Fifteenth Int. Conf. on IFSs, Burgas, 11-12 May 2011
NIFS 17 (2011), 2, 75-81

Intuitionistic fuzzy estimation of the ant colony optimization starting points: Part 2

Stefka Fidanova and Pencho Marinov

Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 25A, 1113 Sofia, Bulgaria
E-mails: {stefka,pencho}@parallel.bas.bg

Abstract: The ability of ant colonies to form paths for carrying food is rather fascinating. The problem is solved collectively by the whole colony. This ability is explained by the fact that ants communicate in an indirect way by laying trails of pheromone. The higher the pheromone trail within a particular direction, the higher the probability of choosing this direction. The collective problem solving mechanism has given rise to a metaheuristic referred to es Ant Colony Optimization (ACO). On this work we use intoitionistic fuzzy estimation of start nodes with respect to the quality of the solution. Various start strategies are offered. Sensitivity analysis of the algorithm behavior according estimation parameters is made. As a test problem is used Multidimensional (Multiple) Knapsack Problem (MKP).
Keywords: Ant colony optimization, Intuitionistic fuzzy sets, Knapsack problem.
AMS Classification: 03E72, 90C59, 68T20

1 Introduction

A large number of real-life optimization problems in science, engineering, economics, and business are complex and difficult to solve. They can not be solved in an exact manner within a reasonable amount of computational resources. Using approximate algorithms is the main alternative to solve this class of problems. The approximate algorithms are specific heuristics, which are problem dependent, and metaheuristics, which are more general approximate algorithms applicable to a large variety of optimization problems. One of the most successful metaheuristic is Ant Colony Optimization (ACO) [4].

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 travelling at random, but 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.In comparison, a shorter path will be visited by more ants and thus the pheromone density remains high for a longer time.

ACO is implemented as a team of intelligent agents which simulate the ants behavior, walking around the graph that represents the problem for solving using mechanisms of cooperation and adaptation. Examples of such optimization problems are the Travelling Salesman Problem [13], Vehicle Routing [15], Minimum Spanning Tree [11], Multiple Knapsack Problem [6], etc.

The transition probability $p_{i,j}$, to choose the node j from the graph of the problem, 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 $i, j = 1, \ldots, n$.

$$p_{i,j} = \frac{\tau_{i,j}^{a}\eta_{i,j}^{b}}{\sum\limits_{k \in Unused} \tau_{i,k}^{a}\eta_{i,k}^{b}}, \tag{1}$$

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. The pheromone trail update rule is given by:

$$\tau_{i,j} \leftarrow \rho\tau_{i,j} + \Delta\tau_{i,j}, \tag{2}$$

where ρ models evaporation in the nature and $\Delta\tau_{i,j}$ is newly added pheromone, which is proportional to the quality of the solution.

This paper uses intuitionistic fuzzy estimations of start nodes with respect to the quality of the solution and thus to better manage the search process. It is a continuation of our previous work [9]. Here, we specify the intuitionistic fuzzy parameters, which manage the starting process and their influence to the quality of the achieved solutions.

In intuitionistic fuzzy (IF) logic any statement can be evaluated with a degree of validuty and degree of non-validity. On the basis of the estimations we offer several start strategies and their combinations. As a benchmark problem Multiple Knapsack Problem (MKP) is used, which is a representative of the class of subset problems. A lot of real world problems can be represented by it. Moreove,r MKP arises as a subproblem in many optimization problems.

The rest of the paper is organized as follows. In section 2, intuitionistic fuzzy estimation of start node is introduced and several start strategies are proposed. In section 3, the strategies are applied to MKP and sensitivity analysis of the algorithm according to the strategy parameters is made. In the end some conclusions and directions for future work are proposed.

2 Intuitionistic fuzzy estimation

The known ACO algorithms generate a solutions starting from a 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 that does not belong to the optimal solution, probability to construct it is zero. In this paper, intuitionistic fuzzy estimation of start nodes is offered and after that several start strategies are proposed. The aim is to use the ants' experience from previous iterations in choosing the better starting node. Other authors use this experience only by the pheromone, when the ants construct the solutions.

