

APPLICATION OF MULTIDIMENSIONAL ACO PHEROMONE FOR MULTI-OBJECTIVE OPTIMIZATION

Grazyna Starzec¹, Mateusz Starzec¹, Malgorzata Zajecka-Bebel¹,
Aleksander Byrski^{1,*}, El-Ghazali Talbi², Danuta Rutkowska³

¹*Department of Computer Science, AGH University of Krakow,
al. Adama Mickiewicza 30, 30-059 Krakow, Poland*

²*University of Lille
42 rue Paul Duez 59000 Lille, France*

³*SAN University,
ul. Sienkiewicza 9, 90-113 Lodz, Poland*

**E-mail: olekb@agh.edu.pl*

Submitted: 5th March 2026; Accepted: 22nd May 2026

Abstract

In this work, we investigate an approach that enriches the information encoded within the ACO pheromone structure in order to enhance the performance of swarm optimization. While this concept has been extensively studied in the context of single-objective optimization, this paper provides a comprehensive evaluation of its extension to multi-objective optimization problems. The results demonstrate that this enhanced pheromone representation remains effective in multi-objective settings.

Keywords: metaheuristics, ant-colony optimization, multi-dimensional pheromone

1 Introduction

Algorithms based on Ant Colony Optimization (ACO) remain among the most effective methods for solving transport optimization problems [14, 22, 1, 7]. In recent years, our research has focused on extending these algorithms by enriching pheromone-based stigmergic communication through the introduction of multiple pheromone dimensions. The central idea is to increase the amount of information encoded in the pheromone deposited on each edge, allowing ants to exploit a richer set of signals than in the classical formulation.

The initial stage of this research addressed single-objective optimization problems and demonstrated that this multidimensional pheromone representation [26] can improve solution quality [27, 29]. More recently, we reported preliminary but promising results on applying the same concept to multi-objective optimization [28]. In this paper, we present a substantially broader experimental study of the multidimensional pheromone ACO framework applied to the bi-objective Traveling Salesman Problem (TSP) [11].

The remainder of this paper is organized as follows. First, we review the state of the art. Next, we present the proposed approach, followed by a de-

scription of the experimental setup and the corresponding results. Finally, we conclude the paper with a summary of the contributions and directions for future research.

2 State of the art

To provide readers with the necessary context for our research, we first present background information on two key areas: the ACO algorithm, including its application to multi-objective optimization problems, and the automatic configuration of algorithms.

2.1 Main concepts of ACO

The ACO algorithm was originally introduced by Marco Dorigo [9], with its first application being the Traveling Salesman Problem (TSP). It was later generalized into a metaheuristic framework by Dorigo and Caro [10]. This approach is inspired by the foraging behavior of real ant colonies and their methods of information exchange. The ACO can be applied to single-objective optimization problems that can be represented as graphs. In each iteration, the agents (ants) construct feasible solutions as paths on the graph. These solutions are then evaluated based on their objective function, and the resulting information is deposited in a global structure known as pheromone, which subsequently influences the agents' decisions in future iterations.

The solutions are constructed incrementally using two types of information: the heuristic value $\eta_{ij}(t)$, which reflects the attractiveness of a given edge from a purely local viewpoint (i.e., without considering a broader context), and the pheromone level $\tau_{ij}(t)$, which reflects how frequently this edge has appeared in low-cost solutions in previous iterations. The relative influence of these two components is controlled by exponents α and β , respectively. The probability of selecting a particular edge by an ant k is determined by comparing its value with the analogous quantities computed for all other feasible moves A_k . The precise probabilistic decision rule for iteration t , ant k , and the edge connecting the vertices i and j is given as follows:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{l \in A_k} \tau_{il}^\alpha(t) \eta_{il}^\beta(t)} & \text{if } j \in A_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In the original algorithm, after each iteration, all m generated solutions are used to update the pheromone trail. First, pheromone values are reduced by an evaporation mechanism governed by the parameter $\rho \in [0, 1)$. Then they are increased based on the solutions constructed, their costs L_k , and a constant Q . The following formulas present this procedure in detail:

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

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{-th ant uses edge } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Several variants of the ACO algorithm have since been developed as its successors. Among the most well-known, and the one we use as the baseline for our research and comparisons, is the Max-Min Ant System (MMAS) [32]. The MMAS introduces two key modifications to the original algorithm: the pheromone update is based solely on the best solution, and explicit lower and upper bounds on pheromone levels are applied at each iteration.

