Rojas-Morales, N., Montero, E., Pérez, L., Ochoa, G., & Riff, M. C. (2026). Understanding ant colony optimization search through local optima networks. Applied Soft Computing, 201(Part A), 115549. https://doi.org/10.1016/j.asoc.2026.115549

Highlights

Understanding Ant Colony Optimization Search through Local Optima Networks

Nicolás Rojas-Morales, Elizabeth Montero, Leslie Pérez, Gabriela Ochoa, María Cristina Riff

- First study analyzing Ant Colony Optimization using Local Optima Networks.
- MMAS, ILS, and RAS search process comparison using a Lin-Kernighan local search.
- LON model captures pheromone-driven transitions between solutions in ACO.
- MMAS dominant exploration behavior represented in LON networks.
- Analysis of LON network structures based on ACO pheromone scheduling.

Understanding Ant Colony Optimization Search through Local Optima Networks

Dr. Nicolás Rojas-Morales[a,*], Dr. Elizabeth Montero[a], Dr. Leslie Pérez[b], Dr. Gabriela Ochoa[c] and Dr. María Cristina Riff[a]

[a]*Departamento de Informática, Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso, 2390123, Chile*
[b]*Escuela de Ingeniería Informática,Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2950, Valparaíso, 2340025, Chile*
[c]*Computing Science, University of Stirling, Stirling, Scotland, FK9 4LA, United Kingdom*

ARTICLE INFO

Keywords:
Fitness Landscape Analysis
Ant Colony Optimization
Local Optima Networks

ABSTRACT

The behavior of Ant Colony Optimization (ACO) algorithms is based on a collective learning process defined by a pheromone mechanism. This process is difficult to understand due to the scheduling of pheromone deposition and evaporation, the influence of parameter values, the bias in solution construction, and the problem instance size. This work aims to extend Local Optima Networks (LONs) to understand the path traversed by the pheromone learning mechanism in the fitness landscape described by ACO algorithms. Our ant-based LONs incorporate a definition of network edges that expresses the persistence of the pheromone influence in the search process. Also, we study a simplified network, *Deposit LON*, that contains only nodes that deposit pheromones. We aim to analyze a population-based search algorithm's exploitation and exploration behavior through its network features. We evaluate our proposal on a $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System algorithm coupled with a Lin-Kernighan local search for solving the Traveling Salesman Problem. The comparative analysis of the networks reflects the exploration/exploitation balance of the algorithms, indicating the exploratory behavior of the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. We study networks generated under different pheromone influence persistence levels and conclude that this setting does not significantly affect the overall network structure. To complement our analysis, we present plots of the generated networks, a detailed report of their metrics, and a comparison with other algorithms's LONs. Our work contributes to providing a tool for analyzing ant-based algorithms' search performance using LONs.

1. Introduction

Ant Colony Optimization (ACO) algorithms are search strategies based on the behavior of certain ant species when collecting food [7]. Since its proposal, ACO approaches have been successfully applied to numerous computationally hard problems, and several variants have been proposed to improve different search components [6, 8]. The search behavior of ACO algorithms can be difficult to analyze as several components interact to steer the search process. ACO considers a set of artificial ants that iteratively construct solutions based on problem knowledge (heuristic information) and collective search experience (pheromone). This experience stems from the indirect communication between agents known as *stigmergy*, in which a pheromone structure serves as a learning medium for the algorithm. Understanding the solution landscapes traversed by the artificial ants is challenging because of several interacting factors: the scheduling of pheromone deposition and evaporation, the specific values of algorithm parameters, the biases derived from the solution representation or pheromone structure, and the size of the search space.

Fitness landscape analysis aims at providing insights on how an optimization algorithm's performance relates to the characteristics of the problem solved [39]. These approaches can be a useful tool for understanding why certain algorithms are better than others at solving one or more problem instances, and which algorithm characteristics are most relevant for an effective search process. A large set of studies proposing diverse approaches to represent and analyze fitness landscapes can be found in the literature, such as studies applying deception, correlation, epistasis, and dispersion measures [22, 18, 32]. In particular, Local Optima Network (LON) models can be applied to represent a

`nicolas.rojasm@usm.cl` (N. Rojas-Morales); `elizabeth.montero@usm.cl` (E. Montero); `leslie.perez@pucv.cl` (L. Pérez); `gabriela.ochoa@stir.ac.uk` (G. Ochoa); `maria.cristina.riff@gmail.com` (M.C. Riff)

`www.researchnrm.cl` (N. Rojas-Morales)

ORCID(s): 0000-0001-7662-1397 (N. Rojas-Morales); 0000-0002-1690-3875 (E. Montero); 0000-0001-5553-6150 (L. Pérez); 0000-0001-7649-5669 (G. Ochoa); 0000-0003-1676-266X (M.C. Riff)

combinatorial fitness landscape based on a network structure built from local optima obtained by an algorithm. Metrics on these networks – related to location and clustering of local optima – allow the characterization of the reachable solution space and provide insights into the difficulty of the instances. Examples of such analyses applied to local search-based algorithms are [10, 12, 11], and in machine learning approaches [37, 33, 41, 1].

In this work, we apply LON models to study ACO algorithms fitness landscape. Our main objective is to understand the search trajectories followed by ant-based algorithms, and how these are connected by the influence of the pheromone mechanism of the algorithm. For this, we define different node types of the network based on their role during the search process. In particular, we pay special attention to solutions that perform pheromone deposits as they influence the search trajectory. Moreover, we define the network edges definition by considering the ant-based search process as an iterative local search process that applies a sophisticated learning perturbation strategy. We evaluate our proposal on the well-known $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MMAS}$) [17] with Lin-Kernighan as the local search operator [16]. To analyze different search spaces, we solve Traveling Salesman Problem (TSP) instances with different features: randomly distributed cities, randomly clustered cities, and real-world problem instances of sizes from 506 to 1521 cities. Our contributions can be summarized as follows:

- An extension of the local optima network model for ant-based algorithms.
- An analysis of the LON model for the well-known ant-based algorithm $\mathcal{MMAS}$.
- An analysis of the exploration and exploitation of the search space traversed by the ant-based approach, focusing on the effect of the pheromone and heuristic information component.
- A LON model-based comparison of the $\mathcal{MMAS}$ search process with an Iterated Local Search (ILS) procedure and another ACO variant (Rank-based Ant System).

The proposed model allows to understand the behavior and relevant features of ACO algorithms, as measured by LON models metrics. Specifically, the model provides insights such as the proportion of solutions that bias the search, the length of ant paths from initial to global optima, the impact of pheromone-depositing ants on solution quality in subsequent iterations, and the relationship between instance features and algorithm behavior. This study also compares the LON model obtained from $\mathcal{MMAS}$ with the models generated from (a) an Iterated Local Search approach, (b) an alternative $\mathcal{MMAS}$ parameter configuration, and (c) a Rank-based Ant System approach, to illustrate the pheromone-driven effect on this problem. Notably, this article is the first to employ LONs to enhance the understanding of ACO algorithms, and it provides guidelines and identifies limitations for extending this initial LON-ACO model.

The article's structure is as follows: Section 2 presents background summary about fitness landscape analysis, local optima networks, and ant-based algorithms. Section 3 presents our proposed extension of local optima networks for ant-based algorithms, and section 4 shows the details of our proposal's evaluation. Section 5 presents the obtained results, plots of the networks, and a comparison with other algorithms. Section 6 presents a set of guidelines for extending our proposed LON model to diverse variants of ACO algorithms, along with a brief analysis of the LON results with classical landscape analysis approaches. Section 7 presents the conclusions and some possible paths for future work.

2. Background and Related Work

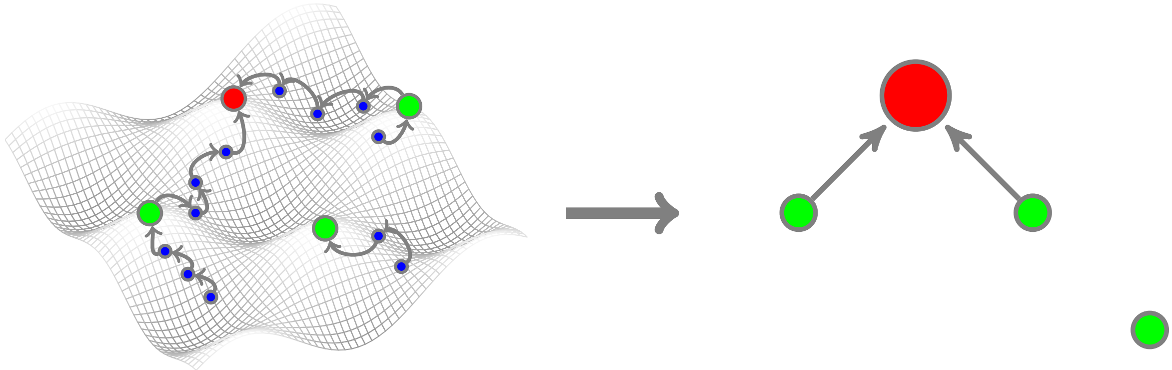
This section presents preliminary definitions of LON models and a brief literature review of existing LON approaches (Section 2.1). It also provides a general definition of ACO algorithms (Section 2.2), describes the two ACO variants used in this work ($\mathcal{MMAS}$ and Rank-based Ant System), the definition of ILS (Section 2.3), and the local search operator used in our experiments (Section 2.4).

2.1. Local Optima Networks

Fitness landscapes are characterizations of the relationship between the solutions defined by the formulation of an optimization problem and the neighborhood structure used in a search process. The fitness landscape of an optimization problem $\mathcal{P}$ requires the definition of three main components: a search space ($\mathcal{S}$), a neighborhood structure ($\mathcal{N}$), and a fitness function ($\mathcal{F}$). The search space ($\mathcal{S}$) contains all the potential solutions for $\mathcal{P}$, the fitness function assigns a numerical value to each solution in the search space, and the neighborhood structure defines a relationship between the solutions.

An *attraction basin* b_i can be defined as an area of the search space in which a set of solutions ($s \in b_i$) are grouped around a local optimum (lo_i), where is possible to obtain lo_i by applying a local search procedure over each of the solutions in b_i such that $b_i = \{s \in S \mid local\_search(s) = lo_i\}$. Note that in a search space, there might be multiple basins of attraction and their distribution may not be uniformly connected in the space. The existence of a transition in the search (edge) between two nodes (local optima) implies the connection between their basins of attraction.

Considering the huge number of solutions that a search space can contain, LONs were proposed in order to obtain a compressed representation of a fitness landscape [26]. A LON can be defined as a directed graph $G = (N, E)$ where the set of nodes N are all the local optima solutions in the fitness landscape and E represents possible transitions between local optima solutions. Figure 1 presents an example of this representation, where in (a) sequences of candidate solutions are visited by an algorithm (blue nodes). These sequences yield local optima solutions (green nodes) or a global optima solution (red node). In a LONs (Figure 1(b)), the graph represents a compressed representation of the search space, considering only the local optima solutions (green nodes). Edges in the network describe the trajectory of the algorithm visiting local and global optima solutions.



(a) Fitness Landscape (b) Local Optima Network

Figure 1: *Fitness landscape* to LON. Small blue nodes represent the solutions that are not local optima solutions. Green nodes represent local optima, and the red node represents the global optimum of a maximization problem (both considered in LONs). In (b) only local optima solutions are represented. The size of nodes is proportional to the number of visits while the lines represent transitions between local optima during the search described in (a).

Local Optima Networks can be used as a tool to understand the characteristics of a fitness landscape since different features can be studied from these networks [28]. Some of these features are the degree distribution, the clustering coefficient, and the distance between nodes, among others [23]. Furthermore, the structure of LONs can be analyzed to describe features related to the difficulty of the associated problem instance. For example, in [14, 13], an analysis of the *PageRank centrality* of LONs and the performance of local search-based metaheuristics is presented.

