STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **10**, February 2022, pp 282–294.
Published online in International Academic Press (www.IAPress.org)

The Ant Heuristic "Reloaded"

Djamel Zeghida [1,2,*], Nora Bounour [1], Sarra Khelifi [2]

[1] *LISCO Laboratory, Badji Mokhtar-Annaba University, Annaba, Algeria*
[2] *LICUS Laboratory, 20 Août 1955-Skikda University, Skikda, Algeria*

Abstract As a swarm-based optimization heuristic, the Ant System (AS) was proposed to deal with the Traveling Salesman Problem (TSP). Classified as a constructive approach, AS was inspired by the ants' social behavior.In Practice, its implementation reveals three basic variants that exploit two operating models: a first "natural" model, where ants move and update pheromones, with consideration of distances between towns, to define the Ant-Quantity variant, and without considering distances for the Ant-Density variant. Or an "abnormal" second model, where ants move and delay updates until all ants have completed a full cycle, then consider the tour length which defines the Ant-Cycle that was claimed to be the best variant. We reload the AS and reconsider these three basic ant algorithms and their respective two models. We propose to explore an "overlooked" third model, which consists of further expanding the "abnormal" ants' attitude, forcing them to move and update pheromones, simultaneously, after every single move, thus conceiving the Ant-Step model. Keeping the option whether or not to consider distances, we propose two new AS basic algorithms: the Ant-Step-Quantity and the Ant-Step-Density. The aim of this paper is to present and assess these two new basic AS variants in respect to the three existing ones, specially the one considered the best, through an experimental study on various symmetric TSP benchmarks.

Keywords Swarm-Based Optimization, Ant System Heuristic, Traveling Salesman Problem

AMS 2010 subject classifications 68T20, 90C27, 90C59

DOI:10.19139/soic-2310-5070-1218

1. Introduction

For a class of *Combinatorial Optimization Problems* (COP), the exact solution can be found using exact methods, such as: Linear/Dynamic Programming, Simplex Method, Backtracking, Branch and Bound and Branch and Cut. For other problems, known as NP-hard COP, an exact solution can't be reached (either because it is unreachable within a reasonable time or doesn't exist at all), in this case, we just look for an approximate solution using approximate approaches.

Without any guarantee of optimality, approximate methods use appropriate and specific exploration (diversification) and exploitation (intensification) mechanisms to explore the problem's search space. Some of them deal with a single solution and are known as *unique solution based* methods such as: Local Search, Simulated Annealing, Tabu Search, Variable Neighborhood Search, Iterated Local Search and Global Local Search. Whereas others are *population based*, dealing with an aggregate of solutions, like: Evolutionary Algorithm, Genetic Algorithm and Swarm Based Algorithms such as: Particle Swarm Optimization, Ant Colony Optimization, Cuckoo Search and Grey Wolf Optimizer.

These approaches can be clustered in two categories: *Constructive* or *Ameliorative* ones. A *constructive* approach starts from scratch and gradually constructs a solution or a set of solutions (such as: Ant Colony Optimization). An

*Correspondence to: Djamel Zeghida (Email: dj.zeghida@gmail.com). Department of Computer Science, 20 Août 1955-Skikda University. P.O Box n°26, El-Hadaeik, Skikda, Algeria (21000).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

ameliorative approach starts with a solution or a population of solutions and proposes an improved solution or a chosen one among a set of improved ones (like: Genetic Algorithm or Grey Wolf Optimizer).

Many approaches were proposed, in literature, to tackle the *Traveling Salesman Problem* (TSP), as one of the most challenging and treated NP-hard discrete COP, some methods were more efficient than others.

We can, for instance, cite the use of a Grey Wolf Optimizer [18, 29] or an Artificial Bee Colony Algorithm [22, 6]. Some, relatively, less recent examples were presented in [17, 2] using Particle Swarm Optimization [15, 32], Artificial Neural Network [5] and Evolutionary Computing Algorithms [33, 23]. A survey of old known approaches can be found in [21], such as Tabu Search [19, 24, 14] and Simulated Annealing [1, 28, 16, 26].

As a constructive population-based approach, a novel heuristic, based on the social ants' behavior, was proposed to deal with the TSP, it was called the *Ant System* (AS) or the *Ant Algorithm* (AA) heuristic [7, 10].

Without any referential mathematical model, the AS authors [7, 10] were constrained to develop different implementations of this Ant heuristic, generating three basic variants: Ant-Density, Ant-Quantity and Ant-Cycle algorithms.

Based on the Ant-Cycle variant, proclaimed the best among these three basic AS implementations, many improvements of this heuristic for TSP were proposed such as: *Ant Colony System* (ACS) [11], *MAX-MIN Ant System* (MMAS) [30] and *Rank Based Ant System* (AS_{rank}) [3] and *Elitist Ant System* (AS_{elit} or EAS) [3, 10]. Other improvements were proposed to deal with other optimization domains, such as Graph Coloring [8], Quadratic Assignment [25, 31] or Vehicle Routing [4].

