pp. 513–538, 1979.

[14] T.V. Levanova, M.A. Loresh. Algorithms of ant system and simulated annealing for the p-median problem, Avtomatika i Telemekhanika, vol. 65, no. 3, pp. 80–89, 2004. (in Russian)

[15] G.J. Lim, J. Reese, Holder G. Allen. *Fast and Robust Techniques for the Euclidean p-Median Problem with Uniform Weights*, Computers and Industrial Engineering, vol. 57, issue 3, pp. 896–905, 2009.

[16] P. Soltan. *Extremal problems on convex sets*, Ştiinţa, Chişinau, 115 p., 1976. (in Russian)

[17] A. Tamir. *An $O(pn^2)$ algorithm for the* p-*median and related problems on tree*, Operations Research Letters, vol. 19, issue 2, pp. 59–64, 1996.

[18] J.R. Weaver, R.L. Church. *A median location model with nonclosest facility service* Transportation Science, vol. 19, issue 1, pp. 58–74, 1985.

Gabriel Bodnariuc, Sergiu Cataranciuc Received June 20, 2013

Gabriel Bodnariuc
State University of Moldova, Republic of Moldova
60 A. Mateevici, MD-2009.
E–mail: gabriel.bodnariuc@gmail.com

Sergiu Cataranciuc
State University of Moldova
60 A. Mateevici, MD-2009, Republic of Moldova
E–mail: s.cataranciuc@gmail.com