In general, the definition of the edges varies according to the type of LON to be constructed. These edges can be directed or non-directed or have weights associated with their frequency of use. Examples of LON edges in the literature are the following:

- In [26], an exhaustive enumeration process was used to obtain the fitness landscape data. Based on this information, two nodes lo_i and lo_j are connected by a non-directed edge if there exist two solutions $s_i \in b_i$ and $s_j \in b_j$ that can be obtained by applying a neighborhood operator (bit flip) once. This means, solutions s_i and s_j are neighbors such that $s_j \in \mathcal{N}_{bf}(s_i)$ and $s_i \in \mathcal{N}_{bf}(s_j)$, where $\mathcal{N}_{bf}$ is the bit flip operator neighborhood. Each edge has an associated weight that corresponds to the *transition probability* between two basins of attraction (b_i and b_j). This probability is calculated based on the number of connected solutions between the basins. The higher the number of connections between two basins, the higher their transition probability.

- In [38], the fitness landscape data is obtained by sampling an iterated local search procedure that uses a bit flip operator as a perturbation (1 or 2 applications) and a hill-climbing procedure as a local search. A directed edge exists between two nodes lo_i and lo_j if there exists a solution s whose distance to one of the local optimum is at

Algorithm 1 $\mathcal{A}$: ACO algorithm general structure

Input: α, β, ρ, τ_0, m

1: $\tau \leftarrow$ Initialize Pheromone Trails (τ_0);
2: $\eta \leftarrow$ Generate Heuristic Information ();
3: **while** termination criterion not reached **do**
4:     **for** $k = 1 \rightarrow m$ **do**
5:         $S_k \leftarrow$ Build Solution (τ, η, α, β);
6:         $S_k \leftarrow$ Local Search (S_k);
7:         $S_{Best} \leftarrow$ Update Best Solution (S_k, S_{Best});
8:     **end for**
9:     $\tau \leftarrow$ Update Pheromone (S, τ, ρ);
10: **end while**
11: **return** $\mathbf{S_{Best}}$

most D and s belongs to the basin of attraction of the other local optimum lo_j:

$$d(lo_i, s) \leq D \wedge s \in b_j,$$

where distance function $d(s_x, s_y)$ is defined based on the number of applications of the neighborhood operator (bit flip) required to obtain solution s_y from solution s_x.

- In [24], the fitness landscape data is obtained by sampling an evolutionary algorithm based on the Generalized Asymmetric Partition Crossover for TSP. A directed edge exists between two nodes lo_i and lo_j if lo_j can be obtained by applying the crossover or mutation operator and then applying a 3-opt local search procedure.

To visualize more specific information, Monotonic Local Optima Networks (MLON) and Compressed Monotonic Local Optima Networks (CMLON) were proposed in [27]. An MLON is a network structure whose nodes represent local optima and considers only non-deteriorating transitions between local optima solutions (*monotonic* edges). On the other hand, a CMLON is a network where each node represents a compression of all connected local optima solutions that share the same fitness value. CMLONs allow the identification of local optima plateaus in a landscape. This *neutrality* feature could also be related to the difficulty of problems in which classical local search strategies get stuck. LON models are designed for the analysis of algorithms that handle locally optimal solutions. Consequently, these models are not applicable when identifying locally optimal solutions is feasible or effective for the search process. Search Trajectory Networks (STN) is a model proposed with similar objectives to those of LONs [25]. In contrast to LONs, STNs do not restrict nodes to locally optimal solutions; instead, all visited solutions are stored across multiple independent executions. Given the large number of solutions often visited by algorithms, STNs require the definition of subsets of visited solutions called locations. Different location sizes can lead to different space representations and yield different conclusions about space analysis. We note that LON and STN networks may provide distinct perspectives on a given algorithm.

To the best of our knowledge, this is the first proposal to analyze and understand the search behavior of ACO algorithms using LON models. Due to the complexity of the pheromone learning mechanism in ACO algorithms, new definitions are needed to extend LON models so they can represent ACO search behavior. As we mentioned, some approaches have been proposed to study the search and evolutionary process of other population-based algorithms. We choose to extend LON models for this proposal since, when possible, it is common to include a local search component when applying ACO algorithms. Hence, in this cases we consider that the LON model is better suited to our objective than the STN model.

2.2. Ant Colony Optimization

ACO is a population-based algorithm that represents optimization problems as graphs, in which artificial ants perform a walk to build a solution. Algorithm 1 gives a general overview of the structure of ACO algorithms. Let us consider an algorithm $\mathcal{A}$ for solving the optimization problem $\mathcal{P}$. The algorithm defines a colony of m artificial ants and parameters α, β, ρ, and τ_0. The pheromone is initialized by parameter τ_0 (line 1). The role of the pheromone in the algorithm is to allow the ants to communicate information about solution components that, when selected, have led to high-quality solutions. This component allows the algorithm to focus the search process on promising areas of the search space. The heuristic information (η) is generated in line 2. This information is often defined by a function that estimates the local cost or benefit of including a solution component in a solution. Consequently, the role of the heuristic information is to bias the construction process towards decisions that can be locally interesting. Pheromone and heuristic information are used to build new solutions (S_k) in line 5. Both τ and η are combined in a probabilistic decision rule which defines the probability that an ant selects each solution component. Equation 1 presents a traditional formula applied to calculate transition probabilities in ACO algorithms:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha} * (\eta_{ij})^{\beta}}{\sum\limits_{(r,l)\in E} (\tau_{rl})^{\alpha} * (\eta_{rl})^{\beta}} \tag{1}$$

where p_{ij} is the probability of selecting the solution component (i, j), τ_{ij} and η_{ij} are the pheromone and heuristic information associated with component (i, j), E is the set of candidate solution components, and α and β are parameters that control the influence of the pheromone and heuristic information respectively. Once a new solution is constructed using this rule, a local search procedure may be applied (when available) to the constructed solution to obtain a locally optimal solution (line 6).

The pheromone is updated in each iteration (line 9). First, pheromone levels are reduced based on the evaporation rate parameter (ρ) and then, one or more ants deposit pheromone increasing the pheromone levels associated with the components selected in their current solutions. Usually, the pheromone is updated by applying the following rule:

$$\tau_{ij}^{new} = (1 - \rho) * \tau_{ij}^{old} + \Delta\tau_{ij}^{(a)}, \tag{2}$$

where τ_{ij}^{new} is the new amount of pheromone associated to the solution components (i, j), τ_{ij}^{old} is the current amount of pheromone in (i, j), and $\Delta\tau_{ij}^{(a)}$ is the increment of the pheromone in (i, j), often proportional to the quality of the solution generated by the ant a that uses the component (i, j).

The first ACO algorithm proposed in the literature was Ant System (AS) [5] and later, several other ACO variants such as $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MMAS}$) and Rank-based Ant System (RAS) have been proposed. Unlike the traditional AS algorithm, RAS modifies the pheromone update process by allowing the pheromone deposit of only the top-ranked ants in each iteration. The deposit amount is proportional to its rank, allowing better solutions to have a stronger influence in the search process. $\mathcal{MMAS}$ introduces pheromone limits (τ_{min} and τ_{max}) to bound the pheromone values in order to avoid the stagnation induced by the traditional update rule, maintaining the exploration during the whole search process. Additionally, $\mathcal{MMAS}$ approaches commonly include a restart mechanism as an exploration strategy. To assess the evaluation of our LON proposal, we use a well-known existing framework of ACO algorithms, named ACOTSP [17], which implements different ACO variants designed for solving the TSP. In this work, we analyze the LONs generated by $\mathcal{MMAS}$ and RAS thus, their implementations are further described in the following.

2.2.1. Rank-based Ant System

RAS [3] is an extension of the original Ant System that improves the pheromone update process by introducing a ranking-based deposit mechanism. RAS sorts the ants according to solution quality and then restricts updates to only the top w ants, plus the best-so-far ant. Each of these selected ants contributes an amount of pheromone proportional to its rank, so higher-quality solutions have a stronger influence on the search process, while poorer solutions have little effect.

$$\tau_{ij}^{new} = (1 - \rho) * \tau_{ij}^{old} + \sum_{r=1}^{w} (w - r) * \Delta\tau_{ij}^{(r)} + w * \Delta\tau_{ij}^{\text{best}} \tag{3}$$

In each iteration ants construct solutions using the transition rule described in Equation 1. Constructed solutions and then improved by a local search procedure to obtain locally optimal solutions. In this work, we apply the well-known Lin-Kernighan [16] heuristic as a local search procedure in RAS, which is described in Section 2.4.

Algorithm 2 $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

Input: m, α, β, ρ

1: τ_{max}, τ_{min} ← Generate Limits ($F(S^{Near})$);
2: τ ← Initialize Pheromone Trails (τ_{max});
3: η ← Generate Heuristic Information ();
4: **while** termination criterion not reached **do**
5:     **for** $k = 1 \rightarrow m$ **do**
6:         S_k ← Build Solution (τ, η, α, β);
7:         S_k ← Local Search (S_k);
8:         S_{Best} ← Update Best Solution (S_k, S_{Best});
9:     **end for**
10:     τ_{max}, τ_{min} ← Generate Limits ($F(S_{Best})$);
11:     **if** Stagnation **then**
12:         τ ← Re-Initialize Pheromone Trails (τ_{max});
13:     **else**
14:         τ ← Update Pheromone (S_{Best}, τ, ρ, τ_{min}, τ_{max});
15:     **end if**
16: **end while**
17: **return** $\mathbf{S_{Best}}$

2.2.2. $\mathcal{MAX}$–$\mathcal{MIN}$ *Ant System*

$\mathcal{MMAS}$ [35] is one of the most popular and successful variants among ACO algorithms [31, 2, 36, 4, 40, 34]. The general outline of the $\mathcal{MMAS}$ algorithm studied in this work is presented in Algorithm 2. As already mentioned, $\mathcal{MMAS}$ introduces pheromone limits to bound the pheromone values during the search. The pheromone limits are calculated based on the tour length of the solution generated by the nearest neighbor heuristic (line 1). The pheromone limits are updated each iteration based on the best-so-far ant tour length such that $\tau_{max} = 1/\rho \cdot F(S_{Best})$ and $\tau_{min} = \tau_{max}/2n$ (line 10), where n is the number of cities, $F(S_{Best})$ is the fitness of the best-so-far solution, and ρ is the pheromone evaporation parameter. The pheromone is initialized with the τ_{max} value to benefit the initial exploration of the search space (line 2), and the heuristic information is defined as $\eta_{ij} = 1/d(i,j)$, where $d(i,j)$ is the distance between cities i and j (line 3). In each iteration, m ants build new solutions based on the probabilistic decision rule described in Equation 1. Then, a locally optimal solution is obtained by applying a local search procedure to the constructed solutions (see Section2.4 for details). When the algorithm shows signs of stagnation, the pheromone values are restarted to the initial τ_{max} value (line 12). A pheromone restart is triggered when: (1) no improving solution is found in a specified number of iterations and, (2) the branching factor (λ) indicates that the pheromone matrix will not allow a minimum level of exploration.[1] Finally, the pheromone update (line 14) follows Equation 2 performing evaporation and deposit at the end of each iteration. The pheromone deposit is performed following the update schedule described in [35], where the iteration best ant deposits pheromone every t iterations and otherwise, the best-so-far or restart-best ant performs the deposit.

2.3. Iterated Local Search

ILS algorithms are stochastic techniques that iteratively apply a local search operator, followed by a perturbation operator, repeating the intensification process to explore different areas of the search space. The general structure of an ILS approach is shown in algorithm 3. The algorithm starts by randomly generating an initial solution and applying a local search operator to obtain a local optimum (lines 1-2). Every iteration, a perturbation operator is applied over the current solution (line 4), where the p_{size} parameter controls the size of the perturbation. The local search operator is then applied to the perturbed solution (s_{pert}) to obtain a new optimal solution. Note that the selection of the perturbation together with the local search operator is important in these techniques. If the perturbation is not large enough with respect to the local search operator, the ILS will not be able to escape the current local search area and no other local optima will be found.

