Adjoint Estimatoin of Ant Colony Optimization Starts Methods

Stefka Fidanova, Pencho Marinov and Krassimir Atanassov

1 Introduction

Many combinatorial optimization problems are fundamentally hard. This is the most typical scenario when it comes to realistic and relevant problems in industry and science. Examples of optimization problems are Traveling Salesman Problem [6], Vehicle Routing [7], Minimum Spanning Tree [5], Multiple Knapsack Problem [4], etc. They are NP-hard problems and in order to obtain solution close to the optimality in reasonable time, metaheuristic methods are used. One of them is Ant Colony Optimization (ACO) [2].
ACO algorithms have been inspired by the real ants behavior. In nature, ants usually wander randomly, and upon finding food return to their nest while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. However, as time passes, the pheromone starts to evaporate. The more time it takes for an ant to travel down the path and back again, the more time the pheromone has to evaporate and the path to become less prominent. A shorter path, in comparison will be visited by more ants and thus the pheromone dencity remains high for a longer time.
ACO is implemented as a team of intelligent agents which simulate the ants behavior, walking around the graph representing the problem to solve using mechanisms of cooperation and adaptation. ACO algorithm requires to define the following [1, 3]:

- The problem needs to be represented appropriately, which would allow the ants to incrementally update the solutions through the use of a probabilistic transition rules, based on the amount of pheromone in the trail and other problem specific knowledge. It is also important to enforce a strategy to construct only valid solutions corresponding to the problem definition.
- A problem-dependent heuristic function, that measures the quality of components that can be added to the current partial solution.
- A rule set for pheromone updating, which specifies how to modify the pheromone value.

- A probabilistic transition rule based on the value of the heuristic function and the pheromone value, that is used to iteratively construct a solution.

The structure of the ACO algorithm is shown by the pseudocode below. 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 $i, j = 1, \dots, n$.

$$p_{i,j} = \frac{\tau_{i,j}^a \eta_{i,j}^b}{\sum_{k \in Unused} \tau_{i,k}^a \eta_{i,k}^b},$$

where $Unused$ is the set of unused nodes of the graph.
The higher the value of the pheromone and the heuristic information, the more profitable it is to select this move and resume the search. In the beginning, the initial pheromone level is set to a small positive constant value τ_0; later, the ants update this value after completing the construction stage. ACO algorithms adopt different criteria to update the pheromone level.

```
Ant Colony Optimization
Initialize number of ants;
Initialize the ACO parameters;
while not end-condition do
    for k=0 to number of ants
        ant k choses start node;
        while solution is not constructed do
            ant k selects higher probability node;
        end while
    end for
    Update-pheromone-trails;
end while
```

Figure 1: Pseudocode for ACO

The pheromone trail update rule is given by:

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

where ρ models evaporation in the nature and $\Delta \tau_{i,j}$ is new added pheromone which is proportional to the quality of the solution.
Our novelty is to use estimations of start nodes with respect to the quality of the solution and thus to better menage the search process. On the basis of the estimations we offer several start strategies and their combinations.

2 Start Strategies

The known ACO algorithms create a solution starting from random node. But for some problems, especially subset problems, it is important from which node the search

process starts. For example if an ant starts from node which does not belong to the optimal solution, probability to construct it is zero. Therefore we offer several start strategies.
Let the graph of the problem has m nodes. We divide the set of nodes on N subsets. There are different ways for dividing. Normally, the nodes of the graph are randomly enumerated. An example for creating of the nodes subsets, without lost of generality, is: the node number one is in the first subset, the node number two is in the second subset, etc. the node number N is in the $N-th$ subset, the node number $N+1$ is in the first subset, etc. Thus the number of the nodes in the separate subsets are almost equal. We introduce estimations $D_j(i)$ and $E_j(i)$ of the node subsets, where $i \geq 2$ is the number of the current iteration. $D_j(i)$ shows how good is the j^{th} subset and $E_j(i)$ shows how bad is the j^{th} subset. $D_j(i)$ and $E_j(i)$ are weight coefficients of $j-th$ node subset $(1 \leq j \leq N)$, which we calculate by the following formulas:

$$D_j(i) = \phi.D_j(i-1) + (1-\phi).F_j(i),$$
$$E_j(i) = \phi.E_j(i-1) + (1-\phi).G_j(i),$$

where $i \geq 1$ is the current process iteration and for each j $(1 \leq j \leq N)$:

$$F_j(i) = \begin{cases} \frac{f_{j,A}}{n_j} & \text{if } n_j \neq 0 \\ F_j(i-1) & \text{otherwise} \end{cases}, \tag{1}$$

$$G_j(i) = \begin{cases} \frac{g_{j,B}}{n_j} & \text{if } n_j \neq 0 \\ G_j(i-1) & \text{otherwise} \end{cases}, \tag{2}$$

and $f_{j,A}$ is the number of the solutions among the best $A\%$, and $g_{j,B}$ is the number of the solutions among the worst $B\%$, where $A+B \leq 100$, $i \geq 1$ and