However, it has been shown, recently [35, 36], that the Ant-Cycle variant can't be proclaimed as the best among the three basic variants any more, since other "overlooked" basic variants can exhibit better performance too.

The "premises" of such an idea come from the flowing "Naturalist" observation; How can a "biased" imitation (the Ant-Cycle algorithm) of the natural ant behavior perform better than the "exact" imitation of nature (the Ant-Density or Ant-Quantity algorithms)?.

According to these basic variants' authors [7, 10], the Ant-Cycle's "supremacy" was due to the use of "global information" in the system. So, if we provide "global information" in the case of Ant-Density and Ant-Quantity, for instance, it can improve their performance too.

The use of "global information" in Ant-Cycle results from the ants' attitude to synchronize the pheromones updates, all together, after an entire tour (a complete tour synchronization). In the case of Ant-Density or Ant-Quantity, the use of "global information" can be performed if the ants "choose" to "wait" for the others to update pheromones, all together, after every single move or step (a single move synchronization). Thus giving the idea to explore a third "overlooked" choice when defining the AS basic variants.

In this paper, we define two new variants of the Ant Algorithm: the Ant-Step-Density and the Ant-Step-Quantity Algorithms, where Ants are dotted with an efficient data disseminating mechanism to increase the system's reactivity.

Through Section 2, we introduce, as a background of our work, a brief overview of the application domain: defining the Traveling Salesman Problem, and presenting the principles of the three basic algorithms of the AS heuristic: Ant-Quantity, Ant-Density and Ant-Cycle, as proposed by *Dorigo et al.*.

In Section 3, we present, in our style, the two models used to define the previous three basic variants.

Through Section 4, we give some comments on the basic variants, their corresponding models and algorithms.

In Section 5, we present our contribution. We signal, first, an "overlooked" choice, when commenting the *Dorigo et al.*'s work. After that, we give the corresponding models and algorithms of the two proposed new methods: the Ant-Step-Density and the Ant-Step-Quantity.

Throughout Section 6, we assess and validate our new basic variants in respect to the previous existing ones. Section 7 concludes the paper.

2. The Traveling Salesman Problem and the Ant Heuristic

Mostly, discrete *Combinatorial Optimization Problems* (COP) belong to NP-hard class problems. For this class of problems an exact solution is out of reach, forcing the use of heuristic approaches (more details in [13]). The Traveling Salesman Problem (TSP) may be the most famous example of these problems.

This problem consists in a traveling salesman looking for the shortest way to visit *n* towns without going twice to any town, except starting and ending in the same town. When paths between any given two towns (in case of more than one path) are equal the TSP is symmetric otherwise the problem is asymmetric.

The use of ants for symmetric TSP begins with the use of a novel heuristic based on the ants' social behavior, giving rise to the *Ant System* (AS) or the *Ant Algorithm* (AA) of *Dorigo et al.*, with their three basic variants: *Ant-Density*, *Ant-Quantity* and *Ant-Cycle*.

The AS heuristic [7] was always competitive and can be considered as a landmark in solving TSP. Reference [20] presents a comparative study between *Ant Colony Optimization* (ACO), *Genetic Algorithm* (GA) and *Simulated Annealing* (SA).

Dorigo et al. [7, 11, 12] deal with a symmetric TSP with euclidean distance, and use the following notations:

- *n*: number of towns, *m*: number of ants;
- $Tabulist_k$ memorizes for Ant_k the towns already visited up to time *t*;
- d_{ij}, the euclidean distance between two towns T_i and T_j: $d_{ij} = \sqrt{(x_1^i - x_1^j)^2 + (x_2^i - x_2^j)^2}$;
- The intensity of pheromone trail on $Path_{ij}$ at *t+1*, given by equation (1):

$$\tau_{ij}(t+1) = \rho.\tau_{ij}(t) + \Delta\tau_{ij}(t,t+1) \tag{1}$$

  ρ the evaporation coefficient, at *t*=0, $\tau_{ij}(0) = Q_c$(Initial pheromone quantity);
- $\eta_{ij} = \frac{1}{d_{ij}}$ the visibility;
- The Ant_k transition probability from T_i to T_j, calculated by equation (2):

$$P_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t).\eta_{ij}^\beta}{\sum_{k \in allowed_k} \tau_{ik}^\alpha(t).\eta_{ik}^\beta} \tag{2}$$

  $allowed_k$= n - $Tabulist_k$, α and β parameters that control effect of pheromone intensity versus visibility.

3. The Exploited Ant System Models and the Three Basic Variants

In practice, and without any mathematical referential model, two choices were exploited:

- ***Model 1***: Desynchronized updates after every single move.
- ***Model 2***: Synchronized updates after an entire tour.

The first model generates two variants and the second one gives rise to one variant, as presented in the following subsections.