2.1 Intuitionistic fuzzy sets

In the beginning, we will define the concept of intutionistic fuzzy sets [1]. Let X, Y and Z be ordinary finite non-empty sets. Let X be a given set. An intuitionistic fuzzy set in X is an expression A given by:

$$A = \{< x, \mu_A(x), \nu_A(x) > | x \in A\}, \tag{3}$$

where

$$\mu_A : X \to [0, 1] \quad \nu_A : X \to [0, 1]$$

such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1$ for all $x \in A$. The numbers $\mu_A(x)$ and $\nu_A(X)$ denote respectively the degree of membership and the degree of non-membership of the element x in the set A. When $\nu_A(x) = 1 - \mu_A(x)$, set A is a fuzzy set. When $\nu_A(x) \leq 1 - \mu_A(x)$, set A is a intuitionistic fuzzy.

Let the graph of the problem have m nodes. The set of nodes is divided into N subsets. There are different ways for making this division. Normally, the nodes of the graph are randomly enumerated. An example for creating of the subsets, without lost of generality, is the following: the node number one is in the first subset, the node number two - 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 nodes in the separate subsets are almost equal. After the first iteration the estimations $D_j(i)$ and $E_j(i)$ are introduced of the node subsets, where $i \geq 2$ is the number of the current iteration and $D_j(i)$ and $E_j(i)$ are weight coefficients of j-th node subset $(1 \leq j \leq N)$, which are calculated by the following formulas:

$$D_j(i) = \varphi.D_j(i-1) + (\psi - \varphi).F_j(i), \tag{4}$$

$$E_j(i) = \varphi.E_j(i-1) + (\psi - \varphi).G_j(i), \tag{5}$$

where $i \geq 2$ is the current process iteration and for each j $(1 \leq j \leq N)$:

$$F_j(i) = \begin{cases} \dfrac{f_{j,A}}{n_j} & \text{if } n_j \neq 0 \\ F_j(i-1) & \text{otherwise} \end{cases}, \tag{6}$$

$$G_j(i) = \begin{cases} \dfrac{g_{j,B}}{n_j} & \text{if } n_j \neq 0 \\ G_j(i-1) & \text{otherwise} \end{cases}, \tag{7}$$

where $f_{j,A}$ is the number of the solutions among the best $A\%$, $g_{j,B}$ is the number of the solutions among the worst $B\%$, where $A + B \leq 100$, $i \geq 2$ and

$$\sum_{j=1}^{N} n_j = n, \tag{8}$$

where n_j $(1 \leq j \leq N)$ is the number of solutions obtained by ants starting from nodes from subset j and n is the number of ants. Initial values of the weight coefficients are: $D_j(1) = 1$ and $E_j(1) = 0$. The parameter φ, $0 \leq \varphi \leq 1$, shows the weight of the information from the previous iterations and from the last iteration. When $\varphi = 0$ only the information from the last iteration is taken into account. If $\varphi = 0.5 * \psi$, the influence of the previous iterations versus the last one is equal. When $\varphi = \psi$, only the information from the previous iterations is taken in to account. The balance between the weights of the previous iterations and the last one is important. In 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 last iteration is too high, then information for good and bad solutions from previous iterations is ignored, which can distort estimation too.

We try to use the ants' experience from previous iterations to choose the better starting node. Other authors use this experience only by the pheromone, when the ants construct the solutions [5].

2.2 Start strategies

Let us fix threshold E for $E_j(i)$ and D for $D_j(i)$, then we construct several strategies to choose start node for every ant. On every iteration the threshold E increases 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 $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ be a random number. If $r_2 > r_1$, we randomly choose a node from subset $\{j \mid D_j(i) > D\}$, otherwise we randomly choose a node from the not forbidden subsets, r_1 is chosen and fixed in the beginning.

