
Ant colony optimization with re-initialization

Matej Ciba, Ivan Sekaj

Institute of Control and Industrial Informatics, Bratislava, Slovakia

Email address:

bigmato@centrum.sk(M. Ciba), ivan.sekaj@stuba.sk(I. Sekaj)

To cite this article:

Matej Ciba, Ivan Sekaj. Ant Colony Optimization with Re-Initialization. *Automation, Control and Intelligent Systems*. Vol. 1, No. 3, 2013, pp. 59-63. doi: 10.11648/j.acis.20130103.14

Abstract: This contribution introduces an Ant Colony Optimization (ACO) algorithm with re-initialization mechanism. The whole search process is broken by re-initialization into shorter semi-independent steps called “macro cycles”. The length of macro cycle depends on pheromone accumulation and can be adjusted by a user parameter. It is shown that re-initialization mechanism prevents ACO algorithm from pheromone saturation and consecutive stagnation. This approach avoids overhead caused by algorithm run with excessive pheromone values where further exploration is hardly possible. The solution offers lower CPU cost of the search process and enables automation of heuristic search especially in changing environments like dynamic networks. The efficiency of proposed method is demonstrated on a path minimization problem on 50 node graph.

Keywords: Ant Colony Optimization, Re-Initialization, Pheromone Saturation, Minimal Path Search

1. Introduction

Wide range of problems like Routing problem, Assignment problem, Scheduling problem and others can be transformed into graph representation. Exact algorithms for instance Dijkstra or Bellman-Ford appear to be slow and inefficient on large scale graphs. It is not always necessary to find the best possible solution. Instead good quality solution in reasonable time is often preferred. This is the reason why heuristic methods become popular. Among the well-known graph search algorithms that utilizes heuristic are A* search [1] and ACO algorithm.

Ant colony optimization represents an efficient tool for optimization and design of graph oriented problems. It is a multi-agent meta-heuristic approach and was first purposed in 1991 by Dorigo at al. as Ant system (AS) algorithm.

Ants are using indirect form of communication via trailing pheromone. As ants are passing the terrain (graph) they mark used routes (arcs of the graph) by chemical substance called pheromone. Other ants can sense the substance and follow the same track.

During the search process each ant sets off from the ant colony (start position) and moves to search food (destination). The aim is to find the shortest path. On their way back they use the same way from which abundant loops have been removed. The amount of pheromone (1)

$\Delta\tau_{ij}^k(t)$ the k -th ant produces is inversely proportional to the tour length $L^k(t)$.

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q / L^k(t) & \text{if } (i, j) \in T^k(t) \\ 0 & \text{if } (i, j) \notin T^k(t) \end{cases} \quad (1)$$

$T^k(t)$ is the tour generated by ant k , Q is a constant and tuple (i, j) denotes beginning and termination node of an arc. All pheromone tracks (2) are preserved by arcs of the graph

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (2).$$

Where $\rho \in (0,1)$ is the pheromone persistence ($1 - \rho$ is evaporation rate) and m is the number of ants. Evaporation rate is a user adjusted parameter and affects pheromone durability; i.e. how long the acquired information will be available. Too high values causes random search (quick evaporation), too low values get algorithm stock in local optimum.

An ant in each node has to make a decision which arc to take. This probabilistic choice is called *random proportional rule* (3). Ant chooses the next node only from its neighborhood N_i^k except the node it visited previously.

$$p_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t) + \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha(t) + \eta_{ij}^\beta} \quad (3)$$

The probability $p_{ij}(t)$ of choosing the particular arc (i, j) depends on pheromone $\tau_{ij}(t)$ and the heuristic η_{ij} values associated with the arc. Symbols α and β are weight parameters and represents balance between ants' gathered knowledge and user preferred area. Heuristic values η_{ij} affect probability only at the beginning when pheromone values are low. The more pheromone is located on particular arc, the more attractive it appears.

2. ACO Algorithms with Diversity Control

Parameters for setting the balance between exploration and exploitation belong to the most important variables for majority of heuristic algorithms.

Even author of ACO realized the need for diversity control and introduced Ant Colony System (ACS) [3] which differs from original Ant System in three main aspects: (i) *pseudo-random proportional* rule, (ii) global and (iii) local pheromone update rule.

Pseudo-random proportional rule uses random uniformly distributed variable $q \in (0,1)$ which is compared with a tunable parameter $q_0 \in \langle 0,1 \rangle$ (4).

$$j = \begin{cases} \arg \max_{j \in N_i^k} \{ \tau_{ij}(t) \eta_{ij}^\beta \} & \text{if } q \leq q_0 \\ J & \text{otherwise} \end{cases} \quad (4)$$

where J is a random variable selected according to the probability distribution given by (3). Low q_0 values prefer balanced exploration of new paths.