[1] Considering the default values of ACOTSP, 250 iterations of stagnation and branching factor $\lambda = 0.05$

Algorithm 3 Iterated Local Search

Initialize s_{start} randomly
$s_{curr} \leftarrow$ Local search (s_{start})
while Stop criterion is not met **do**
  $s_{pert} \leftarrow$ Perturbation (s_{curr}, p_{size})
  $s_{ls} \leftarrow$ Local search (s_{pert})
  if $f(s_{ls})$ is better than $f(s_{curr})$ **then**
    $s_{curr} \leftarrow s_{ls}$
  else
    $stuck \leftarrow stuck + 1$
  end if
end while

2.4. Lin-Kernighan local search

The Lin-Kernighan (LK) heuristic [16] was selected as the local search operator for the $\mathcal{MMAS}$, RAS, and ILS implementations.

In contrast to the original LK heuristic, Helsgaun's implementation supports both sequential and non-sequential exchanges for any $2 \leq k \leq n$. Sequential 5-opt moves serve as the primary search component. The implementation applies several mechanisms to enhance computational efficiency and solution quality, including candidate-set pruning, backbone-guided search, partitioning strategies, partial transcription, and dynamic candidate reordering based on previously identified high-quality tours.

3. Local Optima Networks for ACO

The search process of ant-based algorithms is guided by the pheromone and the heuristic information, as together they define the probabilistic rule applied to select solution components. Our objective is to model the network that represents the fitness landscape defined by the search process of an ant-based algorithm $\mathcal{A}$ based on the pheromone information. In trajectory-based algorithms, trajectories are defined as sequences of solutions visited during the search process. In population-based approaches it is not clear to which extent solutions influence the generation of new ones and thus, it is not straightforward to define an overall trajectory along the search space.

In this section, we present our proposal to extend the Local Optima Network model for ant-based algorithms. Section 3.1 presents the definition of nodes; Section 3.2 presents the definition of edges in the LON for ACO algorithms; and Section 3.3 details the LON's structure and introduces a new type of LON specifically designed for ACO algorithms. In our proposal, we use the pheromone component of ant-based algorithms to define their trajectory through the search space. We pay special attention to solutions that contribute to the search by depositing pheromone and the effects of algorithmic components that may influence search trajectories, such as the deposit and the evaporation schedules. We designed an extension of the existing LON model that considers the pheromone deposit of the iteration's best-quality solutions, and the evaporation across all arcs of the pheromone matrix.

The LONs are obtained through a sampling process that involves several executions of the target algorithm. Thus, the network nodes are obtained from aggregating the pheromone-based trajectories from different independent runs. All the details of the experimental setup for evaluating our proposal are presented in Section 4.

3.1. LON Nodes

We consider the use of a local search procedure that is applied to each generated solution. Consequently, all solutions obtained at the end of each iteration are locally optimal as shown in line 7 of algorithm 2. We consider the pheromone as the responsible for the ants trajectory's definition. We define two types of nodes according to their contribution to the construction of solutions on the next iterations, i.e., related to whether they deposit pheromone or not. Regarding $\mathcal{MMAS}$ explained in section 2.2.2, we define S^i as the set of solutions constructed on iteration i. As we can classify the solutions considering their quality value, the best solution is $S^i_{best} = \min_{k \in 1:m}\{F(S^i_k)\}$. We differentiate the nodes in our network according to whether they directly participate in the pheromone updating (the set of S_{best} solutions reached in an independent execution), or they just appear during the search process as an exploration effort of the search. We define:

- **Deposit** Nodes (S_{dep}): local optima solutions that contribute depositing pheromone during the search process.
- **Exploration** Nodes (S_{exp}): local optima solutions that were generated during the search process and do not deposit pheromone.

After an independent execution, S_{exp} groups the following local optima solutions:

$$S_{exp} = S_{exp} \cup (S^i \setminus \{S^i_{best}\}) \tag{4}$$

and consequently, S_{dep} is the union of all S^i_{best} reached in each iteration i. Then, to perform the sampling process, LON models aggregate the S_{dep} and S_{exp} of a number of independent executions. The aim of the definition of these node types is to study the effect of the pheromone component on the algorithm's search behaviour and the exploration capacities of the population-based nature of the algorithm.

3.2. LON Edges

During the construction process, the decision of which solution component to select next is based on its pheromone values, its heuristic knowledge, and the values of the other available solution components. Thus, differences can exist among the available solution components regarding their attractiveness for inclusion. After the construction performed by all the ants, the pheromone is updated according to Equation 2. Here, their main components are: (1) the deposit of pheromone, and (2) the evaporation of pheromone. In each iteration of $\mathcal{MMAS}$, one solution updates the pheromone matrix, increasing the pheromone concentration ($\Delta\tau$) in its solution components, where the shorter the path, the more pheromone is deposited by an ant. Moreover, evaporation is controlled by the parameter ρ that is in charge of equally decreasing all the pheromone trails with exponential speed [19]. Due to the multiple variables and factors that simultaneously affect the construction and updating processes, the influence of pheromones is difficult to represent and generalize. As a consequence, we define the connection between pairs of nodes solely based on pheromone deposits.

We define edges as the connection between two local optima lo_i and lo_j for which lo_i indirectly contributed to generating lo_j in a later iteration. Consequently, we create an edge from each local optimum that deposited pheromone to the solutions generated at some point after this deposit. Note that these edges are directed, as we are interested in the analysis of the effect of the pheromone deposit on the search. Moreover, as edges start at deposit nodes, the set of locally optimal solutions that were only an exploratory attempt during the search process is segregated. Also, we can measure the influence of a deposit node on the algorithm's trajectory by considering its number of outgoing edges, which represent the number of influenced new nodes (exploration or deposit).

Analyzing the influence of a pheromone deposit on the solution construction process should consider both pheromone persistence and pheromone distribution across the solution components. Note that the persistence of a pheromone deposit depends on the deposit amount and the evaporation process. We approximate the number of iterations required to reduce the influence of a pheromone deposit to a defined threshold by estimating the number of iterations in which the selection probability of a solution component is reduced to a defined threshold. Assume that a pheromone deposit of $\Delta\tau$ is performed on solution component A at iteration 1, the amount of pheromone remaining on this solution component after N iterations can be expressed as $\tau^A = \Delta\tau(1-\rho)^N$. Note that this expression is isolating the effect of the deposit by assuming that only evaporation is applied after it. Also, assume that $\Delta\tau$ pheromone deposits are performed on a different solution component B when no deposit was performed on component A. The amount of pheromone deposited on component B after N iterations can be expressed as: $\tau^B = \Delta\tau \sum_0^{N-1} (1-\rho)^i$. The probability of selecting the solution component A can be expressed as:

$$p^A = \frac{\tau^A}{\tau^A + \tau^B} = \frac{\Delta\tau(1-\rho)^N}{\Delta\tau \sum_0^N (1-\rho)^i} = \frac{\rho(1-\rho)^N}{1-(1-\rho)^{N+1}}. \tag{5}$$

We can then approximate the number of iterations N that are required so the probability of selecting the component is s as following:

$$N \approx log\left(\frac{s}{s+\rho(1-s)}\right) \,/\, log(1-\rho). \tag{6}$$

Considering a value of $\rho = 0.02$ and a probability target of $s = 0.2$ (20%), the number of iterations required is $N = 3.81 \approx 4$, while for $s = 0.15$ the number of iterations is $N = 5.31 \approx 6$. We remark that this analysis is based on a single solution component with a constant deposit amount, whereas in constructing solutions, several such choices must be made, and the resulting probabilities will depend heavily on the fitness landscape.

To express the pheromone influence and analyze different pheromone persistence effect scenarios, in this work, we define a number p of iterations since a local optimum deposited pheromone. The p parameter then corresponds to the maximum number of iterations a local optimum can be connected to another one after a deposit. The idea is to analyze, for the same set of deposit and exploration nodes, what changes in the network if we consider different persistence values. This decision allows us to analyze the landscape's complexity and how pheromone information enables ants to reach global optima.

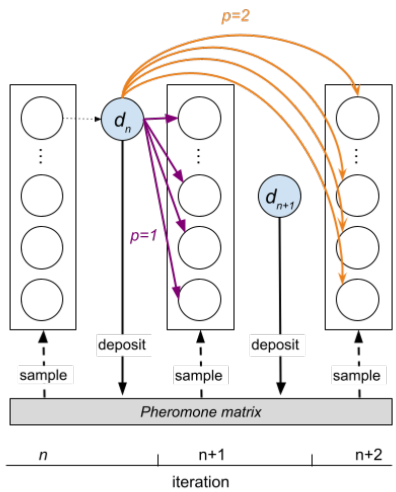


Figure 2: Node and edge definitions based on pheromone effect for ant-based algorithms. Blue nodes are deposit nodes and white nodes are exploration nodes. The pheromone persistence p denotes the maximum number of iterations the deposit node influence the generation of new solutions. Purple lines represent the edges related to pheromone persistence considering $p = 1$ and orange lines represent the edges associated to $p = 2$.

Figure 2 shows an example of how edges are defined. In iteration number n, pheromone is deposited by a selected solution d_n (blue color node). The solution d_n could be one of the solutions generated in the current iteration or a previously generated solution selected by the algorithm. Considering a pheromone persistence of $p = 1$ iterations, an edge exists between d_n and the local optima solutions constructed in the next iteration (exploration nodes, shown in white, generated in iteration $n+1$). If pheromone persistence is $p = 2$, d_n will also be connected to the exploration nodes generated in iteration number $n + 2$. Thus, the connectivity of the network will reflect how the pheromone information influences the search trajectory of $\mathcal{A}$.

3.3. LON structure

Figure 3 shows an example of a local optima network with $p = 1$, that represents the solutions connectivity during an execution of an ant-based algorithm. Numbers in nodes represent the iteration the solution they represent deposit pheromone. For simplicity, in this example only one solution deposits pheromone at each iteration. It is interesting to notice that the solution labeled "2; 4" deposits two times during the process (in the second and fourth iterations). This can be due to the use of special criteria in ant-based algorithms to determine which solution deposits pheromone. In our proposal, to simplify our analysis, we do not distinguish between different types of deposit.

Exploration nodes are shown in white in figure 3. We consider these nodes as representative of regions of the search space that were not further exploited by the algorithm. At each iteration, a set of solutions compete for being selected to bias the search process by depositing pheromone. In most ant-based algorithms, this selection is made according to the quality of the solutions. Exploration nodes can be used to quantify the exploration potential of a given search

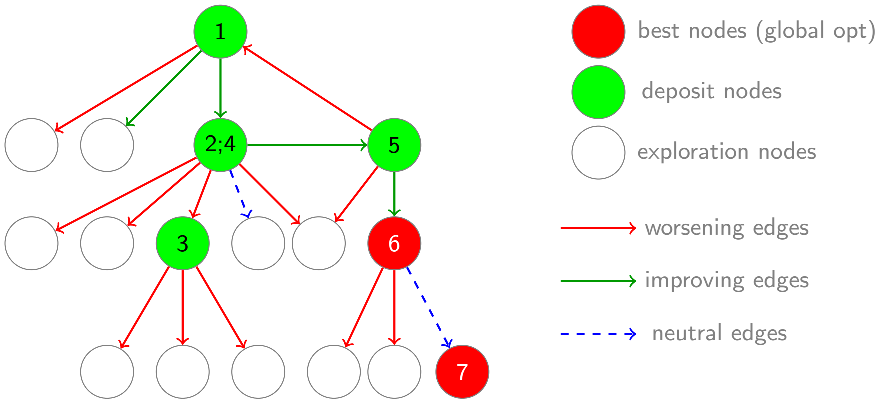


Figure 3: Example of a LON considering an execution of an ant-based algorithm.

process on a particular problem instance. It is important to notice that an exploration node at a given iteration can be selected as a deposit node later (and viceversa).

Depositing nodes are local optima that have deposited pheromone in any step of the search process. We consider these nodes as part of pheromone-based search trajectories in our own definition of LON for ACO. Moreover, we consider these nodes related to the exploitation of the process because they represent the way the algorithm decides where to search. In figure 3 the depositing nodes are shown in green color. The red nodes are the best quality depositing local optima. Edges connect deposit nodes to solutions generated in the following iteration. These edges represent the influence of the pheromone information on the search trajectories.