2.2 Multi-objective Ant Colony Optimization

Given the strong performance of the ACO algorithm on many graph-structured global optimization problems, researchers naturally began exploring its use in other contexts, including multi-objective optimization. In such problems, multiple objective functions must be considered, and improving a solution with respect to one objective often leads to its deterioration with respect to others. Consequently, there is usually no single optimal solution; instead, a set of solutions emerges in which no individual solution is clearly superior. A common method for selecting a subset of high-quality solutions—when the number of objectives is moderate (roughly up to a dozen), is to use dominance relations to identify the Pareto front. However, as the number of objectives increases, the solution space becomes subject

to the curse of dimensionality, and a weighted aggregation approach is often used to evaluate solution quality [23].

Given two solutions s and s^* with associated cost vectors $y = (y_1, y_2, \dots, y_k)$ and $y^* = (y_1^*, y_2^*, \dots, y_k^*)$, respectively, where each component represents the value of one of the k objective functions to be minimized, we say that solution s dominates solution s^* if:

$$\forall_i y_i \leq y_i^* \wedge y \neq y^* \quad (4)$$

The dominance relation between feasible solutions allows the construction of the Pareto front, which is composed of all solutions that are not dominated by any others. The fact that the algorithm's outcome is a set of solutions, rather than a single one, also makes it difficult to directly evaluate the algorithm's efficiency. Improving the quality of the solutions themselves is challenging, both with respect to implementation and evaluation. Moreover, in multi-objective settings, it can be crucial not only to find high-quality solutions but also to achieve a wide and uniform coverage of the Pareto front. To compare the performance of multi-objective algorithms, several metrics have been proposed, among which the hypervolume [34] and the epsilon indicator [33] are the most widely used.

The wide variety of methods for extending the ACO algorithm to multi-objective problems has resulted in numerous publications over the years that focus on comparing these approaches or organizing them into taxonomies [13, 3, 21, 12]. These works indicate that, aside from other less critical aspects, the proposed algorithms primarily differ in the following elements:

- number of colonies – single or multiple;
- number of pheromone matrices – single or multiple (typically one per objective);
- number of heuristic matrices – single or multiple (typically one per objective);
- solution construction strategy:
 - focused on a single objective;
 - dynamic – merges information from multiple objectives (e.g., assigning different objective weights to each ant);

- fixed – analogous to the dynamic case, but the combination of objectives is predetermined *a priori* for the whole run of the algorithm, usually using domain-specific knowledge;
- evaluation & pheromone update – may or may not rely on Pareto dominance, for example:
 - non-dominated solutions;
 - best solution for each individual objective;
 - best solution according to a weighted aggregation of objectives.

Since all these algorithms share the same underlying framework, with only certain parts adapted for the multi-objective context, the ACO algorithm can be viewed as a collection of interchangeable components [8, 4, 16, 2]. These components are generally compatible, resulting in thousands of possible algorithmic variations. To explore this design space, Lopez-Ibanez et al. [19] proposed an automated framework for testing different combinations of components specifically on problems with two objectives.

2.3 Automatic algorithm configuration

Metaheuristic optimization algorithms typically rely on multiple parameters, which makes objectively assessing their performance challenging. As the number of parameters controlling an algorithm increases, the corresponding search space grows exponentially, turning the task of finding suitable values into a complex hyper-optimization problem. In practice, only a limited number of configurations can be evaluated. A simple approach is to select a small set of candidate values for each parameter and exhaustively test all combinations. A more advanced, publicly available tool for automatic algorithm configuration is *irace* [17]. This software employs *iterated racing* procedures, including the iterated F-race algorithm, which has been successfully applied, among other methods [6, 20, 24], to the automatic configuration of multi-objective ACO algorithms [18, 19, 5]. Given an algorithm, its parameter definitions, and a set of problem instances, *irace* systematically explores the parameter space to identify configurations that are statistically superior to the alternatives.

3 Multidimensional pheromone

In this section, the multidimensional ACO pheromone in application to multi-objective optimization is detailed.

3.1 Pheromone representations in ACO

As discussed in the previous section, the most popular and widely adopted variants of the ACO share, among other aspects, two common features on which we would like to concentrate:

1. A similarly simple representation of the pheromone table: each edge is associated with a single scalar value (or, in the multi-objective case, at most one value per objective).
2. The number of solutions employed in the learning process: although not strictly fixed, these algorithms follow an elitist strategy to update the pheromone, typically relying on only a small subset of the generated feasible solutions (often even a single solution per iteration), those evaluated as the best (according to various evaluation criteria).

Our earlier work [30, 31] investigated the scalability of ACO-based algorithms in high-performance computing (HPC) environments and their ability to support extremely large ant colonies that generate hundreds or thousands of feasible solutions per iteration. The experiments indicated that incorporating a greater number of solutions into the pheromone update process improves the quality of the final results. However, the traditional pheromone model is too limited to effectively encode the richer information that can be extracted from more than just the top-performing solutions. To address this limitation, we aim to extend the pheromone structure so that it can store more information.