Since in a *global pheromone update rule* only the *best-so-far* ant is allowed to add pheromone the pheromone value is weighted by pheromone persistence parameter ρ to prevent rapid pheromone accumulation (5).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^{bs}(t); \forall (i,j) \in T^{bs} \quad (5)$$

According to *local pheromone update rule* each ant immediately after crossing the arc reduces the pheromone value (6). This supports further exploration since the crossed arc becomes less desirable for the following ants.

$$\tau_{ij}(t+1) = (1-\xi)\tau_{ij}(t) + \xi\tau_0 \quad (6)$$

Nakamichi and Arita [4] chose another approach to diversity control. Instead of controlling diversity in depositing pheromone they control diversity in finding tours. Authors introduced a mechanism of *random selection* in addition to the probabilistic selection. Random selection is a simple operation which selects next node from the list of unvisited neighbors with the equal probability. *Random selection rate*

r is probability with which random selection operates each time an ant has to choose the next node and represents a user parameter which adjusts the balance between exploration and exploitation.

Kumar, et al. [5] enriched AS with (i) prevention of quick convergence and (ii) stagnation avoidance mechanisms.

The mechanism for prevention of quick convergence (i) is based on pseudo-random proportional rule (4), but the tunable parameter q_0 is dependent on algorithm iteration (7).

$$q_0 = \frac{\log_e(NC)}{\log_e(N_{max})} \quad (7)$$

N_{max} is the maximum number of iterations and NC is the iteration counter. At the beginning of the search process when q_0 values are low the bias exploration is preferred. In this way a very quick convergence of the algorithm into locally optimized solution is prevented.

The stagnation avoidance mechanism (ii) is based on comparison of random generated quantity $q \in (0,1)$ with probability $p_{ij}^k(t)$ of the selected arc. If $q < p_{ij}^k(t)$, then use the probability selection rule $\max(p_{ij}^k(t))$, if $q \geq p_{ij}^k(t)$, then choose the next node randomly. This occurs in later stages of the search process when pheromone values on the most selected arcs tend to be high and thus the chance of further exploration is low. Randomized selection at the later stages of the search process decreases the change of stagnation in local optimum.

Stützle and Hoos [6, 7] applied pheromone limit values in their MAX-MIN Ant system (MMAS). MMAS uses similar update rule to ACS [3] where only one ant is allowed to update the pheromone. But instead best-so-far solution only also iteration-best solution is used in an alternate way. Since the elitist strategy in update rule leads to stagnation in suboptimal solution, MMAS limits possible range of pheromone values to the interval $[\tau_{min}, \tau_{max}]$. Pheromone trail limits have the effect of limiting the probability values p_{ij} of selecting arc which favors exploration over exploitation and consecutive stagnation.

Furthermore, the pheromone trails are initialized to upper pheromone limit values, which, together with a small evaporation rate increases the exploration of new paths from the beginning of the search process. In the later phases when the system approaches stagnation (i.e. no improved has been generated for a certain number of consecutive iterations), pheromone values are re-initialized.

3. Pheromone Value Limits

The overall pheromone from all the ants which is accumulated on the graph is described by (2). Let suppose an ideal progress of the algorithm, i.e., from the first iteration all the ants will use the same path. The amount of pheromone accumulated on the arcs of the path is given by (1) and (2):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + m \frac{Q}{L} \quad (7)$$

Pheromone accumulation over iterations will be:

$$\begin{aligned} \tau_{ij}(0) &= \tau_0 \\ \tau_{ij}(1) &= (1-\rho)\tau_0 + m \frac{Q}{L} \\ \tau_{ij}(2) &= (1-\rho)^2 \tau_0 + (1-\rho)m \frac{Q}{L} + m \frac{Q}{L} \\ &\dots \\ \tau_{ij}(t) &= (1-\rho)^t \tau_0 + \sum_{e=0}^{t-1} (1-\rho)^e m \frac{Q}{L} \end{aligned} \quad (8)$$

