FLYING ANT COLONY OPTIMIZATION ALGORITHM FOR COMBINATORIAL OPTIMIZATION

Summary. In this paper is introduce "flying" ants in Ant Colony Optimization (ACO). In traditional ACO algorithms the ants construct their solution regarding one step forward. In proposed ACO algorithm, the ants make their decision, regarding more than one step forward, but they include only one new element in their solutions.

Keywords: Evolutionary computation, Ant colony optimization, Combinatorial optimization

ALGORYTM OPTYMALIZACJI KOŁONII LATAJĄCYCH MRÓWEK W CELU OPTYMALIZACJI KOMBINATORYCZNEJ

Streszczenie. Artykuł przedstawia "latające" mrówki w problemie optymalizacji algorytmów mrówkowych. W tradycyjnych podejściach dla algorytmów mrówkowych agenci (mrówki) budują swoje rozwiązanie w kolejnych krokach. W zaproponowanym podejściu optymalizacji algorytmu mrówkowego agenci podejmują decyzję na podstawie więcej niż jednego kroku, jednakże tylko jeden element wprowadzany jest do rozwiązania.

Słowa kluczowe: obliczenia ewolucyjne, optymalizacja algorytmów mrówkowych, optymalizacja kombinatoryczna

1 Work presented here is partially supported by the Bulgarian National Scientific Fund under Grants DFNI I02/20 "Efficient Parallel Algorithms for Large Scale Computational Problems" and DFNI DN 02/10 "New Instruments for Data Mining and their Modeling".
1. Introduction

Ant Colony Optimization (ACO) is an evolutionary method suitable for solving combinatorial optimization problems. It belongs to the so called metaheuristic methods for optimization. In metaheuristic algorithms, values of several algorithm components and parameters have to be set, due to their significant impact on the algorithm’s efficacy and performance [1-6]. Therefore, it is important to study how the algorithm parameters affect the performance.

The idea for application of ant behavior for solving combinatorial optimization problems is done by Marco Dorigo twenty five years ago [7-9]. ACO is applicable for a broad range of optimization problems. It can be adapted to dynamic changes of the problem [1-3]. At the beginning it is applied on traveling salesman problem [10]. Later it is successfully applied on a lot of complex optimization problems. During the years, various variants of ACO methodology was proposed: ant system [9]; elitist ants [9]; ant colony system [10]; max-min ant system [3]; rank-based ant system [9]; ant algorithm with additional reinforcement [11]. Main difference between them is pheromone updating. For some of them is proven that they converge to the global optimum [9]. Fidanova et all [12-14] proposed semi-random start of the ants, comparing several start strategies. ACO can compete with other global optimization techniques like genetic algorithms, simulated annealing, tabu search and other metaheuristic methods.

Generally, more information gives a possibility for better decision making and better opportunity for finding good solutions of issue problem. The authors of this paper try to improve the traditional ant algorithm, giving to the ants a possibility to receive more information about the search space imitating flying ants. The main difference with traditional ACO algorithm is that an ant can make decision about the direction to move, taking in to account the information from more than one step ahead. The idea is tested on two very different optimization problems, Global Positioning System (GPS) surveying problem and Multiple Knapsack Problem (MKP). The GPS surveying problem is a minimization problem similar to the traveling salesman problem. It belongs to the set of ordering problems. The MKP is a maximization problem and is representative of the subset problems.

The paper is organized as follows. The ACO algorithm and the variant with flying ants is proposed in Section 2. The GPS surveying problem and MKP problem are described in Section 3. The numerical results and a discussion are presented in Section 4. Conclusion remarks are done in Section 5.
2. Ant colony optimization

The idea of the ACO algorithms comes from real ants behavior. When look for a food the ants mark their way back, laying down a chemical substance called pheromone. When there is not a pheromone ants moves essentially at random, but they can detect previously laid pheromone and decide to follow it with high probability. Thus they reinforce it with other quantity of pheromone. If a trail is followed by more ants, it will accumulate more pheromone and will become more attractive. An isolated ant has not its own intelligence, but a group of ants demonstrate swarm intelligence.