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

where $0 \leq K_1 \leq$ "number of iterations" is a parameter. If $K_1 = 0$, then strategy 3 is equal to the random choice of the start node. If $K_1 = 1$, then strategy 3 is equal to strategy 1. If $K_1 =$"maximal number of iterations", then strategy 3 is equal to 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: $St1 = \{$strategy 1, strategy 2, strategy 3$\}$ and $St2 = \{$strategy 4, strategy 5$\}$. The strategies from same set can not be used at once. When we combine strategies from $St1$ and $St2$, first we apply the strategy from $St1$ and according it some of the regions (node subsets) become forbidden, and after that we choose the starting node from the not forbidden subsets according to the strategy from $St2$.

3 Experimental results

The intuitionistic fuzzy estimation and start strategies performance are analyzed in this section. As a test Multiple Knapsack Problem (MKP) is used because it is a subset problem. The Multiple Knapsack Problem has numerous applications in theory as well as in practice. It also arises as a subproblem in several algorithms for more complex problems and these algorithms will benefit from any improvement in the field of MKP. The following major applications can be mentioned: problems in cargo loading, cutting stock, bin-packing, budget control and financial management may be formulated as MKP. In [12], it is proposed to use the MKP in fault tolerance problem and in [3] is designed a public cryptography scheme whose security depends on the difficulty of solving the MKP. In [10] is mentioned that two-processor scheduling problems may be solved as a MKP. Other applications are industrial management, naval, aerospace, computational complexity theory.

The MKP can be thought as a resource allocation problem, where there are m resources (the knapsacks) and n objects and every object j has a profit p_j. Each resource has its own budget c_j (knapsack capacity) and consumption r_{ij} of resource i by object j. The aim is maximizing the sum of the profits, while working with a limited budget.

The MKP can be formulated as follows:

$$\begin{aligned} &\max \sum_{j=1}^{n} p_j x_j \\ &\sum_{j=1}^{n} r_{ij} x_j \leq c_i \;\; i = 1, \ldots, m \\ &x_j \in \{0, 1\} \;\; j = 1, \ldots, n \end{aligned} \tag{9}$$

x_j is 1 if the object j is chosen and 0, otherwise.

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 $r_{ij} \leq c_i \leq \sum_{j=1}^{n} r_{ij}$ 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.

In the MKP one is not interested in solutions giving a particular order. 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 need not be involved in the process of selecting the next element. Moreover, solutions of ordering problems have a fixed length as one searches for a permutation of a known number of elements. Solutions for MKP, however, 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 the nodes. Fully connected graph means that after the object i one can chooses the object j for every i and j if there are enough resources and object j is not chosen yet.

The computational experience of the ACO algorithm is shown using 10 MKP instances from "OR-Library" available at `http://people.brunel.ac.uk/~mastjjb/jeb/orlib`, with 100 objects and 10 constraints. To provide a fair comparison for the above implemented ACO algorithm, a predefined number of iterations, $k = 100$, is fixed for all the runs. The developed technique has been coded in C++ language and implemented on a Pentium 4 (2.8 Ghz). The parameters are fixed as follows: $\rho = 0.5$, $a = 1$, $b = 1$, number of used ants is 20, $A = 30$, $B = 30$, $D = 1.5$, $E = 0.5$, $K_1 = 5$, $R = 0.5$, $r_3 = 0.01$. The values of ACO parameters (ρ, a, b) are from [7] and it is experimentally found that they are best for MKP. The tests are run with 1, 2, 4, 5 and 10 nodes within the node subsets. The following combinations of parameters φ and ψ are used in our previous work [9]: $(0.125, 0.25)$, $(0.125, 0.5)$, $(0.125, 0.825)$, $(0.25, 0.5)$, $(0.25, 0.75)$, $(0.25, 0.825)$, $(0.5, 0.75)$, $(0.5, 0.825)$, $(0.75, 0.825)$, $(0.5, 1)$. For every experiment, the results are obtained by performing 30 independent runs, then averaging the fitness values obtained in order to ensure statistical confidence of the observed difference. The computation time taken by start strategies is negligible with respect to running time of the algorithm. Tests with all combinations of strategies and with a random start (12 combinations) are run. Thus, we perform 180 000 tests.