3.1. The First Variant: Ant-Density Algorithm

Updates are performed after every single move and distances between towns aren't considered.
Ant_k going from T_i to T_j leaves a Q_1 units of pheromone on $Path_{ij}$, between *t* and *t+1*, according to equation (3):

$$\Delta\tau_{ij}^k(t,t+1) = Q_1, \tag{3}$$

3.2. *The Second Variant: Ant-Quantity Algorithm*

Updates are performed after every single move and distances between towns are considered.
Ant_k going from T_i to T_j leaves a quantity Q_2 of pheromone depending on the length of $Path_{ij}$, between *t* and *t+1*, according to equation (4):

$$\Delta\tau_{ij}^{k}(t,t+1) = Q_2/d_{ij}, \tag{4}$$

3.3. *The Third Variant: Ant-Cycle Algorithm*

Updates are performed after an entire tour, it is an adaptation of the Ant-Quantity approach. The pheromone update is not computed at every move or step but after a complete tour (n steps/moves).
If Ant_k goes from T_i to T_j between *t* and *t+n*, it leaves on $Path_{ij}$ a quantity Q_3 by the length L_k of its performed tour, according to equation (5):

$$\Delta\tau_{ij}^{k}(t,t+1) = Q_3/L_k, \tag{5}$$

Equation (1), in this case, becomes:

$$\tau_{ij}(t+n) = \rho.\tau_{ij}(t) + \Delta\tau_{ij}(t,t+n) \tag{6}$$

The experimental study on several TSP benchmarks allows the Ant System authors to take the following decisions:

- m = n;
- Q_c=5, Q_1=Q_2=Q_3= 100 (the values do not mater);
- For Ant-Density/Quantity Algorithms, the best parameters set is: α=1, β=5, ρ=0.99; the evaporation is equal to: $1-\rho = 1-0.99 \approx 0$, (no evaporation);
- For Ant-Cycle Algorithm, the best parameters set is: α=1, β=5, ρ=0.5; evaporation rate: $1-\rho = 1-0.5 =$ 0.5, (50% evaporation);

The authors concluded that the most performing variant was the Ant-Cycle, due to the use of global information [7, 11, 12].

4. Analysis and Comments on the Exploited Ant System Models and the Three Basic Variants

Figures 1 and 2 illustrate and depict the two models used by the three AS basic variants presented in Section 3, respectively, the Ant-Density/Quantity and the Ant-Cycle variants.

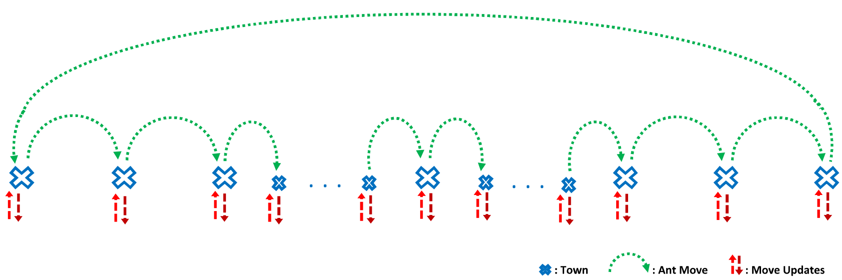

Figure 1. Illustration of the Ant-Density/Quantity model.

The obvious observation on these three variants is the fact that the Ant-Cycle variant biases, clearly, the natural ant's behavior. In this case, ants carry out the updates (pheromone's evaporation and deposit) after an entire cycle, using the realized tour's length. After that, they join the start line and wait for a given next iteration, defining, *"undeliberately"*, an *"implicit synchronization point"* (See the model Figure 2 and Algorithm 2 as a proposed algorithm for the case).

Whereas Ant-Density and Ant-Quantity reflect the ant behavior in nature, and the latter even more; considering the possible effect of distance on the pheromone evaporation in nature. In this first used model and for both options, ants carry out the updates (pheromone's evaporation and deposit) with each step, after any single move between two towns, either taking in consideration or not the distance between towns (See the model, Figure 1 and the proposed algorithm for the case, Algorithm 1).

Algorithm 1 (Ant-Density/Quantity Algorithm)

```
Initialization                                % t=0

 For k = 1 to m                               % m Ant
   Tabulist_k[1] = T_k                        % Ant_k Start Town T_k
 End For
 For t = 1 to t_max                           % t_max iterations
   Parallel                                   % Parallel treatment
     Ant Ant_1 :                              % The first Ant
       ...
       ⟨Instructions⟩
       ...
     End Ant
     ...
     ...
     Ant Ant_k :                              % The k^th Ant
       For i = 2 to n                         % Perform a tour
         Select T_j,                          % Next Town
         using equation (2)                   % Best P_ij
         Tabulist_k[i] = T_j                  % Ant moved to T_j
         Update τ_ij                          % Each ant alone, after any single move
         using equation (1) and equation (3)/(4)
       End For
     End Ant
     ...
     ...
     Ant Ant_m :                              % The last Ant
       ...
       ⟨Instructions⟩
       ...
     End Ant
   End Parallel
     Get the shortest tour among the m tours
 End For
 Get the shortest tour among the t_max tours
```

So, how can the Ant-Cycle variant as a "biased" imitation, implementing an "abnormal" ant behavior perform better than an "exact" imitation of nature realized with the Ant-Density or the Ant-Quantity variants? [35].