The main idea of ACO algorithms is the solved problem to be represented as a graph in such a way that, the solutions to can be represented as a path in a graph. Thus the aim is to find a shortest path under some restrictions, if a minimization problem is solved or longest path under some restrictions, if maximization problem is solved [8,9]. The method is constructive and does not need initial solution. Every ant starts to create their solution from random node from the graph. After that it includes new nodes in the solution applying probabilistic rule called transition probability. The transition probability $p_{i,j}$, to choose the node $j$ when the current node is $i$, is based on the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ of the move, where $i,j = 1,\ldots,n$, $n$ is the number of the nodes of the graph of the problem. The heuristic information is problem dependent and normally is an appropriate combination of the problem parameters.

$$p_{i,j} = \frac{\tau_{i,j}\eta_{i,j}}{\sum_{k \in U} \tau_{i,k}\eta_{i,k}}$$

where $U$ is the set of unused nodes of the graph.

The higher the value of the pheromone and the heuristic information, the more advantageous is to select this move and resume the search. In the beginning, the initial pheromone level is set to a small positive constant value $\tau_0$, later, the ants update this value after completing the construction stage. The ACO algorithms adopt different criteria to update the pheromone level. The main rule is the elements of better solutions to receive more pheromone than others and thus to become more desirable in the next iteration. It is a possibility for intensification the search around best found so far solutions. The pheromone trail update rule is given by:

$$\tau_{i,j} \leftarrow \rho \tau_{i,j} + \Delta \tau_{i,j},$$

where $\rho$ models evaporation in the nature, $0 < \rho < 1$ and $\Delta \tau_{i,j}$ is a new added pheromone which is proportional to the quality of the solution (value of the objective function).

The contribution of this paper is introduction of flying ant. The main idea is to give a possibility of the ant to receive more information about possible movement, to can "see
farther”, and thus to can make better decision, which node to include in the current partial solution.

Let’s the last node, included in the partial solution by an ant is the node $i$. The transition probability $p_{i,j}$, to include the node $j$ in the solution (for $j$ from $U$) is calculated according Formula 1. After, let's "imagine" that the node $j$ is included and the transition probability $p_{j,k}$ the next node $k$ to be included in the solution using Formula 1 is calculated. Then in the new variant of the ACO algorithm the transition probability to include node $j$ is a weighted product of the transition probability to include the node $j$, calculated by Formula 1, and transition probability to include the node $k$, calculated by Formula 1 too, and is calculate in a following way:

$$ P_{i,j} = P_i^\alpha \cdot \max_{k\in U} (p_{i,k}) $$

Thus the algorithm mimic the behavior of the flying ant. The ant includes in its partial solution the node $j$ for which the probability $P_{i,j}$ is maximal. If the there are several nodes for which the transition probability has the maximal value, than the node to be included is chosen, in a random way, between them. This step is repeated until the solution is constructed. The ant stops to include new nodes in the solution, when $P_{i,j}=0$ for every $i,j = 1,\ldots,n$. When all ants construct their solutions, the pheromone is updated according Formula 2. The best solution from the current iteration is compared with the global best solution. If the iteration best solution is better, then the global one, it becomes the new global best solution. In the next iteration the ants repeat the construction stage of the algorithm. The algorithm stops when the end condition is achieved. In most of the cases the end condition is number of iterations.

3. Test problems

The modified ACO algorithm, which mimics the flaying ants is tested on two combinatorial optimization problems, GPS surveying problem and MKP. They are very different problems. The GPS surveying problem is an ordering problem, the solution consists of all nodes of the graph and their order is very important. The MKP problem is a typical representative of subset problems. Only part of the nodes of the graph of the problem are included in the solution and their order is not important.

3.1. GPS surveying problem

The Global Positioning System (GPS) was developed for military purposes in the US. Very soon it was used for civil applications too [15]. It consists of a certain number of satellites that orbit around earth and they are able to communicate with receivers located on
earth. The power that is necessary for establishing a satellite-earth communication allows for estimating the distance between the two communicating machines. Since the machine located on the earth lies over a sphere that does not contain the satellite, a very precise information about the distance between the earth surface and the satellite would allow for determining the precise location of the machine on the earth [16]. Moreover, the precision in locating sensor machines on earth can still be high when the distance information is not very precise, but the communication with more than one satellite can be established [17].

GPS technology can provide very accurate locations for all sensors forming a given sensor network. The related costs can however be too high, when it is necessary to deal with large networks. For this reason the researchers have been trying to design and install local ground networks having the task of recording satellite signals with the aim of decreasing the overall network functioning cost [18,19]. A network is composed by a number of receivers working in different stations at different times. The problem is to find a suitable order for such sessions, which reduces the overall cost. This cost is in fact strictly related to the order of the sessions, because receivers need to be moved from one station to another when stepping from one session to another. Therefore, the distance between two involved stations is important for the computation of the costs. The sessions order is also named sessions schedule. Generally, in order to facilitate the impact of measurement errors in the data, at least two receivers per session are considered [20].