The pheromone limit value for arc which is never used by any ant ($m=0$) is zero

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) = \lim_{t \rightarrow \infty} (1-\rho)^t \tau_0 = 0; \quad (1-\rho) < 1 \quad (9)$$

and the pheromone limit value for arc that is always used by all the ants is infinitive geometric series

$$\begin{aligned} \lim_{t \rightarrow \infty} \tau_{ij}(t) &= \lim_{t \rightarrow \infty} \left((1-\rho)^t \tau_0 + \sum_{e=0}^{t-1} (1-\rho)^e m \frac{Q}{L} \right) = \\ &= m \frac{Q}{L} \sum_{e=0}^{\infty} (1-\rho)^e \end{aligned} \quad (10)$$

of which convergence test is the ratio

$$\left| \frac{a_{n+1}}{a_n} \right| = \frac{(1-\rho)^1}{(1-\rho)^0} = (1-\rho) < 1 \quad (11)$$

$$\tau_{ij}(\infty) = m \frac{Q}{L\rho} \quad (12)$$

Simulation of (2) and its limit value is on Figure (1). Three phases of the search process can be distinguished: preliminary ($0 < t < 40$), saturation ($40 < t < 80$) and stagnation ($t > 80$) phase.

Preliminary phase is characteristic by rapid growth of overall pheromone value. In saturation phase pheromone evaporation takes effect. Stagnation phase represents excessive pheromone values accumulated in the most attractive part of the search area given by local or global optimum.

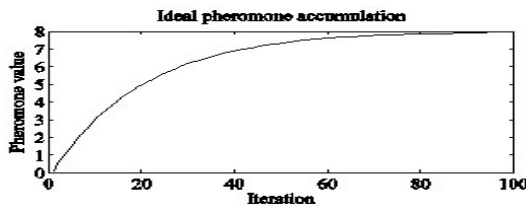


Figure 1. Simulation of pheromone accumulation (2) for $m=2$, $L=5$, $Q=1$ and $\rho=0.05$.

4. Re-Initialization

In the stagnation phase pheromone evaporation is in balance with accumulation. Due to pseudo-random proportional rule (4) not all the ants will use the same path. Majority of the ants will be attracted by the same search area with excessive pheromone values and further exploration is hardly possible. For this reason re-initialization is proposed. It is not able to estimate the pheromone limit value by using (12). However, pheromone derivation can be used instead to monitor the search process.

In this work it is proposed to re-start the search process in user adjustable point x where pheromone derivation equals pheromone value (13).

$$x\bar{\tau}(t) = \frac{d\bar{\tau}(t)}{dt} \quad (13)$$

At the re-initialization point, excessive pheromone values are decreased with the following requirements: deteriorate the difference between individual pheromone values and pheromone arithmetic mean (i), reduction will be directly proportional to its size (ii) and overall pheromone value will be reduced (iii). In this case a non-linear transformation is used (14).

$$\begin{aligned} \tau_{ij}(t) &= (s_{ij}(t)d_{ij}(t) + \bar{\tau}(t))rc \\ s_{ij}(t) &= \text{sign}(\tau_{ij}(t) - \bar{\tau}(t)) \\ d_{ij}(t) &= \log_{10} \left(1 + |\tau_{ij}(t) - \bar{\tau}(t)| \right) \end{aligned} \quad (14)$$

The overall pheromone value $\bar{\tau}$ should be decreased with respect to the information lost. The aim is to escape excessive pheromone values but preserve acquired knowledge about the problem. For that purpose reducing coefficient rc reflects the speed of pheromone accumulation (15).

$$rc = \frac{m\bar{\tau}(t)}{|A|(t-t_r)} \quad (15)$$

$|A|$ is a number of arcs of the graph, t is current iteration and t_r is iteration of the last re-initialization or zero in the beginning. Let call a section between two re-initializations a macro cycle.

The search process is divided into macro cycles. The number of macro cycles as user defined parameter is easier to estimate than number of iterations. It has been shown that low number of macro cycles is sufficient [8].

4.1. Stagnation Avoidance Mechanism

A pseudo-random proportional rule with utilization of q_0 variable parameter is used, but instead of the whole search process (7) it should be applied within each single macro

cycle.

The simple approach is to create a rank of equally distributed vales for q_0 (16), one for each macro cycle [8]. Since q_0 is compared with probability of choosing particular arc $p_{ij}^k(t)$, the maximum interval span $\langle p_{\min}, p_{\max} \rangle$ for q_0 values falls in range $\langle 0,1 \rangle$. Provided the arc with the highest probability $p_{ij}^k(t)$ should not be always chosen during the first macro cycle (probabilistic selection (3)) and never chosen during the last macro cycle (pseudo-random selection rule (4)), the interval have be reduced according to user defined values.

$$q_0 \in \left(q_{01}, q_{02}, q_{03} \dots | q_{0i+1} - q_{0i} = \Delta; \right. \\ \left. \Delta = \frac{p_{\max} - p_{\min}}{N_{mc} - 1} \right) \quad (16)$$

N_{mc} is the total number of macro cycles and N is current macro cycle.