This "abnormal" behavior adopted by the Ant-Cycle variant, forces ants to "synchronize" the pheromone updates after a whole cycle, all together, just before starting a next iteration (See Figure 2), such behavior, paradoxically,

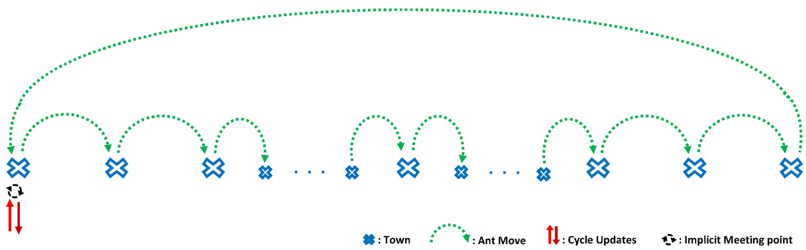

Figure 2. Illustration of the Ant-Cycle model.

increases the ants' reactivity. Effectively this second model provides, for ants, an efficient disseminating data process in the system by depositing, adequately, the needed data and increases consequently their reactivity.

Algorithm 2 (Ant-Cycle Algorithm)

Initialization % t=0

For $k = 1$ **to** m % m Ant
  $Tabulist_k[1] = T_k$ % Ant_k Start Town T_k
End For
For $t = 1$ **to** t_{max} % t_{max} iterations
  Parallel % Parallel treatment
    Ant Ant_1 : % The first Ant
      ...
      $\langle Instructions \rangle$
      ...
    End Ant
    ...
    ...
    Ant Ant_k : % The k^{th} Ant
      For $i = 2$ **to** n % Perform a tour
        Select T_j, % Next Town
        using equation (2) % Best P_{ij}
        $Tabulist_k[i] = T_j$ % Ant moved to T_j
      End For
      Synchronization
        Update τ_{ij} % All ants together, after an entire cycle
        using equation (6) and equation (5)
      End Synchronization
    End Ant
    ...
    ...
    Ant Ant_m : % The last Ant
      ...
      $\langle Instructions \rangle$
      ...
    End Ant
  End Parallel
    Get the shortest tour among the m tours
End For
Get the shortest tour among the t_{max} tours

Consequently, all improved variants of this heuristic for TSP such as: *Ant Colony System* (ACS) in [11], *MAX-MIN Ant System*; MMAS in [30], *Rank Based Ant System*; AS_{rank} in [3] and *Elitist Ant System*; AS_{elit} or EAS in [3, 10] (Table 1) gained on behalf of the Ant-Cycle model's adequacy (Figure 2).

Table 1. Ant System ameliorated variants' characteristics [36].

Variants	**Updates**	**Frequency**	**Ants behavior**
MMAS	Evaporate + Q / L_{best}	A Tour	Only Elitist Ants*
AS_{elit}	Evaporate + Q / L_k + $\sigma.Q$ / L_{best}	A Tour	Only Elitist Ants
AS_{rank}	Evaporate + $(\sigma - \mu).Q$ / L_μ + $\sigma.Q$ / L_{best}	A Tour	Only Elitist Ants

Q: Pheromone, L_{best} : Best performed tour's length,
σ: Elitist Ant's number, μ: Elitist Ant's rank,
L_k: Ant_k tour's length, L_μ : Ant_μ tour's length.
*: Under Upper/Lower pheromone bounds.

For most cases, the obtained results place the AS approach among the best available heuristics, which encouraged *Dorigo* to propose the *ACO* meta-heuristic [12].
We have to highlight here that the application of an ant approach for a given problem is constrained by the following facts and conditions [9]:

1. The problem has to be seen as a graph;
2. The emerging collective behavior is autocatalytic, it defines a positive feedback loop process: the more pheromone trail on a path, the more it is attractive and used by ants;
3. The greedy force to construct a solution and to guide the autocatalytic process: the transition probability between towns;
4. The satisfaction and stop criterion to avoid a pheromone trail combinatorial explosion: the tabu list.

5. The Overlooked Ant System Model and the Ant-Step Basic Variants

The use of "global information", within the Ant-Cycle model, results from the ants' attitude to update pheromones, all together, after an entire tour, so, if we provide "global information" in the case of the alternative Ant-Density and Ant-Quantity variants, it "can" improve their performance too.

Analogically, using "global information", in the case of Ant-Density or Ant-Quantity, can be performed if the ants "choose" to "wait" for the others to update pheromones, all together, after every single move or step, defining, *"deliberately"*, an *"explicit synchronization point"*.

So when analyzing the two choices explored by *Dorigo et al.* to define their three AS basic variants, an idea emerges to explore a third "overlooked" choice as follow:

- ***Model 3***: Synchronized updates after every single move (See Figure 3 and Algorithm 3).

It is a "combination" of the two previous models: ***Model 1*** and ***Model 2***; the updates are realized after every single move (like in ***Model 1***), all ants synchronize together their updates (such in ***Model 2***) either or not to consider the distances between towns (like offered in ***Model 1***). It generates two new basic variants, as presented in the following subsections.