$$\sum_{j=1}^{N} n_j = n,$$

where n_j $(1 \leq j \leq N)$ is the number of solutions obtained by ants starting from nodes subset j. Initial values of the weight coefficients are: $D_j(1) = 1$ and $E_j(1) = 0$. The parameter ϕ, $0 \leq \phi \leq 1$, shows the weight of the information from previous iterations and from current iteration. When $\phi = 0$ only the information from current iteration is taken in to account. If $\phi = 0.5$ the influence of previous versus current iterations is equal. When $\phi = 1$ only the information from previous iterations is taken in to account. When $\phi = 0.25$ the weight of the information from previous iterations is three times less than this one of the current iteration. When $\phi = 0.75$ the weight of the previous iterations is three times higher than this one of the current iteration. The balance between the weights of the previous and current iterations is important. At the beginning when the current best solution is far from the optimal one, some of the node subsets can be estimated as good. Therefore, if the value of the parameter ϕ is too high the estimation can be distorted. If the weight of the current iteration

is too high than in the estimation are ignored good and bad solutions from previous iterations, which can distort estimation too.
We try to use the experience of the ants from previous iteration to choose the better starting node. Other authors use this experience only by the pheromone, when the ants construct the solutions. Let us fix threshold E for $E_j(i)$ and D for $D_j(i)$, than we construct several strategies to choose start node for every ant, the threshold E increase every iteration with $1/i$ where i is the number of the current iteration:

1 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for current iteration and we choose the starting node randomly from $\{j \mid j$ is not forbidden$\}$;

2 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for current simulation and we choose the starting node randomly from $\{j \mid j$ is not forbidden$\}$;

3 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for K_1 consecutive iterations and we choose the starting node randomly from $\{j \mid j$ is not forbidden$\}$;

4 Let $r_1 \in [0.5, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ we randomly choose node from subset $\{j \mid D_j(i) > D\}$, otherwise we randomly chose a node from the not forbidden subsets, r_1 is chosen and fixed at the beginning.

5 Let $r_1 \in [0.5, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ we randomly choose node from subset $\{j \mid D_j(i) > D\}$, otherwise we randomly chose a node from the not forbidden subsets, r_1 is chosen at the beginning and increase with r_3 every iteration.

Where $0 \leq K_1 \leq$"number of iterations" is a parameter. If $K_1 = 0$, than strategy 3 is equal to the random choose of the start node. If $K_1 = 1$, than strategy 3 is equal to the strategy 1. If $K_1 =$"maximal number of iterations", than strategy 3 is equal to the strategy 2.
We can use more than one strategy for choosing the start node, but there are strategies which can not be combined. We distribute the strategies into two sets: $St1 = \{strategy1,\ strategy2,\ strategy3\}$ and $St2 = \{strategy5,\ strategy6\}$. The strategies from same set can not be used at once. Thus we can use strategy from one set or combine it with strategies from other set. Exemplary combinations are ($strategy1$), ($strategy2$; $strategy5$), ($strategy3$; $strategy6$).

3 Conclusion

In this paper we address the modelling of the process of ant colony optimization method by using adjoint estimations, combining five start strategies. So, the start node of each ant depends of the goodness of the respective region. In a future we will focus on parameter settings which manage the starting procedure. We will investigate on influence of the parameters to algorithm performance. We will apply our modification of ACO algorithm on various classes of problems. We will investigate the influence of the estimations and start strategies on the achieved results.

Acknowledgments: This work has been partially supported by the Bulgarian National Scientific Fund under the grants ID-Modeling Processes with fixed development rules and TK-Effective Monte Carlo Methods for large-scale scientific problems.

References

[1] Bonabeau E., Dorigo M., Theraulaz G., Swarm Intelligence: From Natural to Artificial Systems, New York,Oxford University Press, 1999.

[2] Dorigo M., Gambardella L.M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation 1,53-66,1997.

[3] Dorigo M., Stutzle T., Ant Colony Optimization, MIT Press, 2004.

[4] Fidanova S., Evolutionary Algorithm for Multiple Knapsack Problem, Int. Conference Parallel Problems Solving from Nature, Real World Optimization Using Evolutionary Computing, ISBN No 0-9543481-0-9,Granada, Spain, 2002.

[5] Reiman M., Laumanns M., A Hybrid ACO algorithm for the Capacitated Minimum Spanning Tree Problem, In proc. of First Int. Workshop on Hybrid Metahuristics, Valencia, Spain, 2004, 1-10.

[6] Stutzle T. Dorigo M., ACO Algorithm for the Traveling Salesman Problem, In K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux eds., Evolutionary Algorithms in Engineering and Computer Science, Wiley, 163-183, 1999.

[7] Zhang T., Wang S., Tian W., Zhang Y., ACO-VRPTWRV: A New Algorithm for the Vehicle Routing Problems with Time Windows and Re-used Vehicles based on Ant Colony Optimization, Sixth International Conference on Intelligent Systems Design and Applications, IEEE press, 2006, 390-395.