In this article it is proposed to apply the similar approach for stagnation avoidance mechanism to original paper [5]. The parameter q_0 will vary within the range in each single macro cycle. The solution (17) is based on the difference (13) and will be close to zero at the beginning and close to one at the end of each macro cycle.

$$q_0 = 1 - \frac{d\bar{\tau}(t)/dt - x\bar{\tau}(t)}{d\bar{\tau}(t_r)/dt} \quad (17)$$

The name of the algorithm is derived from variable parameter q_0 as ACO with variable macro cycle (ACO_{VMC}).

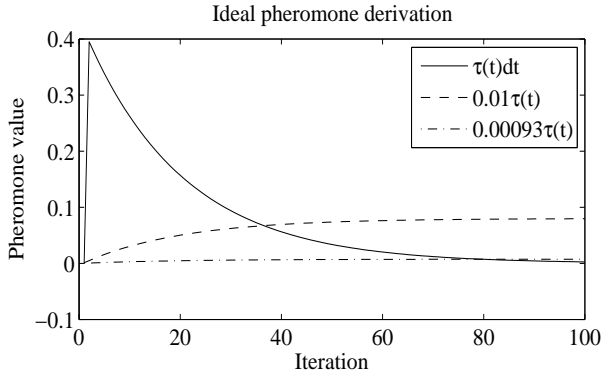


Figure 2. Ideal pheromone derivation

5. Test Case

The aim of the test case is to show the impact of (i) macro cycle length and (ii) q_0 value on algorithm performance as well as comparison ACO with re-initialization with one of the best performing ACO algorithm ACO_{KTS} [5].

Common ACO parameters for both algorithms were set in accordance with [9] and are listed in the Table 1. Other variable parameters like number of ants and number of

macro cycles are listed in the result Table 2.

Limits for variable parameter q_0 for pseudo-random proportional rule (16) were set within the interval $\langle 0.1, 0.9 \rangle$.

Variable parameter for macro cycle length x (13) was chosen to cover (i) preliminary phase and (ii) preliminary and saturation phase. The values were estimated as (i) 0.0093 and (ii) 0.00093 respectively (Figure 2).

For each setting 500 trials were performed. At first ACO_{VMC} was tested and mean value of termination cycle \bar{C}_t was determined for each setting. Then \bar{C}_t was considered as number of iterations for ACO_{KTS}.

Test graph is random generated symmetrical multi-graph with 50 nodes and 200 arcs (Figure 3). Node coordinates x, y fall in range $\langle 0,1 \rangle$ and arc values c_{ij} are equal to Euclidean distances between arc nodes (i,j) . Arc lengths c_{ij} have been considered as heuristic values η_{ij} . This setting simulates common traffic optimization problem. The task is to find the shortest path between two given nodes - the start node $n_s=2$ and the end node $n_e=31$.

For test evaluation only probability of finding the global optimum ($T = [2 \ 26 \ 47 \ 22 \ 18 \ 24 \ 31]$) was used.

Table 1. Common ACO parameters settings

Parameter name	Value
Initial pheromone value $\tau_{ij}(0)$	0.1
Weight of pheromone information α	0.5
Heuristic values η_{ij}	0.1
Weight of heuristic information β	0.1
Pheromone persistence ρ	0.05
Number of ants m	10
Number of cycles	200

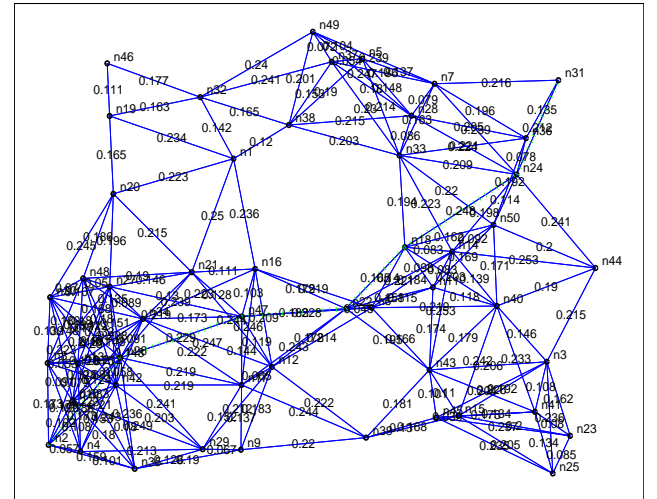


Figure 3. 50 node graph with the shortest path in green.