To capture more intricate information on the feasible solutions generated so far, we introduce a multidimensional pheromone representation. Instead of storing a single pheromone value $\tau_{ij}(t)$ for each edge (for each objective, when applicable), we propose using a vector $\tau_{ij}(t) = (\tau_{ij}^1(t), \dots, \tau_{ij}^K(t))$, where K is the total number of pheromone dimensions, and each component is independently updated by a different group of feasible solutions. By

increasing the dimensionality of the pheromone, we expect to be able to uncover, exploit, and benefit from indirect information about the problem's solution space that is embedded in a wider set of feasible solutions.

In the context of this research, it is also relevant to mention an alternative approach to extend the pheromone with an additional dimension, as presented in [15]. Although there is some conceptual overlap between that work and our proposal, it should be emphasized that the extension in [15] is tailored specifically to the single-objective Vehicle Routing Problem (VRP). In contrast, our aim is to design a problem-independent pheromone extension that is also suitable for multi-objective optimization.

3.2 Application to global optimization

As an initial step in investigating this research direction, we applied the proposed concept to the domain of single-objective optimization problems [27, 29]. These papers describe a modification of the MMAS algorithm in which the pheromone table is replaced by a multidimensional counterpart, and they report the experimental results used to assess this change and compare it with the original algorithm.

Introducing such a modification into the ACO algorithm requires suitable adaptations in the components related to the pheromone table, namely its update procedure and its role in the decision-making process.

The pheromone update consists of two mechanisms: evaporation and reinforcement. Both must be revised to reflect the fact that instead of a single value per edge, there are now multiple values. For evaporation, all coordinates of $\tau_{ij}(t)$ are multiplied by the evaporation parameter ρ , so each of them decreases by a certain fraction after every iteration. However, the reinforcement mechanism has to be reformulated in a more elaborate manner.

The primary purpose of the multidimensional pheromone is to extract information from a larger set of solutions generated in each iteration. To this end, we group the solutions used for pheromone updating according to their cost, and each k -th group then updates the corresponding k -th coordinate of the pheromone:

$$\tau_{ij}^k(t+1) = \rho \tau_{ij}^k(t) + \sum_{l=1}^{S_k} \Delta \tau_{ij}^{kl} \quad (5)$$

$$\Delta \tau_{ij}^{kl} = \begin{cases} Q & \text{if the } l\text{-th ant in the} \\ & k\text{-th group uses edge } (i,j) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Unlike the standard ACO formulation (see Eq. (3)) where the reinforcement parameter Q is divided by the cost of the solution, here a constant Q is used. This is because the solutions have already been partitioned into quality-based groups; applying a constant Q shifts the emphasis of learning from absolute cost to relative group membership.

We refer to the way solutions are grouped as the update type (UT) and propose two variants:

1. PartFromIndex (PFI) – solutions are sorted by cost and divided into groups containing the same number of elements.
2. PartFromEvaluation (PFE) – the cost interval defined by the minimum and maximum solution values is divided into equal subranges and the solutions with cost within each subrange form a group.

Equation (1) shows how the pheromone trail, represented as a single value $\tau_{ij}(t)$ for each edge, is used to construct solutions in the original ACO algorithm. If multiple values are stored instead of just one, they must be aggregated to a single value. We refer to this aggregation method as an aggregation type (AT). In our initial research for multi-objective optimization [28] we have used the following three variants that proved to be more effective in the area of single-objective optimization [27, 29]:

1. ExponentialRandom (ER) – a randomized selection of one of the available coordinates, where the probability of choosing a coordinate updated by worse solutions decreases exponentially, starting from a probability of 0.5 for the first coordinate (i.e., the one updated by the best solutions).
2. ExponentialRandomMax (ERM) – a modification of the Exponential Random variant; if the ER method selects the k -th coordinate of the pheromone, the ERM returns the maximum value among the first k coordinates.

3. PairingCombination (PC) – a multi-step procedure that aims to promote edges that appear more frequently in better solutions than in worse ones, and to demote edges that occur more often in worse solutions than in better ones.

Those three variants were used in the first part of the experiments – grid search (see section 4.2). However, in the second part of the experiments – automatic algorithm configuration (see section 4.3) – we have also included the two remaining aggregation types proposed in [29], namely,

1. Expected Combination (EC) – a multi-step procedure that stems from viewing the pheromone values as a type of probability distribution indicating how frequently a solution containing this edge belongs to the group represented by each value and then computing a kind of expected evaluation value for the edge on that basis.
2. Weighted Combination (WC) – the pheromone values are merged using a weighted product, where the first value is assigned a weight of 0.5, and each subsequent value receives a weight equal to half that of the preceding one (with the last weight adjusted so that all weights sum to 1).