Excepting node 1 in figure 3, all nodes have at least one incoming edge. This node is the starting point of a search trajectory. Note that a reduced number of nodes have outgoing edges, these edges connect current depositing solutions with all the solutions generated on the next iteration. It is interesting to notice that there are also depositing nodes that do not lead to new depositing nodes. For example, node 3 deposits pheromone but none of the solutions generated at the next iteration does. This can be due to a change in the depositing schedule and pheromone restarts, defined by the ant-based algorithm. In some cases, in order to diversify the search process, the ant-based algorithm can determine that another solution deposits pheromone (the best so far for example).

Deposit LON: In each iteration of an ACO algorithm, at least one artificial ant will deposit pheromone in the pheromone matrix. In $\mathcal{MMAS}$, the best-quality solution of the iteration deposits pheromone. As a consequence, this reinforcement biases the ants' search process and determines the pheromone transmission path along the algorithm's trajectory. To focus our analysis on these solutions, we propose the Deposit LON (DLON) to analyze pheromone-based trajectories and to focus on the algorithm's exploitation capabilities. Regularly, the number of nodes in a LON makes plotting difficult and increases the cost of computing some metrics. We aim to complement the LON analysis and improve visualization by excluding exploration nodes, yielding compressed, easy-to-analyze network plots. To build a DLON, after the LON construction and metrics analysis, we filter the existing nodes, retaining only those that deposited pheromone in at least one iteration. Moreover, the size of the deposit nodes is proportional to the number of times they were used to deposit pheromone in the search process. Note that this size will be proportional to the number of nodes (both exploration and deposit) connected to the depositing node. This makes DLONs a compressed representation of the full LONs, retaining most of the information they encompass. The idea is to maintain a visualization of the search process's history, showing the influence of the filtered nodes on the ants' trajectory.

Table 1
TSP problem instances

Instance	Edge Type	Description	Optimum
Random uniform instances			
E506.25	EUC_2D	Euclidean distance	16,313,719
E755.73	EUC_2D	Euclidean distance	20,158,565
E1010.37	EUC_2D	Euclidean distance	22,904,325
E1243.85	EUC_2D	Euclidean distance	25,227,141
E1521.33	EUC_2D	Euclidean distance	28,027,563
Random clustered instances			
C506.25	EUC_2D	Euclidean distance	6,816,950
C755.73	EUC_2D	Euclidean distance	9,867,050
C1010.37	EUC_2D	Euclidean distance	10,716,003
C1243.85	EUC_2D	Euclidean distance	12,943,477
C1521.33	EUC_2D	Euclidean distance	13,608,402
Somewhat structured instances (city problems)			
att532	ATT	Pseudo-Euclidian distance	27,686
gr666	GEO	Geographical distance	294,358
pr1002	EUC_2D	Euclidean distance	259,045
rl1304	EUC_2D	Euclidean distance	252,948
nrw1379	EUC_2D	Euclidean distance	56,638
Highly structured instances (drilling problems)			
u574	EUC_2D	Euclidean distance	36,905
rat783	EUC_2D	Euclidean distance	8,806
d1291	EUC_2D	Euclidean distance	50,801

4. Experimental Setup

This section presents the experimental setup scenario used to evaluate our LON model proposal. We provide details about the algorithms settings, the sampling process (Section 4.1) and the LON metrics considered in our experiments (Section 4.2).

We conducted experiments on 18 TSP benchmark instances. Table 1 shows the 18 problem instances used in our experiments and summarizes their main features. These instances can be classified into random uniform, random clustered, city, and drilling problems.

4.1. Sampling process

4.2. LON Metrics

To evaluate the complexity of search spaces we measure a set of indicators for each local optima network. We aim to correlate these measurements with the quality of solutions found by the algorithms in these spaces. These metrics are described below:

- #optima: Number of nodes in LON network. It indicates the total number of optima, including global and local.
- #glob: Number of different global optima.
- #g_e_impr: Number of incoming improving edges to the global optima solution.
- #quality: Number of different quality/fitness values of local optima in the LON network.
- #neut: Number of local optima connected to neutral connected nodes.
- #com: Number of connected components in the LON network.
- #e_impr: Number of improving edges in LON network.
- #e_wors: Number of worsening edges in LON network.
- %imp: Percentage of improving edges with respect to the total edges in the LON network.
- #e_neut: Number of neutral edges in LON network.

Table 2
Networks metrics using p=1

Instance	Best	#optima	#glob	#g_e_imp	#quality	#com	#dep	%dep	avg_d_g
Random uniform instances									
E506.25	16,313,719	1,241	1	14	1,223	1	16	1.3	1.94
E755.73	20,158,565	49,549	1	72	36,519	1	107	0.2	1.70
E1010.37	22,904,325	103,140	1	77	57,987	1	154	0.1	1.92
E1243.85	25,227,141	611,634	1	287	76,935	1	592	0.1	1.73
E1521.33	28,027,563	1,575,006	1	105	84,911	1	2,150	0.1	2.19
Random clustered instances									
C506.25	6,816,950	9,738	1	88	7,956	1	90	0.9	1.63
C755.73	9,867,050	7,093	1	73	6,863	1	153	2.2	1.97
C1010.37	10,716,003	262,733	1	129	62,239	1	170	0.1	1.56
C1243.85	12,943,477	520,374	1	48	189,635	1	965	0.2	2.24
C1521.33	13,608,402	1,094,690	1	123	252,522	1	1,213	0.1	2.34
Somewhat structured instances (city problems)									
att532	27,686	8,707	2	215	279	1	115	1.3	1.87
gr666	294,358	18,093	2	129	2,435	1	141	0.8	1.90
pr1002	259,045	176,984	1	64	2,468	1	88	0.1	1.84
rl1304	252,948	49,005	1	275	8,109	1	332	0.7	2.68
nrw1379	56,638	4,584,800	96	14,252	231	1	9,862	0.2	2.50
Highly structured instances (drilling problems)									
u574	36,905	2,318	4	90	209	1	29	1.3	2.03
rat783	8,806	291,591	1,024	394	50	1	1,032	0.4	1.87
d1291	50,801	5,120,877	512	71,437	1,817	1	65,809	1.3	-

- #dep: Number of depositing nodes in LON network.
- avg_d_g: Average distance to global optima solution in DLON network.

5. Results

This section discusses the obtained network metrics for $\mathcal{MMAS}$. The organization of the section is as follows: Section 5.1 presents the metrics and the analysis of the results, Section 5.2 presents a comparison between the LON networks of $\mathcal{MMAS}$ and an Iterated Local Search algorithm, Section 5.3 includes a comparison of the networks when changing pheromone influence information ($\beta = 0$), Section 5.4 presents a comparison with RAS, and Section 5.5 presents a DLON plot analysis considering one instance per subset of instances.

5.1. Metrics results and Analysis

We consider three values for $p = \{1, 2, 5\}$ to study the pheromone persistence representation in constructing new candidate solutions. The idea is to simulate and analyze the effect of different length persistence scenarios on the network structure. We have chosen persistence values consistent with the analysis performed in section 3.2. Tables 2 and 3 present the results of the metrics with $p = 1$. For the random uniform instances, some metrics increase with instance size, indicating larger networks. Specifically, this trend is observed in the number of local optimal solutions, the number of quality values, the number of improving and worsening edges, and the number of deposit nodes. This effect is related to the size of the search space, defined by the number of cities in each instance. The number of worsening edges increases significantly compared to the other metrics mentioned above. This behavior indicates a greater exploration of $\mathcal{MMAS}$ when solving these problem instances. Furthermore, the percentage of depositing nodes is low regarding the network size, with the largest value being only 1% of the nodes. This suggests that only a small fraction of nodes contribute to the pheromone learning process.

A similar tendency is observed in the random clustered instances, where an increase in some metrics correlates with the instance size (#e_wors and #dep). Note that the number of deposit nodes (#dep) in the random clustered instances is generally higher than in the uniform instances. This situation can be related with solution diversity during the search, as it indicates that $\mathcal{MMAS}$ performs exploitation widely in the search space, allowing different solutions to deposit

Table 3
Networks metrics using p=1 (cont.)

Instance	#e_impr	#e_wors	% imp	#e_neut
Random uniform instances				
E506.25	84	1,914	4.2	0
E755.73	1,394	107,296	1.3	0
E1010.37	2,748	245,961	1.1	0
E1243.85	9,586	1,349,956	0.7	3
E1521.33	68,442	3,930,427	1.7	11
Random clustered instances				
C506.25	1,602	21,992	6.8	1
C755.73	6,600	30,835	17.6	0
C1010.37	1,853	360,475	0.5	0
C1243.85	18,143	1,106,612	1.6	1
C1521.33	40,766	2,490,103	1.6	3
Somewhat structured instances (city problems)				
att532	1,753	21,836	7.3	321
gr666	2,230	100,652	2.2	40
pr1002	1,220	265,647	0.5	42
rl1304	8,625	128,541	6.3	23
nrw1379	90,097	8,074,230	1.1	89,459
Highly structured instances (drilling problems)				
u574	202	7,241	2.7	43
rat783	463	1,792,378	0.0	298,808
d1291	195,272	6,551,812	2.8	166,067

pheromones. In this subset, the #optima, #quality, and #e_impr are correlated with the instance size. However, the only network that does not show correlations in these metrics is *C755.73*. In this network, the number of optima and quality values is considerably low compared to other random uniform and clustered instances. Moreover, the largest percentage of deposit nodes and %imp were obtained for this instance (near 2% and 18% respectively). This indicates that the search landscape defined by $\mathcal{MMAS}$ for this instance appears to be smooth, allowing regular improving transitions and thus, an effective exploitation.

In city problems, compared to the previously mentioned subsets, the metrics are not heavily correlated to the instance size. This behavior is linked to a higher level of neutrality in the landscape defined by these instances. The number of neutral nodes (and edges) for random instances is consistently lower than those observed in city problems, indicating the existence of plateaus that contain local optima solutions with the same quality consecutively visited by $\mathcal{MMAS}$. For example, we see a significant increase in neutrality when comparing the instance *att532* to *C506.25*, a random clustered instance with a similar size. The number of quality values drops from 7,956 in *C506.25* to 279 in *att532*, while the number of neutral edges increases from 1 in *C506.25* to 321 in *att532*. This indicates that the search space defined by *att532* contains different levels of solutions defined by their qualities. In this case, $\mathcal{MMAS}$ is able to traverse the search space generating a fitness landscape that exhibits high neutrality. On the contrary, the search space defined by *E1243.85* and *E1521.33* also contains several levels but the neutrality of the fitness landscape defined by $\mathcal{MMAS}$ is rather low.

The network metrics of city problems instances notably vary between them. An example of this can be observed comparing networks of *gr666* and *pr1002* instances. The #optima are one order of magnitude of difference (*pr1002* larger than *gr666*), have similar #quality values and avg_d_g, and different #dep. However, *gr666* has two global optima and a greater number of incoming improvement edges to the global optimum. This evidence that *gr666* has a highly connected network, that can be traversed by different paths of $\mathcal{MMAS}$ to reach the global optima a higher number of times.

The instance *nrw1379* also exhibits interesting characteristics, particularly that it has the highest number of optima (96 global optima) and worsening, improving, and neutral edges compared to other instances with a similar number of cities (e.g., *C1521.33* and *E1521.33*). Again, the neutrality in this instance can be attributed to the low number of

Table 4
LON metrics with p=2

Instance	%neut	%e_impr	%e_wors	%e_neut
Random uniform instances				
E506.25	0.0	2.4	0.9	0.0
E755.73	0.0	1.9	3.4	0.0
E1010.37	0.0	2.1	6.0	0.0
E1243.85	0.0	1.3	4.0	0.0
E1521.33	0.0	0.8	4.7	0.0
Random clustered instances				
C506.25	0.0	1.5	2.6	0.0
C755.73	0.0	0.9	5.8	0.0
C1010.37	0.0	1.9	3.5	0.0
C1243.85	0.0	2.1	8.2	0.0
C1521.33	0.0	0.3	8.0	0.0
Somewhat structured instances (city problems)				
att532	0.8	2.5	3.0	4.7
gr666	0.0	1.7	3.3	0.0
pr1002	0.0	2.3	7.7	0.0
rl1304	5.7	2.6	10.1	4.3
nrw1379	1.7	1.2	3.1	0.8
Highly structured instances (drilling problems)				
u574	0.0	2.0	2.3	0.0
rat783	0.3	2.4	3.8	1.0
d1291	0.6	0.6	3.7	0.5

quality values (231), allowing $\mathcal{MMAS}$ to identify several structurally different local optima with the same quality value. About the exploitation of $\mathcal{MMAS}$, this instance appears to be easy to reach the global optima solutions, considering that #g_e_imp is two orders of magnitude larger than the other instances of the subset.