6. Results

The more resources (ants or macro cycles) are available the higher the probability of finding a global optimum is. Resource allocation between ants and macro cycles depends on graph complexity. The algorithm with macro

cycles benefits more from additional ants than from more macro cycles (Table 2) which similar to other ACO algorithms and complies with [9].

6.1. Macro Cycle Length

Extended macro cycle which covers saturation phase has more than twice probability of finding the global optimum in most cases compared to short macro cycle which includes only preliminary phase (Table 2, columns 3, 4). Since most of the pheromone is already accumulated at the end of the preliminary phase (Figure 1), mainly re-distribution of trailing pheromone occurs in the later stages of the search process. Pheromone concentration into particular area of the search space is caused by further utilization of acquired knowledge in stagnation phase.

Depending on parameter x (13) setting for macro cycle length, extended macro cycle can be twice longer (Figure 1). This results in twice longer overall search process. However, ACO_{VMC} results comparison four short macro cycles with two long macro cycles reveals higher performance of long macro cycles in each case.

6.2. q_0 Parameter

Results show better performance of ACO with variable macro cycles (ACO_{VMC}) over constant macro cycles (ACO_{MC}) (Table 2, columns 4, 5). This can be explained by graduate values of q_0 parameter (17) in stagnation avoidance mechanism which disturbs autocatalytic process mainly during the saturation phase. This results weaker selection pressure which favors exploration and better local optimization during saturation phase.

6.3. ACO_{VMC} and ACO_{KTS}

Comparison between ACO with variable macro cycles and ACO_{KTS} (Table 2, columns 4, 6) on sample graph reveals that results are varying and fully comparable. In general, test data shows slightly better performance of ACO_{VMC} for more resources (ants or macro cycles).

Table 2. Simulation results on 50 node graph.

ants	macro cycles	Probability of finding the global optimum [%]			
		ACO _{VMC} $x = 0.01$	ACO _{VMC} $x = 0.00093$	ACO _{MC} $x = 0.00093$	ACO _{KTS}
2	2	5.2	14.8	9.8	11.4
2	4	12.8	18.4	20.4	28.2
2	6	14.4	34.2	29.2	42.2
4	2	13.2	27.2	23.2	31.4
4	4	20.6	49.4	37.6	49.4
4	6	31.2	65.2	53.2	62.6
6	2	19.4	45.2	35.2	41.8
6	4	32.4	65.6	54.6	63
6	6	46.6	80.6	68.4	78.8

7. Conclusion

It has been proved on sample test that ACO with long variable macro cycles is fully comparable with one of the best performing ACO adaption of Kumar et al. (2003). The ACO with re-initialization benefits from (i) parameter representing number of macro cycles which is easier to estimate than number of iterations, (ii) reducing overhead caused by saturation phase of the search process and (iii) self-termination ability.

The above mentioned advantages enable implementation of heuristic algorithms for automatic optimization in manufacturing systems especially in dynamic environment where good solution has to be delivered in real time.

Acknowledgments

Thanks to Science Publishing reviewers for valuable feedback and provided comments which increased the paper quality.

References

- [1] P. E. Hart, N. J. Nilsson and B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 4(2), 1968, 100–107
- [2] M. Dorigo, V. Maniezzo and A. Colomi, The Ant System: An autocatalytic optimizing process, Technical report 91-016 revised, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991
- [3] M. Dorigo and L. M. Gambardella, Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1), 53–66,
- [4] Y. Nakamichi and T. Arita, Diversity control in ant colony optimization, In Abbas HA (ed) Proceedings of the Inaugural Workshop on Artificial Life (AL'01), Adelaide, Australia, Dec 11, 2001, 70-78
- [5] R. Kumar M. K. Tiwari and R. Shankar, Scheduling of flexible manufacturing systems: an ant colony optimization approach, proc. Instn. Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture, 2003, 1443–1453
- [6] T. Stützle and H. H. Hoos, Improving the Ant System: A detailed report on the MAX-MIN Ant System. Technical report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt, Germany, 1996
- [7] T. Stützle and H. H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems, 2000, 16(8), 889–914
- [8] M. Ciba, ACO algorithm with macro cycles, Proceedings on 14th Conference of Doctorial Students on Elitech'12, Slovak Technical University of Bratislava, May 2012
- [9] M. Becker and H. Szczerbicka, Parameters influencing the performance of ant algorithms applied to optimization of buffer size in manufacturing, IEMS Vol. 4, No. 2, December 2005, 184–191