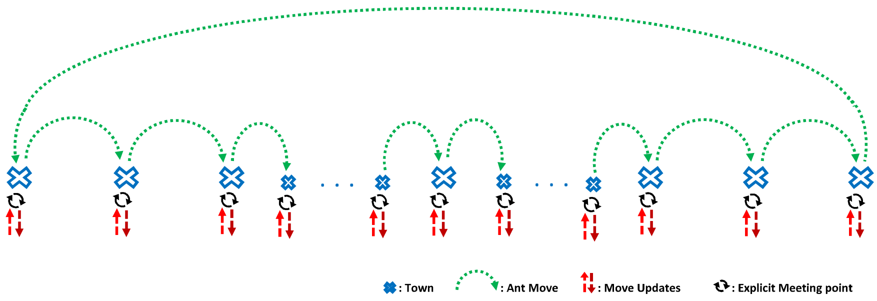

Figure 3. Illustration of the Ant-Step-Density/Quantity model.

Algorithm 3 (Ant-Step-Density/Quantity Algorithm)

```
Initialization                          % t=0

 For k = 1 to m                         % m Ant
   Tabulist_k[1] = T_k                  % Ant_k Start Town T_k
 End For
 For t = 1 to t_max                     % t_max iterations
   Parallel                             % Parallel treatment
     Ant Ant_1 :                        % The first Ant
       ...
       ⟨Instructions⟩
       ...
     End Ant
     ...
     ...
     Ant Ant_k :                        % The k^th Ant
       For i = 2 to n                   % Perform a tour
         Select T_j,                    % Next Town
         using equation (2)             % Best P_ij
         Tabulist_k[i] = T_j            % Ant moved to T_j
       Synchronization
         Update τ_ij                    % All ants together, after any single move
         using equation (7) and equation (3)/(4)
       End Synchronization
      End For
     End Ant
     ...
     ...
     Ant Ant_m :                        % The last Ant
       ...
       ⟨Instructions⟩
       ...
     End Ant
   End Parallel
     Get the shortest tour among the m tours
 End For
 Get the shortest tour among the t_max tours
```

5.1. The Fourth Variant: Ant-Step-Density Algorithm

Synchronized updates are performed after every single move and distances between towns aren't considered. All ants Ant_k going from any given T_i to T_j leave, simultaneously, a Q units of pheromone on all used $Path_{ij}$, between *t* and *t+1* according to equation (3).

5.2. The Fifth Variant: Ant-Step-Quantity Algorithm

Synchronized updates are performed after every single move and distances between towns are considered. All ants Ant_k going from any given T_i to T_j leave, simultaneously, a quantity Q of pheromone for every unit of length on all used $Path_{ij}$, between *t* and *t+1* according to equation (4).

Equation (1), for these cases, becomes:

$$\tau_{ij}(t+1) = \rho.\tau_{ij}(t) + \cup_{i=1}^{k} \Delta\tau_{ij}(t, t+1) \tag{7}$$

Table 2 recapitulates features of the three AS basic variants presented in Section 3, and our two new proposed ones (proposed in Section 5).

Table 2. Former and new Ant System basic variants' characteristics [36].

Variants	**Updates**	**Frequency**	**Ants behavior**
Ant-Density	Evaporate + Q	A Move	Individualized
Ant-Quantity	Evaporate + Q / d_{ij}	A Move	Individualized
Ant-Cycle	Evaporate + Q / L_k	A Tour	Synchronized
Ant-Step-Density	Evaporate + Q	A Move	Synchronized
Ant-Step-Quantity	Evaporate + Q / d_{ij}	A Move	Synchronized

Q: Pheromone, d_{ij}: Distance (Towns T_i, T_j) , L_k: Ant_k tour's length.

We recognize and agree that these new variants (Ant-Step-Density Algorithm and Ant-Step-Quantity Algorithm) display, even "more or less", deeper "abnormal" ant behavior, forcing the ants to "wait" every single move between two towns, to perform, at the same time, the pheromone updates, instead of an entire cycle within the Ant-Cycle variant (See the model, Figure 3 and the proposed algorithm for the case, Algorithm 3) [35]. To be more precise, considering the *frequency* of the "abnormal" ant behavior (every single move), the Ant-Step is more "abnormal", but considering the *waiting delay* (a whole complete tour), the Ant-Cycle will be, then, more "abnormal".

6. Experimental Validation

To validate our proposed new algorithms, we perform many executions for each variant implementation (100-500 iterations) on various symmetric TSP benchmarks.

First, we implement the previous three AS basic variants to produce Ant-Density Algorithm, Ant-Quantity Algorithm using Algorithm 1, and the Ant-Cycle Algorithm using Algorithm 2, namely:

- A-D A: Ant-Density Algorithm.
- A-Q A: Ant-Quantity Algorithm.
- A-C A: Ant-Cycle Algorithm.

Table 3 provides a results' synthesis of these three basic AS variants.

After that, we use Algorithm 3 to implement our two proposed variants: the Ant-Step-Density and the Ant-Step-Quantity Algorithms. These last algorithms define the "overlooked" basic AS variants, namely:

Table 3. Comparing performed tours' lengths for the Ant-Density/Quantity/Cycle Algorithms on symmetric TSP benchmarks [36].