In drilling problems, the metrics are different from other sets, considering the highest number of global optima and neutral edges (obtained by *rat783*), the highest number of optima, neutral, and deposit nodes (obtained by *d1291*). In this subset, the number of optima increases with the number of cities. However, the increment size is greater than in the other subsets of instances. The number of different quality values is less than the 0.5% of the obtained optima (for *rat783* and *d1291*) and 9% for *u574*, showing the high probability that $\mathcal{MMAS}$ reaches a plateau of local optima solutions during its search process. The network of *d1291* has metrics that exhibit the exploitative capabilities of $\mathcal{MMAS}$ to reach an optimal solution. This network has a large set of globally optimal solutions (512), and consequently, a large number of incoming improving edges (71,437) to these global optimal solutions. Between the instances considered in this study, it has the highest value for the #e_impr metric.

Increasing the value of p: Table 4 presents results for $p = 2$, focusing on the percentage of variation compared to $p = 1$ concerning the number of neutral nodes, improving edges, worsening edges, and neutral edges. In the random sets of instances analyzed, there were no changes in the numbers of neutral nodes or edges. Since the presence of neutrality in these instances is very low, changes in the persistence of the pheromone do not significantly affect these metrics. However, slight variations can be observed in the percentage of improving edges, which range from 0.3% to 2.4%. On the other hand, the percentage of worsening edges showed a larger increase, ranging from 0.9% to 8.2%. These changes are expected given the structure of the networks analyzed for $p = 1$, where no neutrality was observed and the number of worsening edges significantly exceeded the number of improving edges. In this sense, the network using $p = 2$ follows the same trend.

For structured instances (city and drilling), the improving edges also slightly increment as well as the worsening edges, having the larger increment with a maximum value of 10% in *rl1304*. However, in these subsets, the number of neutral nodes and edges increment in some instances, ranging from 0.3 to 5.7 for the nodes and 0.5 to 4.7 for the edges. There are some networks where the neutrality values do not change: *gr666*, *pr1002* and *u574*. This behavior is

Table 5
LON metrics using p=5

Instance	%neut	%e_impr	%e_wors	%e_neut
Random uniform instances				
E506.25	0.0	13.1	2.9	0.0
E755.73	0.0	11.7	9.2	0.0
E1010.37	0.0	12.0	13.4	0.0
E1243.85	0.0	13.2	13.0	0.0
E1521.33	25.0	9.9	16.8	18.2
Random clustered instances				
C506.25	0.0	8.6	6.9	0.0
C755.73	0.0	4.4	10.6	0.0
C1010.37	0.0	17.3	7.7	0.0
C1243.85	100.0	14.3	18.2	100.0
C1521.33	66.7	11.5	19.0	66.7
Somewhat structured instances (city problems)				
att532	2.4	12.2	8.3	13.4
gr666	4.7	10.2	9.3	5.0
pr1002	9.4	16.9	15.3	7.1
rl1304	20.0	12.5	20.1	17.4
nrw1379	10.5	17.2	14.6	14.5
Highly structured instances (drilling problems)				
u574	8.7	8.9	6.4	2.3
rat783	1.4	18.4	13.5	5.9
d1291	10.5	10.1	17.8	8.0

related to the presence of separated regions of the search space where nodes have the same quality values. Increasing the p value is not enough to produce new neutral edges that connect these separated regions. The model with $p = 2$ exhibits behavior similar to that of $p = 1$, with only small differences in exploration and exploitation. However, more noticeable changes were observed in the structured instances.

Table 5 displays the percentage increment of computed metrics observed when changing the parameter p from 1 to 5 in the network. In 11 out of 18 instances the improving edges have a larger percent increment compared to numbero of worsening edges, nevertheless the ratio of improving and worsening edges is not significantly changed by this increment. This indicates that the networks include new trajectories when the pheromone is considered to persist for 5 iterations. For these networks, 3 random instances show and increment in the neutrality level compared to the networks obtain by setting $p = 1$ and $p = 2$. This behavior suggests that allowing the persistence of the pheromones over a larger number of iterations can connect local optima in the same quality level in the search space. The increased percentages range from 25% to 100% for neutral nodes and from 18.2% to 100% for neutral edges. However, note that these percentages result from the connection of at most 2 new neutral nodes in the network. Additionally, it is noteworthy that the number of improving and worsening edges increase strongly compared to the network defined by $p = 2$. The percentage of improving edges now falls between 4.4% and 17.3%, while the percentage for worsening edges ranges from 2.9% to 19.0%.

For the structured instances, a similar trend is observed as in the random instances, with an overall increase of up to 20% in both nodes and edges. Additionally, networks that previously showed no increase in neutrality for $p = 2$ (specifically, *gr666*, *pr1002*, and *u574*) now exhibit the addition of neutral edges and nodes. This change increases neutrality measurements, ranging from 4.7% to 9.4% in %neut and from 2.3% to 7.1% in %e_neut. The most significant change occurs in the *rl1304* network, which shows the highest increase in the neutral nodes and worsening edges when comparing $p = 2$ and $p = 5$ against all other structured instances.

In general, the increment in the value of p has a low effect on the composition of the networks, it changes the number of edges according to each instance's characteristics without affecting the overall fitness landscape analysis. Interestingly, the increment of the persistence level allows for new neutral edges and nodes showing how close different neutral regions are in each problem instance concerning pheromone-based trajectories.

Table 6
LON metrics and performance comparison of $\mathcal{MMAS}$ and ILS. Results obtained from 1,000 runs of each algorithm with 1,000 evaluations with no improvement as termination criterion.

Instance	Best		Success rate		#evaluations		#optima		#quality		#glob	
	$\mathcal{MMAS}$	ILS	$\mathcal{MMAS}$	ILS	$\mathcal{MMAS}$	ILS	$\mathcal{MMAS}$	ILS	$\mathcal{MMAS}$	ILS	$\mathcal{MMAS}$	ILS
Random uniform instances												
E506.25	16,313,719	16,313,719	100.0 %	100.0 %	26	130	1,241	189	1,223	189	1	1
E755.73	20,158,565	20,158,565	100.0 %	100.0 %	56	868	49,549	2,305	36,519	2,265	1	1
E1010.37	22,904,325	22,904,325	100.0 %	100.0 %	100	2,600	103,140	5,051	57,987	4,858	1	1
	25,227,141	25,227,141	**100.0** %	36.9 %	122	7,025	611,634	35,484	76,935	24,663	1	1
	28,027,563	28,027,563	**95.5** %	1.0 %	5,999	3,482	1,575,006	67,736	84,911	36,104	1	1
Random clustered instances												
C506.25	6,816,950	6,816,950	100.0 %	100.0 %	28	542	9,738	1,084	7,956	1,054	1	1
C755.73	9,867,050	9,867,050	100.0 %	100.0 %	257	3,327	7,093	622	6,863	620	1	1
C1010.37	10,716,003	10,716,003	**100.0** %	99.9 %	45	1,785	262,733	15,005	62,239	11,254	1	1
	12,943,477	12,944,075	**73.7** %	0 %	3,684	-	520,374	62,648	189,635	51,053	1	1
	13,608,402	13,609,537	**99.5** %	0 %	2,519	-	1,094,690	69,337	252,522	57,178	1	1
Somewhat structured instances (city problems)												
att532	27,686	27,686	100.0 %	100.0 %	25	150	8,707	1,208	279	139	2	2
	294,358	294,358	**100.0** %	98.9 %	154	3,293	18,093	1,151	2,435	648	2	2
	259,045	259,045	**100.0** %	74.5 %	55	2,281	176,984	4,685	2,468	1,320	1	1
rl1304	252,948	252,948	100.0 %	100.0 %	46	2,243	49,005	1,896	8,109	1,369	1	1
	56,638	56,638	**100.0** %	76.3 %	292	5,446	4,584,800	202,150	231	142	96	96
Highly structured instances (drilling problems)												
u574	36,905	36,905	100.0 %	100.0 %	25	151	2,318	258	209	77	4	4
rat783	8,806	8,806	100.0 %	100.0 %	25	34	291,591	6,986	50	29	1,024	939
	50,801	50,801	**100.0** %	24.3 %	92	8,134	5,120,877	932,890	1,817	792	512	141

5.2. ILS Comparison

To lighten the effect of a strong local search procedure and better understand the ACO behaviour, in this section, we compare the results obtained using $\mathcal{MMAS}$ against the results obtained using a simple ILS algorithm.

Table 6 shows the quality of the best local optima reached in the executions, the success rate (percentage of the executions that reached the global optimum), the mean number of evaluations required to reach the global optimum, the number of visited local optima, the number of different qualities, and the number of global optima for each algorithm. Note that executions that were not able to find a global optimum are not considered when calculating the mean number of evaluations metric. Bold values highlight the best-performing algorithm in the comparison.

These results show that both $\mathcal{MMAS}$ and ILS have a 100% success rate on 9 out of 18 of the instances tested; thus, these instances are not challenging for the algorithms with the setup used in these experiments. Despite this, the effort required to solve these instances can be measured by the average number of evaluations required to reach the global optima. In general, the results show that $\mathcal{MMAS}$ performs better than ILS on the set of instances. On average, ILS requires more evaluations than $\mathcal{MMAS}$ to reach a global optimum. Furthermore, three of the largest instances (*E1521.33*, *C1234.85*, and *C1521.33*) are very challenging for ILS ($0-1\%$ success rate), while $\mathcal{MMAS}$ obtains much larger success rates ($73.7-99.5\%$ success rate). We explain the performance difference by the exploration mechanism implemented in both algorithms. For $\mathcal{MMAS}$, new search paths are produced incorporating search space knowledge (the pheromone and heuristic information), this yields a biased exploration towards relatively interesting areas of the search space. On the other hand, the perturbation mechanism on ILS (2-exchange movements) is not flexible and can not be adapted to the search state. Hence, solving larger instances is challenging for ILS and it requires the algorithm to use more evaluations for solving smaller instances. Note that for *rat783* and *d1291* $\mathcal{MMAS}$ found more global optima than ILS. This analysis is reinforced when observing the large number of local optima and different qualities explored by $\mathcal{MMAS}$.

5.3. Pure Pheromone-driven Search Comparison ($\beta = 0$)

Table 7 presents the results of the comparison of $\mathcal{MMAS}$ with different settings for the use of heuristic information. The table provides the percentage change between the LON metrics of $\mathcal{MMAS}$ using $\beta = 2$ with respect to $\beta = 0$ (no heuristic information). It is computed as $100 \cdot (v_{\beta=2} - v_{\beta=0})/v_{\beta=2}$. This representation quantifies the relative impact of incorporating heuristic information during solution construction. We consider one problem instance per instance type in this comparison. The setting $\beta = 0$ corresponds to a pure pheromone-driven search, while $\beta = 2$

Table 7
Percentage change between $\mathcal{MMAS}$ with $\beta = 2$ and $\beta = 0$

Instance	#optima	#g_e_impr	#quality	#neut	#e_impr	#e_wors	%imp	#e_neu	#dep	avg_d_g
E506.25	1.5	-21.4	1.5	0.0	-10.7	2.9	-13.4	0.0	-12.5	33.3
C755.73	-5.8	-2.7	-5.7	0.0	-5.4	-2.5	-2.3	0.0	-7.2	11.9
att532	-6.9	51.6	-2.5	11.8	-5.4	-4.5	-0.8	-6.9	1.7	26.7
u574	-8.5	72.2	-4.3	-13.0	-36.6	-8.6	-24.9	-7.0	-3.4	43.7

Table 8
Percentage change between $\mathcal{MMAS}$ and RAS.