The GPS Surveying Problem can be formalized as follows. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of stations, and let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of receivers, with $m < n$. Sessions can be defined by a function $\sigma: R \rightarrow S$ that associates one receiver to one station. Considering that no more than one receiver should be assigned to the same station, $\sigma$ can be represented by a vector $(\zeta_1, \zeta_2, \ldots, \zeta_m)$ containing, for each of the $m$ receivers, the labels of the chosen stations. Since $m < n$ (and generally fixed to 2 or 3 in the applications), the number of permutations of $m$ objects from $n$ distinguishable ones is $n!/(n-m)!$, which can be huge when the network is large. Notice that not all permutations may actually be possible, depending on the problem at hand.

Let $C$ be an $n \times m$ matrix providing the costs $c(\zeta_u, \zeta_v)$ for moving one receiver from the station $\zeta_u$ to the station $\zeta_v$. This matrix can be symmetric when moving between $\zeta_u$ and $\zeta_v$ is independent from the directionality; the non-symmetric case is however more realistic.

An instance of the GPS surveying problem can be represented by a weighted undirected multi-graph $G = (V, E, c)$ where vertexes represent sessions $\sigma_u$ and arcs $(\sigma_u, \sigma_v)$ indicate the possibility to switch from session $\sigma_u$ to session $\sigma_v$. The upper bound on the cardinality of $V$ is $n!/(n-m)!$, which is the maximum number of possible sessions. The weight associated to the arcs provides the cost $c(\sigma_u, \sigma_v)$ for moving every receiver from the station $\zeta_{u,i}$ to the station $\zeta_{v,i}$, for each $i$: 
The graph $G$ is not simple in general, because it might be feasible to switch from session $\sigma_u$ to session $\sigma_v$, as well as from $\sigma_v$ to $\sigma_u$, but with a different total cost. The problem consists in finding an optimal path on $G$, i.e. a path for which all selected arcs give the minimal total cost, while covering the entire vertex set $V$ [18].

### 3.2. Multiple knapsack problem

A lot of practical problems can be defined as Multiple Knapsack Problem (MKP), therefore it receives wide attention from operation research community. Some of the applications are resource allocation in distributed systems, capital budgeting and cutting stock problems. MKP can be seen as a general model for any kind of binary problems with positive coefficients [21,22]. Let's MKP is defined as a resource allocation problem, where there are $m$ resources (the knapsacks) and $n$ objects. Each object $j$ has a profit $p_j$. Each resource has its own budget (knapsack capacity) and consumption $r_{ij}$ of resource $j$ by object $i$. The aim is to maximizing the profit, while working with a limited budget.

The MKP can be formulated as follows:

$$
\max \sum_{j=1}^{n} p_j x_j
$$
subject to $\sum_{j=1}^{n} r_{ij} x_j \leq c_i \quad i = 1, ..., m$
\[ x_j \in \{0,1\} \quad j = 1, ..., n \quad (4) \]

There are $m$ constraints in this problem, so MKP is also called $m$-dimensional knapsack problem. Let $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$, with $c_i \geq 0$ for all $i \in I$. A well-stated MKP assumes that $p_j > 0$ and $r_{ij} \leq c_i \leq \sum r_{ij}$ for all $i \in I$ and $j \in J$. Note that the $[r_{ij}]_{m \times n}$ matrix and $[c_i]_m$ vector are both non-negative.

The elements of MKP solutions have no particular order. It is a main difference with GPS surveying problem. Therefore a partial solution is represented by $S = \{i_1, i_2, \ldots, i_j\}$ and the most recent elements incorporated to $S$, $i_j$ needs not be involved in the process for selecting the next element. Moreover, solutions for ordering problems have a fixed length as is searched for a permutation of a known number of elements. Solutions for MKP, do not have a fixed length. The graph of the problem is defined as follows: the nodes correspond to the items, the arcs fully connect nodes.
4. Numerical results and discussion

The idea to mimic the flying ant is tested on two very different test problems, the GPS surveying problem, which is a representative of ordering problems and MKP, which is a representative of subset problems. The proposed flying ant can see two steps ahead and includes one new node in their partial solution. The transition probability is calculated as in Formula 3. The achieved results by flying ants are compared with the results achieved by traditional ant algorithm. In both cases the algorithm parameters are the same and tests are run on the same computer, Pentium desktop computer at 2.8 GHz with 4 GB of memory. The reported results are average over 30 independent runs with every one of the tests and algorithms. Used tests consists of 38, 71, 100, 171 and 323 stations for GPS surveying problem and 100 objects and 10 constraints for MKP. The test are from “OR-Library” available within WWW access at http://mscmga.ms.ic.ac.uk/jeb/orlib.