Benchmarks	Variants		
	A-D A	***A-Q A***	***A-C A***
Ulysses16.tsp	84.43	84.43	**84.27**
Bayg29.tsp	**10810.48**	**10810.48**	**10810.48**
30.tsp	28836.83	**28518.71**	28836.83
Att48.tsp	38689.80	40931.70	**38462.74**
Eil51.tsp	510.78	489.06	**487.74**
Berlin52.tsp	8422.98	**8093.35**	**8093.35**
Eil76.tsp	619.06	646.09	**583.29**
Rd100.tsp	9449.85	9531.01	**9274.34**
Eil101.tsp	760.58	751.89	**749.59**
Lin105.tsp	15946.13	16235.62	**15441.64**

Bold Values: Best Performed Tour's Length.

- A-S-D A: Ant-Step-Density Algorithm.
- A-S-Q A: Ant-Step-Quantity Algorithm.

Table 4 provides a results' synthesis of the two new proposed variants, compared to the Ant-Cycle Algorithm.

Table 4. Comparing performed tours' lengths for the Ant-Step-Density/Quantity and the Ant-Cycle Algorithms on symmetric TSP benchmarks [36].

Benchmarks	Variants		
	A-S-D A	***A-S-Q A***	***A-C A***
Ulysses16.tsp	**84.27**	**84.27**	**84.27**
Bayg29.tsp	9621.14	**9371.47**	10810.48
30.tsp	**28193.93**	28535.05	28836.83
Att48.tsp	39528.07	38800.77	**38462.74**
Eil51.tsp	**472.57**	**472.57**	487.74
Berlin52.tsp	8172.42	**8093.35**	**8093.35**
Eil76.tsp	609.42	598.96	**583.29**
Rd100.tsp	**8517.11**	8985.40	9274.34
Eil101.tsp	758.79	**731.59**	749.59
Lin105.tsp	**15197.88**	15240.11	15441.64

Bold Values: Best Performed Tour's Length.

The first set of tests returned the following results:

1. The Ant-Cycle Algorithm performs better than the Ant-Density / Quantity Algorithms, with 7/10 cases, less performant in 1/10 cases and with equal performance in 2/10 cases (Table 3);
2. The *Ant-Step Algorithm (with its two variants: Ant-Step-Density or Ant-Step-Quantity) performs better than the Ant-Cycle Algorithm*, in 6/10 cases, less performant in 2/10 cases and with equal performance in 2/10 cases (Table 4).

These results are a consequence of exploiting global information in the AS, by using an efficient data distribution and availability process. That is a result of the simultaneous pheromone updates operated in a *"synchronous"* manner, after every ants' move. This process provides more efficient data, exactly when and where needed. Consequently, it improves the information dissemination mechanisms and increases the system's reactivity.

Another interesting outcome is the fact that the use of a synchronous ants' updates, such as with the Ant-Cycle Algorithm or with our new variants; Ant-Step-Density and Ant-Step-Quantity, works as a knowledge enhancement, and has to be moderated, neither more nor less than the ant system's need. For that, we have to evaporate the pheromone, allowing ants to forget part of their previous experiences. A balance has to be maintained between the pheromone deposit and its evaporation to advantage less-used good-path, to be chosen, and to disadvantage the used bad ones. A similar observation had been highlighted in [35], and still needs further investigations.

The pheromone evaporation in the Ant System, as defined by *Dorigo et al.* (Section 3), is controlled by the evaporation coefficient, ρ, used in equation (1).

For Ant-Density/Quantity Algorithms, the best, experimentally, chosen value for the evaporation coefficient is: ρ=0.99; the evaporation is equal to: $1 - \rho = 1 - 0.99 \approx 0$, i.e. no pheromone evaporation. For Ant-Cycle Algorithm, the best value is: ρ=0.5; evaporate: $1 - \rho = 1 - 0.5 = 0.5$, i.e. evaporate half the pheromone quantity (Section 3). For Ant-Step-Density/Quantity Algorithms, the best values for the evaporation coefficient to promote the pheromone evaporation are 0.3, 0.5 or 0.7 (Table 5).

We notice that in these three last variants, the synchronized pheromone update process provides more useful knowledge used by ants for their choice of path. In this case, ants must forget part of their previous experience through the evaporation process (Table 5).

On the other side, authors in [7] give some comments on what has been presented at the end of Section 4 about the four conditions to be verified in a given problem to apply an ant approach as follow:

1. A greedy force alone will not be able to construct a solution, but any given tour of towns;
2. The autocatalytic process alone will converge in an exponential manner to a local optimum;

The greedy force proposes the best suggestions to the autocatalytic process, guiding it to converge faster to an optimal solution. To help the greedy force to perform this important and crucial job, the pheromone trail must be controlled by evaporation and the synchronized pheromone update process. Thus avoiding misguiding the transaction probability (the greedy force), this confirms our previous statements.

Table 5. The AS variants and the evaporation coefficient [36].