Instance	#optima	#g_e_impr	#quality	#neut	#e_impr	#e_wors	%imp	#e_neu	#dep	avg_d_g
E506.25	8.9	0.0	9.2	0.0	2.4	6.8	-4.5	0.0	6.3	22.7
C755.73	4.9	-31.5	4.6	-100.0	-17.4	-21.2	2.7	-100.0	1.3	14.4
att532	8.8	55.4	4.3	-133.9	16.4	1.1	14.2	7.5	8.7	37.6
u574	3.8	70.6	-2.9	-69.6	-28.2	-10.5	-15.6	9.3	-10.3	43.4

incorporates both pheromone and heuristic guidance. This highlights how changing the balance between pheromone reinforcement and heuristic bias affects the structure of the search space. The results show clear structural differences in the search space explored by both settings. For the number of optima (#optima), most instances exhibit negative percentage changes (ranging from -5.8% to -8.5%), indicating that $\beta = 2$ produces fewer optima than $\beta = 0$. This reinforces the idea that adding heuristic information reduces dispersion in the search space and concentrates exploration in fewer regions. The only exception is *E506.25*, which shows a small positive change of 1.5%.

For the number of improving edges to global optima, the results present a heterogeneous behavior. Some instances show slight decrements (e.g., -21.4% and -2.7%), others show substantial increments (e.g., 51.6% and 72.2%). The number of distinct fitness values shows small negative variations throughout most instances. This means $\beta = 2$ slightly reduces the diversity of fitness levels. This supports the interpretation that heuristic guidance produces a more focused search, reducing exploratory variability compared to the purely pheromone-driven setting. From a connectivity perspective, neutrality-related metrics (#neut and #e_neut) show limited, instance-dependent variation. In most cases, changes are small or null, suggesting that neutrality is mostly determined by the problem structure rather than the β setting. The presence of both positive and negative changes (e.g., 11.8% and -13.0%) further supports this interpretation. About improving edges, they generally decrease when using $\beta = 2$ (e.g., -5.4% and -36.6%). In contrast, worsening edges show much smaller changes. Consequently, the percentage of improving edges also decreases, sometimes substantially (e.g., -24.9%). This indicates that while heuristic information enhances access to global optima, it does not increase the proportion of improving transitions. Thus, its primary effect is structural, rather than simply boosting improvement rates. The number of depositing nodes (#dep) shows relatively small variations between instances. This indicates pheromone updates are distributed across a similar subset of the network in both configurations. The main differences come from the interaction of pheromone and heuristic information, rather than from the number of contributing nodes. Regarding the average distance to the global optima, setting $\beta = 2$ yields higher values, resulting in more dispersed LON structures. Thus, while heuristic information enhances guidance to global optima (as observed in #g_e_impr), it also increases structural distance within the search space. Conversely, $\beta = 0$ yields more compact networks, in which global optima are closer to other nodes.

5.4. RAS Comparison

Table 8 presents the percentage changes between $\mathcal{MMAS}$ and RAS metrics calculated as $100 \cdot (v_{\text{MMAS}} - v_{\text{RAS}})/v_{\text{MMAS}}$, which quantifies the impact of the pheromone update procedures of both variants on the structure of the LON. As in β analysis, we consider one problem instance per instance type.

Overall, the results show that the pheromone update strategies of $\mathcal{MMAS}$ and RAS produce structural differences in the search space. With respect to the number of optima (#optima), all instances show positive values (ranging from 3.8% to 8.9%). This demonstrates that $\mathcal{MMAS}$ produces more local optima than RAS in these four instances. The bounded pheromone mechanism in $\mathcal{MMAS}$ seems to promote diversified exploration. In contrast, the rank-based pheromone reinforcement in RAS tends to focus the search on fewer regions. The accessibility of global

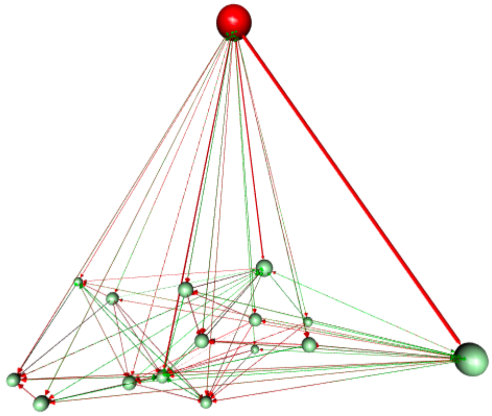


Figure 4: DLON of the instance E506.25. The red node is a global optimal solution, and the green nodes are local optimal solutions. Green edges represent improving transitions, and red ones represent worsening transitions.

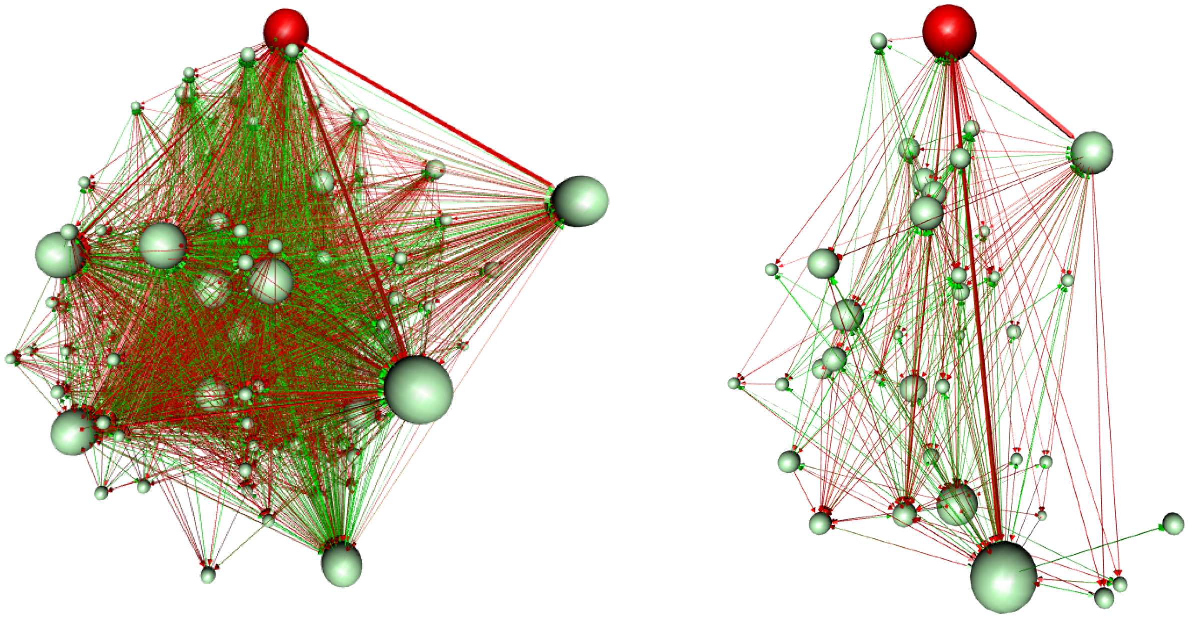
optima (#g_e_impr) depends on the instance. RAS yields higher values in some cases (e.g., *C755.73*, -31.5%), while $\mathcal{MMAS}$ yields higher values in others (e.g., *att532* and *u574*, with increases above 50%). This suggests that rank-based pheromone reinforcement in RAS can either enhance or restrict access to global optima, depending on the instance. $\mathcal{MMAS}$, by contrast, shows more stable behavior across instances. The number of distinct quality values (#quality) remains similar for both algorithms, with only minor variations (near 10%). This indicates that both variants explore solution spaces with comparable fitness diversity despite differing reinforcement strategies. A more pronounced difference is seen in neutrality-related metrics. Negative values in #neut and #e_neu (e.g., -100% and -133.9%) show that RAS introduces significantly more neutral structures than $\mathcal{MMAS}$ in some cases. This suggests that the rank-based update scheme may create plateau-like regions where multiple solutions have similar fitness values. These regions may influence search behavior. Analysis of edge dynamics shows contrasting behaviors between the algorithms. In certain instances (e.g., *att532*), $\mathcal{MMAS}$ produces more improving transitions (16.4%), whereas in others (e.g., *u574*), RAS exhibits a higher number of improving edges (-28.2%). Worsening edges (#e_wors) are generally more frequent in RAS (negative values), indicating a less intensive search process. The percentage of improving edges (%imp) also varies across the instances. These results indicate that no variant consistently dominates in terms of transition quality. The number of depositing nodes (#dep) shows only small differences, indicating both algorithms distribute pheromone updates across similar portions of the search space. However, the effects of these updates differ due to the distinct reinforcement mechanisms in each algorithm. About the average distance to global optima (avg_d_g), $\mathcal{MMAS}$ consistently yields higher values (ranging from 14.4% to 43.4%). This result indicates that $\mathcal{MMAS}$ induces a more dispersed search structure, with global optima located farther from other regions of the network. In contrast, RAS produces more compact landscapes, suggesting stronger convergence pressure from its rank-based reinforcement.

5.5. DLON plots

This section presents DLON 3D plots, which focus exclusively on deposit nodes. To obtain a DLON, the LON network is simplified by removing all exploration nodes. In DLON plots, both nodes and edges have attributes that indicate the quality and connectivity features of the network. Red nodes represent the solutions with the largest quality values, while green nodes indicate regular local optima solutions. The position of the nodes in each plot correlates with their quality. The z-axis represents the inverted fitness of the solutions, meaning that nodes located in the upper area of the plot have better quality values than those in the lower area. The force-directed graph layout from the R igraph library defines each plot's x and y axes. The size of the nodes is determined by the number of outgoing edges, which represent their historical visit frequency (across all independent executions). It allows us to quantify their pheromone influence on generating new candidate solutions. Moreover, node sizes are computed before nodes are removed from the network to capture the total influence of deposit nodes. Regarding edge colors, green edges indicate an improving transition, red edges indicate a worsening transition, and blue edges represent neutral transitions. Additionally, the width of the edges reflects the normalized weight, showing the number of transitions performed between pairs of nodes.

To simplify the visualization of networks, we selected one instance from each subset. We consider $p = 1$ and focus on instances with a low number of deposit nodes, which eases the understanding of the plots.

Figure 4 displays the DLON plot of $\mathcal{MMAS}$ ($\beta = 2$) for the instance *E506.25*, which has the lowest number of deposit nodes (16) from the instances considered in this study. It can be observed that the global optimal solution has been reached more frequently than the other 15 nodes (green nodes). Note that only one deposit node is not directly connected to the global optima by an improving transition. As the termination criterion was set at 400 iterations without improvement (10,000 evaluations), a large number of deteriorating edges from the global optimum can be observed across nearly all deposit nodes. The rightmost node is the easiest to access from the global optimum node, as it has been visited the most among the green nodes. Additionally, no neutrality is observed in this network.