For every test problem, the average result achieved by some strategy is better than without any strategy. For fair comparison, the difference d between the worst and the best average result

Table 1: Estimation of the rate according φ and ψ

φ / ψ	0.40	0.450	0.500	0.550	0.6
0.65	85	84	86	86	86
0.70	92	85	85	86	87
0.75	93	**94**	93	90	90
0.80	89	87	89	92	91
0.85	84	87	89	90	93

for every problem is divided to 10. If the average result for some strategy is between the worst average result and the worst average plus $d/10$ it is evaluated with 1. If it is between the worst average plus $d/10$ and worst average plus $2d/10$, it is evaluated with 2, and so on. If it is between the best average minus $d/10$ and the best average, it is evaluated with 10. Thus, for a test problem the achieved results for every strategy and every nodes division is evaluated from 1 to 10. After that is summed the rate of all test problems for every strategy and every nodes devision. So, their rate becomes between 10 and 100 (see Table 1). The best results are achieved when there is only one node in the node subset, for all combinations of the values of φ and ψ. When the node subsets consist of 1 node, preferably bad or preferably good solutions start from them. In [9] we observe that the best ranking is when $\varphi = 0.5$, $\psi = 0.75$, only one node in node subsets and strategy 3. Therefore, in this paper we will specify the ranking around the best values for φ and ψ. We will use smaller step when we change the values of φ and ψ. Therefore, in Table 1 we show only this case.

On Table 1 we observe that the best rate is when $\varphi = 0.45$ and $\psi = 0.75$.

4 Conclusion

This paper is addressed to ant colony optimization algorithm with controlled start using intuitionistic fuzzy estimation. So, the start node of each ant depends of the goodness of the respective region. We have analyzed the behavior of the ACO algorithm. The rate of the achieved solutions is higher when the rate of the fuzziness is not very high and when the difference between the parameters ψ and φ is not very big. In this case, there is good balance between results achieved in previous iterations and current iteration. We specify the rate using smaller step around the best values of the parameters φ and ψ. The future work will be focused on parameter settings which manage the starting procedure. It will investigate the influence of the parameters on the algorithm's performance. The aim is to study in detail the relationships between the start nodes and the quality of the achieved solutions.

Acknowledgments

This work has been partially supported by the Bulgarian National Science Fund under Grant DID 02/29 “Modeling Processes with fixed development rules” and Grant DTK 02/44 “Effective Monte Carlo Methods for large-scale scientific problems”.

References

[1] Atanassov, K. Intuitionistic Fuzzy Sets, Springer, Heidelberg, 1999.

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

[3] Diffe, W., Hellman, M. E., New direction in cryptography. IEEE Trans Inf. Theory. IT-36, 1976, 644–654.

[4] 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.

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

[6] 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.

[7] Fidanova, S., Ant colony optimization and multiple knapsack problem, in: Renard, J.Ph. (Eds.), Handbook of Research on Nature Inspired Computing for Economics ad Management, Idea Grup Inc., ISBN 1-59140-984-5, 2006, 498–509.

[8] Fidanova S., Atanassov K., Marinov P., Parvathi R., Ant Colony Optimization for Multiple Knapsack Problems with Controlled Starts, Int. J. Bioautomation, Vol 13(4), 271–280.

[9] Fidanova S., Atanassov K., Marinov P., Parvathi R.,Intuitionistic Fuzzy Estimation of the Ant Colony Optimization Starting Points: Part 2, Large Scale sciantific Computing, Lecture Notes in Computer Science (in press).

[10] Martello, S., Toth, P., A mixtures of dynamic programming and branch-and-bound for the subset-sum problem, Management Science 30, 1984, 756–771.

[11] 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.

[12] Sinha, A., Zoltner, A. A., The multiple-choice knapsack problem, J. Operational Research 27, 1979, 503–515.

[13] 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.

[14] Stutzle T. and Hoos H. H.: MAX-MIN Ant System, In Dorigo M., Stutzle T., Di Caro G. (eds). Future Generation Computer Systems, Vol 16, 2000, 889–914.

[15] 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.