Variants	ρ	**Evaporation**	**Knowledge**	**Forget**
Ant-Density	0.99	0 %	Not Enhanced	0 %
Ant-Quantity	0.99	0 %	Not Enhanced	0 %
Ant-Cycle	0.5	50 %	Enhanced	50 %
Ant-Step-Density	0.3/0.7	70/30 %	Enhanced	70/30 %
Ant-Step-Quantity	0.3/0.7	70/30 %	Enhanced	70/30 %

At the end, the main contribution of our paper is:

- Defining a new AA; the *Ant-Step Algorithm* with its two variants: *Ant-Step-Density Algorithm* and *Ant-Step-Quantity Algorithm*.

The performance shown by these new variants with respect to the previous claimed best one: Ant-Cycle variant, defines a new fact, and encourages us to reconsider the use of the previously best variant in the improvement of AA and try to use, instead, the *Ant-Step Algorithms*.

7. Conclusion

Our contribution takes place at an heuristic level and will have consequences at the metaheuristic one. Indeed, we reconsider, in this paper, the Ant System heuristic through an analysis and a study leading to discover an overlooked choice to implement the Ant Algorithms. Exploring this third choice to deal with the AS heuristic, we defined a new alternative: the *Ant-Step Algorithms*.

The *Ant-Step Model* increases the ants' reactivity using a data dissemination process, providing, in the system, an efficient information availability/distribution. This process acts as a knowledge enhancement, and has to be watched to keep a balance between attractive bad paths and less attractive good ones. This equilibrium is handled by the evaporation process, allowing ants to forget part of their previously used data to choose the next move.

It improves, also, the heuristic exploration mechanism of the problem's search space to discover a more optimal solution.

So, at the heuristic level, the main outcome of our work is the definition of new best AS basic variants: *Ant-Step-Density* or *Ant-Step-Quantity*, which we strongly recommend and argue for their use in any improvement of the ant algorithm, instead of the Ant-Cycle variant.

The perspective is to think about *"reloading" the Ant Algorithm* by considering our new model; *Ant-Step Algorithm* through its two new best variants; *Ant-Step-Density* or *Ant-Step-Quantity* as an improved *Computational Model* in any future application of Ant Algorithm for TSP or others NP-hard problems. These new applications have to be assessed in respect to the existing ones. Extremely, at this metaheuristic level, the perspective will be to give an improved version of the ACO metaheuristic.

Acknowledgement

This work was accomplished at the LISCO Laboratory and in collaboration with the LICUS Laboratory. The authors would like to thank the DGRSDT (General Directorate of Scientific Research and Technological Development) - MESRS (Ministry of Higher Education and Scientific Research), ALGERIA.

REFERENCES

1. E. Bonomi, and J. L. Lutton, *The N-city travelling salesman problem: Statistical mechanics and the metropolis algorithm*, SIAM review, vol. 26, no. 4, pp. 551–568, 1984.
2. M. R. Bonyadi, M. R. Azghadi, and H. S. Hosseini, *Population-Based Optimization Algorithms for Solving the Travelling Salesman Problem*, Travelling Salesman Problem, BoD–Books on Demand, pp. 1–34, 2008.
3. B. Bullnheimer, R. F. Hartl, and C. Strauss, *A new rank based version of the Ant System. A computational study*, SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business, 1997.
4. B. Bullnheimer, R. F. Hartl, and C. Strauss, *An improved ant System algorithm for thevehicle Routing Problem*, Annals of operations research, Springer, vol. 89, pp. 319–328, 1999.
5. S. S. Chen, and C. W. Shih, *Solving TSP by Transiently Chaotic Neural Networks*, Travelling Salesman Problem, BoD–Books on Demand, pp. 117–134, 2008.
6. S. S. Choong, L. Wong, and C. P. Lim, *An artificial bee colony algorithm with a modified choice function for the Traveling Salesman Problem*, Swarm and evolutionary computation, Elsevier, vol. 44, pp. 622–635, 2019.
7. A. Colorni, M. Dorigo, and V. Maniezzo, *Distributed optimization by ant colonies*, Proceedings of the first European conference on artificial life, vol. 142, pp. 134–142, 1991.
8. D. Costa, and A. Hertz, *Ants can colour graphs*, Journal of the operational Research Society, Palgrave Macmillan, vol. 48. no. 3, pp. 295–305, 1997.
9. M. Dorigo, V. Maniezzo, and A. Colorni, *The ant system: An autocatalytic optimizing process*, Italy: Dipartimento di Elettronica, Politecnico di Milano, pp. 91–116, 1991.
10. M. Dorigo, V. Maniezzo, and A. Colorni, *Ant system: optimization by a colony of cooperating agents*, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, IEEE, vol. 26, no. 1, pp. 29–41, 1996.
11. M. Dorigo, and L.M. Gambardella, *Ant colonies for the travelling salesman problem*, BioSystems, Elsevier, vol. 43, no. 2, pp. 73–81, 1997.
12. M. Dorigo, and G. Di Caro, *Ant colony optimization: a new meta-heuristic*, Evolutionary Computation, CEC 99. Proceedings of the 1999 Congress on, IEEE, vol. 2, pp. 1470–1477, 1999.
13. J. Dréo, A. Pétrowski, P. Siarry and E. Taillard, *Metaheuristics for hard optimization: methods and case studies*, Springer Science & Business Media, 2006.
14. C. N. Fiechter, *A parallel tabu search algorithm for large scale traveling salesman problems*, Working Paper 90/1, Département de mathématiques, École Polytechnique Fédérale de Lausanne, 1990.
15. E. FG. Goldbarg, M. C. Goldbarg, and G. R. de Souza, *Particle swarm optimization algorithm for the traveling salesman problem*, Travelling Salesman Problem, BoD–Books on Demand, pp. 75–96, 2008.
16. B. L. Golden, and C. Skiscim, *Using simulated annealing to solve routing and location problems*, Naval Research Logistics Quarterly, Wiley Online Library, vol. 33, no. 2, pp. 261–279, 1986.
17. F. Greco, *Traveling salesman problem*, BoD–Books on Demand, 2008.
18. H. Joshi, and S. Arora, *Enhanced grey wolf optimization algorithm for global optimization*, Fundamenta Informaticae, IOS Press, vol. 153, no. 3, pp. 235–264, 2017.
19. J. E. Knox, *The application of tabu search to the symmetric traveling salesman problem*, University of Colorado at Boulder, 1989.
20. N. Kumbharana, and G. M. Pandey, *A comparative study of ACO, GA and SA for solving travelling salesman problem*, International Journal of Societal Applications of Computer Science, vol. 2, no. 2, pp. 224–228, 2013.