(a) Original DLON

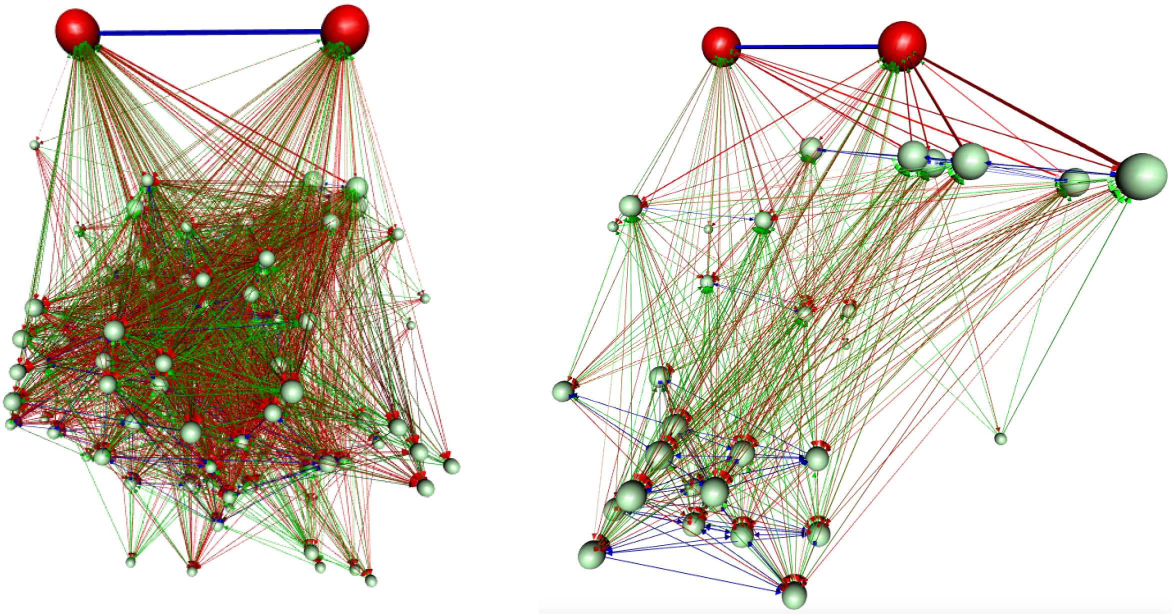
(b) Filtered DLON (25%)

Figure 5: DLON of the instance C755.73, considering all deposit nodes (left), and a filtered version considering the 25% of the best-quality ones (right). The red node is a global optimal solution, and the green nodes are local optimal solutions. Green edges represent improving transitions, and red ones represent worsening transitions.

Figure 5 shows the DLON plot for the C755.73 instance (left) and a filtered version with the 25% top-quality solutions (right). This network shows multiple local optimum solutions that are larger in size than the global optimum solution. In this instance, $\mathcal{MMAS}$ frequently revisits local optimum solutions, which limits its exploration and may lead to difficulty escaping regions. Furthermore, it can be observed that the paths generated by $\mathcal{MMAS}$ involve regularly improving transitions. This is evidenced by the visible number of improving (green) edges in the network. Moreover, the %imp increases from 17.63% in the LON model, to a 77% in the DLON model.

Figure 6 presents the DLON plot for the *att532* instance (left), and a filtered version with the 25% top-quality solutions (right). $\mathcal{MMAS}$ obtained two structurally different global optima solutions connected by the widest neutral edge in the network. Also, these global optima were noticeably visited many times more than all the other nodes in the network, showing that this instance can be easy to solve for $\mathcal{MMAS}$. Additional neutral edges can be observed inside the network, demonstrating the presence of plateaus between deposit nodes. The percentage of neutral edges in this network reaches 6% with respect to the total number of edges. Moreover, a large number of improving edges can also be visualized in this case. Here, the %imp increases from 7.33% (LON model) to 73% (DLON model).

Figure 7 presents the DLON plot for the *u574* instance. This network features four distinct global optimum solutions, which are significantly larger than all other nodes in the network. Similar to the *att532* network, the edges connecting these four global optimum solutions are wider than those connecting the other nodes, indicating that



(a) Original DLON (b) Filtered DLON (25%)

Figure 6: DLON of the instance att532, considering all deposit nodes (left), and a filtered version considering the 25% of the best-quality ones (right). The red nodes represent the global optimal solutions found by ACOTSP, and the green nodes represent the local optimal solutions. Blue edges represent neutral transitions, green ones represent improving transitions, and red ones represent worsening transitions.

$\mathcal{MMAS}$ possibly visits these four solutions sequentially. Regarding neutrality, the network exhibits clearly three levels of plateaus: the highest level contains the four global optimum solutions, which are interconnected; the middle section includes the largest green nodes of the network; and the lowest portion of the plot contains additional neutrally connected nodes. This pattern demonstrates that $\mathcal{MMAS}$ can explore various levels of neutrality, navigate through different quality levels, and ultimately reach the best solutions. The neutrality level, related to the neutral edges of this network, increases from 0.57% in the LON to 9.17% in the DLON.

Furthermore, the trajectories of $\mathcal{MMAS}$ show that local optima in the lower section of the plot (lower quality ones) are directly connected to the global optimum solutions without necessarily traversing the intermediate levels of neutrality. This behavior reflects the intensification capabilities of $\mathcal{MMAS}$ and the LK local search procedure.

6. Discussion

This section discusses the design of the proposed LON model, its generalization to other ACO algorithms, and its main limitations. Additionally, we present a brief analysis of the obtained results, contrasting them with existing classical fitness landscape analyses for TSP.

6.1. Generalization Guidelines and Limitations

The LON model presented in this work was designed for representing the fitness landscape of $\mathcal{MMAS}$. In this section, we discuss how this model can be applied to other ACO algorithms and the limitations of its design.

Local search usage: Given that the proposed model is LON model, it required the application of a local search procedure after each ant constructs a solution. Although in this work the LK procedure was considered, the model can be directly applied with any alternative local search procedure. Consequently, the model may be limited in scenarios where: (1) full visited solutions information is required (e.g., intermediate visited solutions in local search) or (2) when no local search procedure is applied. However, some considerations may address these limitations. In the first scenario

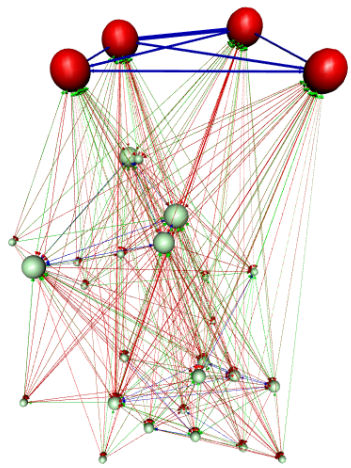


Figure 7: DLON of the instance u574. The red nodes are the global optimal solutions reached by ACOTSP, and the green nodes are the local optimal solutions. Blue edges represent neutral transitions, green ones represent improving transitions, and red ones represent worsening transitions.

(1), non-optimal and locally optimal solutions may be represented independently using a different node type. This enables the analysis of differentiated network's features that may allow better understanding of the fitness landscape. Without local search (2), the number of considered visited solutions and thus nodes in the network increases. This would make the networks not only large but also more complex, obscuring the analysis of search space ruggedness and landscape features. Another possibility is to apply the STN model, as STN analysis can consider both local and non-optimal solutions [25]. However, direct comparison between LON and STN models is not feasible due to their different representations of the algorithmic search process. In STN models, it is necessary to define locations (delimited partitions of the search space) and representative solutions (e.g., the highest-quality solution in each location). As a result, the STN model is focused on visited areas of the search space emphasizing the algorithm's trajectory, while the LON model is focused on local optima and their connectivity.

ACO variants: There are several ACO variants in the literature, we offer recommendations to adapt the model proposed in this work to ACO algorithms different than $\mathcal{MMAS}$. These suggestions cover the most common variants: Ant System (AS), Ant Colony System (ACS), Elitist Ant System (EAS), and Population-based Ant Colony Optimization (P-ACO) [29]. In AS, all ants deposit pheromone in each iteration. In the proposed model, the network will contain deposit nodes only and thus, exploration nodes will not exist. In this scenario, we recommend the definition of quality thresholds to analyze subsets of nodes and establish different levels of influence based on solution quality. The idea would be to depict which solutions can highly bias the search process. In ACS, pheromone evaporation and deposit is performed during construction and at the end of each iteration. The paths considered during the construction of all ants are slightly reinforced and evaporated to increase diversity. At the end of each iteration, the best-quality solution deposits pheromone, as in $\mathcal{MMAS}$. Our proposal models transitions based on changes in pheromone performed at the end of each iteration. Here, we suggest complementing the analysis by defining different edge types (e.g. to represent the generation of solutions in the same iteration) or different-weight edges (e.g., to measure negative influences between solutions). About EAS, in each iteration all ants deposit pheromone as in AS. Additionally, an elite ant (usually, the best-so-far) performs a number of extra deposits defined by a parameter. As we suggest in AS and RAS, an extension of the proposed model could analyze the different deposit types through separate edges or weights. In P-ACO, a set of good-quality solutions is maintained in an external archive, which directly determines the pheromone values. Consequently, pheromone influence is not driven by a single depositing solution per iteration, but by the set of solutions currently stored in the archive. At each iteration, one or more candidate solutions may enter the archive according to a given archive management strategy (e.g., age-based or quality-based), while existing solutions may be removed to maintain a bounded archive size. Therefore, adapting the proposed LON model could require redefining deposit nodes as archive-active local optima, replacing pheromone persistence with archive residence time, and explicitly

Table 9
Results of fitness distance correlation analysis.

LS operator	Instance	min d_{opt}	$avg(d_{opt})$	$avg(d_{loc})$	FDC
3-opt	att532	36	106.48(0.20)	123.17	0.57
	rat783	83	151.82(0.19)	184.77	0.68
	pr1002	123	203.00(0.20)	242.16	0.57
LK	att532	47	106.29(0.20)	122.71	0.54
	rat783	75	151.38(0.19)	184.51	0.70
	pr1002	108	202.15(0.20)	241.77	0.60

representing both archive entry and archive removal events. A main limitation of this extension is that the pheromone effect becomes less direct, since newly generated solutions are influenced by the aggregate pheromone from the entire archive rather than by a single depositing solution. As a result, the inferred connections in the LON approximate the underlying search dynamics rather than a direct causal relationship.

Persistence: In this work, we simulate pheromone persistence across 1, 2, or 5 iterations. Theoretical analysis indicates that estimating the effects of evaporation and simultaneous deposition on different edges is highly complex. We selected these persistence values to represent iterations in which a deposit would significantly influence solution construction. However, the analysis may yield different results if higher levels of persistence are considered. Such setting would increase network connectivity that may obscure relevant influences in the search process, these effects should be considered in the analysis when using higher levels of persistence.

6.2. Fitness Distance Correlation

We present a discussion of the obtained results considering existing fitness landscape analysis techniques in the literature, in particular a Fitness Distance Correlation (FDC) study. The fitness-distance correlation coefficient measures the correlation between the fitness of visited local-optimum solutions and their distance from the known global optimum [15]. Based on a sampling of local optima solutions, if their fitness increases as the distance to the optimal solution decreases, then a “big valley structure” could be predicted, and the classical local search process should be more successful. In [20, 21], the author studies the FDC using Lin-Kernighan and 3-opt operators on TSP problem instances. The authors define the distance between two TSP solutions as the number of edges that differ between the two tours. Moreover, they select 2,500 random initial solutions, apply the local search operator, and obtain a local optimum, whose distance from the known global optima is measured.

Our analysis involves the entire ACO search process, with 1,000 independent executions and a termination criterion of 400 iterations without improvement. Thus, the ACO search processes visit a much higher number of local optima and zones of the search space than the study mentioned. For example, for rat783, $\mathcal{MMAS}$ visits 291,591 local optima solutions, while [20] evaluates at most 2500 local optima solutions. Table 9 presents a subset of the results from [20], focusing only on instances we have in common with their analysis. The table shows the minimum distance of the local optima to a global optimum (min d_{opt}), the average distance of the local optima to the global optimum ($avg(d_{opt})$), the average distance between the local optima (d_{loc}), and the fitness distance correlation coefficient.

From their results, a direct relationship between distance and problem size can be observed in most cases. However, *somewhat structured instances* (*att532* and *pr1002*) seem more complex than highly structured instances (*rat783*), showing lower FDC values. Through the local search operators, LK requires a few less number of steps to reach the optimal solution in *rat783* and *pr1002* compared to 3-opt. However, the average distances and FDC are similar. Compared to our results, the *pr1002* instance has the most incoming edges to global optima solutions (64), followed by *att532* (215) and *rat783* (394). Moreover, the number of distinct quality values among the visited local-optima solutions is only 50 in *rat783*, 279 in *att532*, and 2,468 in *pr1002*. Regarding the average distance to global optimum solutions, *att532* and *rat783* have 1.87 iterations, and *pr1002* has 1.84 iterations. This distance reflects the number of iterations, the number of ants (which produce solutions in each iteration), and the number of steps performed in each application of the LK local search operator. In general, both analyses reach the same conclusions about the difficulty of reaching the global optimum and the distance between local optima in the search space.