<table>
<thead>
<tr>
<th>Stations</th>
<th>38</th>
<th>71</th>
<th>100</th>
<th>171</th>
<th>323</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional ACO</td>
<td>896.5</td>
<td>902.56</td>
<td>40696.5</td>
<td>1647.16</td>
<td>1669.8</td>
</tr>
<tr>
<td>Flaying ACO</td>
<td>1024.66</td>
<td>1053.06</td>
<td>51246.93</td>
<td>4118</td>
<td>3704</td>
</tr>
</tbody>
</table>

Table 1

<table>
<thead>
<tr>
<th>Tests</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>Test 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional ACO</td>
<td>21989.43</td>
<td>22081.36</td>
<td>21027.63</td>
<td>21635.3</td>
<td>21717.93</td>
</tr>
<tr>
<td>Flaying ACO</td>
<td>17107</td>
<td>17067</td>
<td>17805</td>
<td>17849</td>
<td>18167</td>
</tr>
</tbody>
</table>

Table 2

In Table 1 and Table 2 are reported achieved results by both algorithms, applied on GPS surveying problem and MKP respectively. Can be observed that the flying ant achieves wors results compared to traditional ACO, when the parameter $\alpha=1$, for the GPS surveying problem, and improves the results for MKP problem. $\alpha=1$ means that the probability to chose the next node and the probability to choose the second node have same influence in the Formula 3. We can explain this result for GPS surveying problem by loss of diversification in a search, which is a very important especially for ordered problems. One of the most important element of the ant algorithm is diversification in the search. When the parameter $\alpha=1$, the ants receive more information at the expense of diversification. The next step is to improve the algorithm performance by increasing the influence of the probability to chose the next node and to decrease the influence of the probability to chose the second node playing with the value of the parameter $\alpha$, Formula 3. Thus, diversification of the search is increased, without to loss the information received by the flying ant. Let the parameter $\alpha$ is equal to the
number of the nodes of the graph of the problem. Then as is seen from the Table 3 the flying ant achieves better solutions than traditional ACO algorithm, testing on GPS surveying problem, only for the smaller test with 38 stations the achieved solution is not better and statistically the same. Testing on MKP the achieved results are statistically similar with these, achieved by flying ant ACO algorithm, when parameter $\alpha=1$. A variant of the ACO algorithm, where the ant can receive information from more than two nodes ahead is tested, but there are not significant improvement of achieved results and the time increases exponentially according the number of nodes ahead.

<table>
<thead>
<tr>
<th>Stations</th>
<th>38</th>
<th>71</th>
<th>100</th>
<th>171</th>
<th>323</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional ACO</td>
<td>896.5</td>
<td>902.56</td>
<td>40696.5</td>
<td>1647.16</td>
<td>1669.8</td>
</tr>
<tr>
<td>Flaying ACO</td>
<td>901.6</td>
<td>871.6</td>
<td>40250.03</td>
<td>1628.53</td>
<td>1438.86</td>
</tr>
</tbody>
</table>

Table 3

5. Conclusion

In this paper is proposed an amendment in traditional ant algorithm. The main difference is calculation of transition probability. The new idea is to mimic the flying ant and the ant to can receive information from more than one node ahead. The influence of the information from the neighbor nodes and from second nodes is controlled. Thus diversification in a search process is preserved and the ant receives more information and can make better decision which node to chose to be included in a partial solution. The received results are encouraging.

In a future the value of the parameter $\alpha$ will be examined, its influence on achieved results and its correlation with other algorithm parameters.

BIBLIOGRAPHY

Streszczenie

Artykuł przedstawia "latające" mrówki w problemie optymalizacji algorytmów mrówkowych. W tradycyjnych podejściach dla algorytmów mrówkowych agenci (mrówki) budują swoje rozwiązanie w kolejnych krokach. W zaproponowanym podejściu optymalizacji algorytmu mrówkowego agenci podejmują decyzję na podstawie więcej niż jednego kroku, jednakże tylko jeden element wprowadzany jest do rozwiązania.


Addresses

Stefka FIDANOVA: Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str. bl.25A, 1113 Sofia, Bulgaria, stefka@parallel.bas.bg
Krassimir ATANASSOV: Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, Acad. G. Bonchev str. bl.105, 1113 Sofia, Bulgaria, krat@bas.bg