21. G. Laporte, *The traveling salesman problem: An overview of exact and approximate algorithms*, European Journal of Operational Research, vol. 59. n. 2. North-Holland, pp. 231–247, 1992.
22. W.H. Li, W.J. Li, Y. Yang, H.Q. Liao, J.L. Li, and X.P. Zheng, *Artificial bee colony algorithm for traveling salesman problem*, Advanced materials research, Trans Tech Publ, vol. 314, pp. 2191–2196, 2011.
23. Y. H. Liu, *Solving the probabilistic travelling salesman problem based on genetic algorithm with queen selection scheme*, Traveling Salesman Problem, Federico Greco. Vienna, Austria, pp. 157–172, 2008.
24. *M. Malek, Search methods for traveling salesman problems*, Department of Electrical Engineering, The University of Texas, Austin, 1988.
25. V. Maniezzo, and A. Colorni, *The ant system applied to the quadratic assignment problem*, Knowledge and Data Engineering, IEEE Transactions on, IEEE, vol. 11, no. 5, 769–778, 1999.
26. S. Nahar, S. Sahni, and E. Shragowitz, *Simulated annealing and combinatorial optimization*, 23rd ACM/IEEE Design Automation Conference, pp. 293–299, 1986.
27. H.V.D. Parunak, *“Go to the ant”: Engineering principles from natural multi-agent systems*, Annals of Operations Research Journal, JC BALTZER AG, vol. 75, pp. 69–102, 1997.
28. Y. Rossier, M. Troyon, and Th. M. Liebling, *Probabilistic exchange algorithms and Euclidean traveling salesman problems*, Operations-Research-Spektrum, Springer, vol. 8, no. 3, pp. 151–164, 1986.
29. D. S. Sopto, S. I. Ayon, M. Akhand, and N. Siddique, *Modified Grey Wolf Optimization to Solve Traveling Salesman Problem*, International Conference on Innovation in Engineering and Technology (ICIET), IEEE, pp. 1–4, 2018.
30. T. Stützle, and H. Hoos, *Improvements on the ant-system: Introducing the MAX-MIN ant system*, Artificial Neural Nets and Genetic Algorithms, Springer, pp. 245–249, 1998.
31. T. Stützle, and H. Hoos, *The max-min ant system and local search for combinatorial optimization problems*, Meta-heuristics, Springer, pp. 313–329, 1999.
32. M. F. Tasgetiren, Y. C. Liang, Q. K. Pan, and PN. Suganthan, *A Modified Discrete Particle Swarm Optimization Algorithm for the Generalized Traveling Salesman Problem*, Travelling Salesman Problem, BoD–Books on Demand, pp. 97–116, 2008.
33. J. Wei, *Approaches to the travelling salesman problem using evolutionary computing algorithms*, Travelling Salesman Problem, BoD–Books on Demand, pp. 63–74, 2008.
34. D. Zeghida, F. Michel, and D. Meslati, *An agent-based model for biomorphic software systems*, Complex Systems, ICCS’2012, IEEE International Conference on, IEEE, pp. 1–6, 2012.
35. D. Zeghida, D. Meslati, N. Bounour, and Y. Allat, *Agent Influence/Reaction Ant System Variants: An Experimental Comparison*, International Journal of Artificial Intelligence, Ceser Publications, vol. 16, no. 2, pp. 60–77, 2018.
36. D. Zeghida, N. Bounour, and D. Meslati. *The Ant-Step Algorithms: Reloading the Ant System Heuristic and the Overlooked Basic Variants* , 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), IEEE, pp. 1–6, 2020.