7. Conclusions and Future Work

In this work, we proposed a methodology to generate local optima networks for ant-based algorithms. In our model we consider nodes and edge definitions that aim to represent pheromone deposit and the persistence of the pheromone influence in the algorithm's trajectories. Edges are defined between local optima that deposited pheromone in an iteration and newly obtained local optima in the following p iterations after the deposit. Also, we propose to analyze a simplified version of the LON model named Deposit LON model, in which the focus of the analysis is the depositing nodes. The idea is to capture the exploitation and exploration behavior enhanced by the local optima that bias the search process. We studied the local optima networks generated by the well-known $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System algorithm using a Lin-Kernighan local search procedure over a set of TSP instances. We also compare these networks to the LONs generated by $\mathcal{MMAS}$ not using the heuristic information component ($\beta = 0$), an Iterated Local Search and a Rank-based Ant System implementations. Our analysis reveals that:

- The LONs generated by $\mathcal{MMAS}$ have a small number of improving edges compared to the number of worsening edges. These improving edges arise from local-optima solutions intensifying the search process and directing the pheromone-based trajectory. The greater prevalence of worsening edges reflects and measures $\mathcal{MMAS}$'s exploration capacity.
- All networks have a single connected component demonstrating that the pheromone schedule mechanism allows transitions that connect all visited local optima.
- Neutrality is a LON feature that arises from the interaction between the problem instance and the algorithm applied to solve it. In $\mathcal{MMAS}$, the instances with higher neutrality values are the structured instances, signaling plateaus of local optima within the fitness landscape.
- The change in the persistence level produce only slight changes in the networks, mostly observed in the number of worsening edges. This behavior can be compared with an empirically computed pheromone persistence, showing that pheromone deposit influence involves only a few iterations.
- Adjusting the parameter β results in structurally different search behaviors. When $\beta = 0$, the process favors pheromone-driven exploration without the influence of heuristic information. This produces more compact LON structures given that the algorithm can focus its trajectory on the pheromone information. In contrast, setting $\beta = 2$ increases reliance on heuristic guidance, improving accessibility to global optima but leading to a more distributed and complex landscape structure.
- The results show that the pheromone-guided exploration mechanism enables $\mathcal{MMAS}$ to achieve better performance than ILS in terms of search effectiveness. Mainly, this difference is due to search decisions influenced by accumulated pheromone trails, enabling $\mathcal{MMAS}$ to dynamically adapt its exploration to relatively interesting areas of the search space. In contrast, ILS relies on a rigid perturbation strategy that limits its performance, especially on large size instances.
- Distinct pheromone reinforcement strategies produce qualitatively different search behaviors. $\mathcal{MMAS}$ encourages broader exploration, generating more dispersed LON structures. In contrast, RAS focuses the search through rank-based updates, generating more compact landscapes and increasing convergence pressure.

The work presented in this paper is, to our knowledge, the first attempt to perform a fitness landscape analysis using local optima networks of an Ant Colony Optimization Algorithm. This analysis is aimed at validating that LONs can be used to model the fitness landscape of ACO algorithms.

As future work, we aim to enhance and expand our current research along different directions. First, the proposed model can be extended by comparing different local search procedures, ACO variants, or ACO parameter configurations. All these changes can clearly affect the performance of ant-based algorithms, and, as a consequence, the conclusions of our analyses may differ across ACO variants. Second, we are interested in applying this analysis to TSP problem instances with more cities (e.g., more than 1,521). The idea is to analyze whether the search space size can produce more disjoint trajectories and what effect the parameter settings have on the search process. Third, we aim to apply the proposed LON model to ACO algorithms for tackling diverse problems, such as dynamic combinatorial optimization [30] and multi-objective optimization [9]. Finally, we are also interested in using Search Trajectory Networks to study ACO algorithms without a local search component.

CRediT authorship contribution statement

Nicolás Rojas-Morales: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition. **Elizabeth Montero:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision. **Leslie Pérez:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Funding acquisition. **Gabriela Ochoa:** Conceptualization, Methodology, Writing - Review & Editing. **María Cristina Riff:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Funding acquisition.

References

[1] Adair, J., Thomson, S.L., Brownlee, A.E., 2024. Explaining evolutionary feature selection via local optima networks, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1573–1581.

[2] Bai, H., OuYang, D., Li, X., He, L., Yu, H., 2009. Max-min ant system on gpu with cuda, in: 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), IEEE. pp. 801–804.

[3] Bullnheimer, B., Hartl, R.F., Strauss, C., 1999. A new rank based version of the ant system: A computational study. Central European Journal for Operations Research and Economics 7, 25–38.

[4] Crawford, B., Soto, R., Johnson, F., Monfroy, E., Paredes, F., 2014. A max–min ant system algorithm to solve the software project scheduling problem. Expert Systems with Applications 41, 6634–6645.

[5] Dorigo, M., 1992. Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy .

[6] Dorigo, M., Blum, C., 2005. Ant colony optimization theory: A survey. Theoretical Computer Science 344, 243–278. doi:10.1016/j.tcs.2005.05.020.

[7] Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26, 29–41. doi:10.1109/3477.484436.

[8] Dorigo, M., Stützle, T., 2010. Ant Colony Optimization: Overview and Recent Advances. Springer US, Boston, MA. pp. 227–263. doi:10.1007/978-1-4419-1665-5_8.

[9] Falcón-Cardona, J.G., Leguizamón, G., Coello Coello, C.A., Castillo Tapia, M.G., 2022. Multi-objective ant colony optimization: an updated review of approaches and applications. Advances in Machine Learning for Big Data Analysis , 1–32.

[10] Fieldsend, J., Liefooghe, A., Malan, K., Verel, S., 2025. Local optima networks for constrained search spaces, in: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 204–212.

[11] Fieldsend, J.E., Alyahya, K., 2019. Visualising the landscape of multi-objective problems using local optima networks, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1421–1429.

[12] Herrmann, S., 2016a. Determining the difficulty of landscapes by pagerank centrality in local optima networks, in: European Conference on Evolutionary Computation in Combinatorial Optimization, Springer. pp. 74–87.

[13] Herrmann, S., 2016b. Determining the Difficulty of Landscapes by PageRank Centrality in Local Optima Networks. pp. 74–87. doi:10.1007/978-3-319-30698-8_6.

[14] Herrmann, S., Rothlauf, F., 2015. Predicting heuristic search performance with pagerank centrality in local optima networks, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA. pp. 401–408. doi:10.1145/2739480.2754691.

[15] Jones, T., Forrest, S., 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms, in: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. p. 184âĂŞ192.

[16] Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. 21, 498–516. doi:10.1287/opre.21.2.498.

[17] López-Ibáñez, M., Stützle, T., Dorigo, M., 2016. Ant Colony Optimization: A Component-Wise Overview. Springer International Publishing, Cham. pp. 1–37. doi:10.1007/978-3-319-07153-4_21-1.

[18] Malan, K.M., Engelbrecht, A.P., 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. Information Sciences 241, 148–163. doi:10.1016/j.ins.2013.04.015.

[19] Marco Dorigo, .T.S., 2004. Ant colony optimization.

[20] Merz, P., 2000. Memetic algorithms for combinatorial optimization problems : fitness landscapes and effective search strategies. Ph.D. thesis. Siegen University, Germany.

[21] Merz, P., 2012. Memetic Algorithms and Fitness Landscapes in Combinatorial Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 95–119. URL: https://doi.org/10.1007/978-3-642-23247-3_7, doi:10.1007/978-3-642-23247-3_7.

[22] Montgomery, E.J., 2005. Solution biases and pheromone representation selection in ant colony optimisation. Ph.D. thesis. Swinburne.

[23] Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Review 45, 167–256. doi:10.1137/S003614450342480.

[24] Ochoa, G., Chicano, F., Tinós, R., Whitley, D., 2015. Tunnelling crossover networks, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA. pp. 449–456. doi:10.1145/2739480.2754657.

[25] Ochoa, G., Malan, K.M., Blum, C., 2020. Search trajectory networks of population-based algorithms in continuous spaces, in: International Conference on the Applications of Evolutionary Computation (Part of EvoStar), Springer. pp. 70–85.

[26] Ochoa, G., Tomassini, M., Verel, S., Darabos, C., 2008. A study of NK landscapes' basins and local optima networks, in: Genetic and Evolutionary Computation Conference - GECCO 2008, ACM, New York, NY, USA. pp. 555–562. doi:10.1145/1389095.1389204.

[27] Ochoa, G., Veerapen, N., Daolio, F., Tomassini, M., 2017. Understanding phase transitions with local optima networks: Number partitioning as a case study, in: Evolutionary Computation in Combinatorial Optimization, (EVOCOP), Springer, Cham. pp. 233–248.

[28] Ochoa, G., Verel, S., Daolio, F., Tomassini, M., 2014. Local Optima Networks: A New Model of Combinatorial Fitness Landscapes. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 233–262. doi:10.1007/978-3-642-41888-4_9.

[29] Oliveira, S., Hussin, M.S., Roli, A., Dorigo, M., Stützle, T., 2017. Analysis of the population-based ant colony optimization algorithm for the tsp and the qap, in: 2017 IEEE congress on evolutionary computation (CEC), IEEE. pp. 1734–1741.

[30] Oliveira, S.M., Bezerra, L.C.T., Stützle, T., Dorigo, M., Wanner, E.F., de Souza, S.R., 2021. A computational study on ant colony optimization for the traveling salesman problem with dynamic demands. Comput. Oper. Res. 135, 105359. URL: https://doi.org/10.1016/j.cor.2021.105359, doi:10.1016/J.COR.2021.105359.

[31] Pitakaso, R., Almeder, C., Doerner, K.F., Hartl, R.F., 2007. A max-min ant system for unconstrained multi-level lot-sizing problems. Computers & Operations Research 34, 2533–2552.

[32] Pitzer, E., Affenzeller, M., 2012. A comprehensive survey on fitness landscape analysis, in: Fodor, J.C., Klempous, R.and Suárez, C.P. (Eds.), Recent Advances in Intelligent Engineering Systems. Springer, Berlin, Heidelberg. volume 378 of *Studies in Computational Intelligence*, pp. 161–191. doi:10.1007/978-3-642-23229-9_8.

[33] Potgieter, I., Cleghorn, C.W., Bosman, A.S., 2022. A local optima network analysis of the feedforward neural architecture space, in: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.

[34] Skinderowicz, R., 2020. Implementing a gpu-based parallel max–min ant system. Future Generation Computer Systems 106, 277–295.

[35] Stützle, T., Hoos, H.H., 2000. Maxmin ant system. Future Generation Computer Systems 16, 889 – 914. doi:10.1016/S0167-739X(00)00043-1.

[36] Tang, J., Ma, Y., Guan, J., Yan, C., 2013. A max–min ant system for the split delivery weighted vehicle routing problem. Expert Systems with Applications 40, 7468–7477.

[37] Teixeira, M.C., Pappa, G.L., 2022. Understanding automl search spaces with local optima networks, in: Proceedings of the genetic and evolutionary computation conference, pp. 449–457.

[38] Verel, S., Daolio, F., Ochoa, G., Tomassini, M., 2011. Local optima networks with escape edges, in: International Conference on Artificial Evolution (Evolution Artificielle), Springer. pp. 49–60.

[39] Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440 EP –. URL: http://dx.doi.org/10.1038/30918.

[40] Wei, X., Fan, J., Wang, T., Wang, Q., 2016. Efficient application scheduling in mobile cloud computing based on max–min ant system. Soft Computing 20, 2611–2625.

[41] Zhou, Y., Neri, F., Bai, R., 2024. Characterising deep learning loss landscapes with local optima networks, in: 2024 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. doi:10.1109/CEC60901.2024.10611772.