3.3 Generalization for multi-objective optimization

The version originally used in the single-objective algorithm was adapted for the multi-objective context. We implemented a separate pheromone table for each optimization criterion, a common practice in multi-objective ant algorithms. After each iteration, each table was updated in a manner similar to the single-objective case. However, in the original setting, all solutions generated in the most recent iteration were used directly, a choice that is less straightforward in the multi-objective scenario. To address this, we introduced an additional configuration parameter — the solution selection strategy (SSS), which determines which solutions from the repository are used in the pheromone update phase. Three strategies were defined:

1. LastIterationAll (LIA) – all solutions from the most recent iteration (our current approach in the global optimization variant),

2. LastIterationPareto (LIP) – Pareto front formed by the solutions of the most recent iteration,
3. GlobalPareto (GP) – Pareto front formed by all solutions found so far by the algorithm.

Note that the first strategy always returns exactly as many solutions as there are ants. For the other two, the number of retrieved solutions is not fixed: for LastIterationPareto, it is at most the number of ants, whereas for GlobalPareto, it can go up to the total number of feasible solutions found so far.

The selected solutions were then sorted by cost with respect to each objective, and the best N (or all if fewer than N were available) were passed on for further processing. For the basic pheromone tables, all of these selected solutions contributed to updating the pheromone values on their corresponding edges. This yields an N -best-per-objective approach for the standard pheromone update. Observe that if N is at least as large as the number of solutions obtained from the repository, then all retrieved solutions are used in the update, and therefore all objective-specific pheromone tables are updated identically. However, this is not the usual situation and depends on the configuration of N , the chosen solution selection strategy, and the current state of the algorithm.

For the multidimensional pheromone, the procedure was modified accordingly. After extracting the solutions from the repository, sorting them by the cost for a given objective, and selecting the top N , we processed them in essentially the same way as in the global optimization algorithm. Thus, they were partitioned into groups and each group updated a particular segment of the pheromone table. As a result, even if all the solutions selected participate in the update, the tables corresponding to different objectives are modified differently. This improves both the learning behavior and the diversity of the search.

4 Experimental results

In this section, the conducted experiments and their results, with regard to the multidimensional ACO pheromones are presented.

4.1 Hypervolume measure for the algorithm's evaluation

The evaluation of the proposed algorithm was performed with the framework already used for previous experiments for global optimizations described in [27] and [29]. The framework was extended to solve the Multi-Objective Traveling Salesman Problem (MOTSP). The efficacy of various configurations of the algorithm was assessed based on a hypervolume measure [34]. For a bi-objective problem, the hypervolume is a surface determined by the Pareto front discovered by the algorithm and a reference point with both coordinates larger than those in the Pareto front elements.

Assessing the algorithm by the hypervolume measure makes it a maximization problem since the better the results (smaller costs), the larger the surface. In the case of TSP problems, the solutions costs are greater than 0. Therefore, in order to deal with a minimization problem in our experiments and keep the value in range $[0,1]$, we propose to use a hypervolume-based indicator I defined as follows:

$$I = 1 - \frac{H}{x \cdot y} \quad (7)$$

given the reference point with coordinates (x,y) and hypervolume surface H (see also Fig. 1). This indicator normalizes hypervolume with respect to the total reference area $(x \cdot y)$, effectively translating the maximization of the surface H into a minimization metric constrained to the interval $[0,1]$.

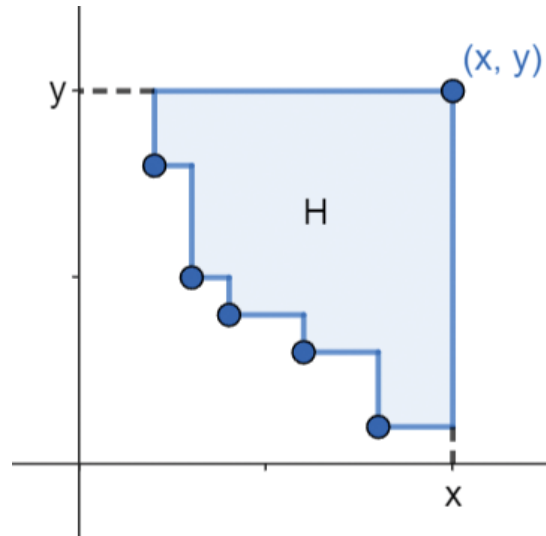


Figure 1. Hypervolume measure visualization for bi-objective optimization

4.2 Grid search with cross-validation

In the first phase of the experiments, we tested the algorithm on a bi-objective TSP instance derived from the kroA100 and kroB100 instances in TSPLIB [25], here referred to as kroAB100. In the bi-objective variant of TSP, the two objective functions correspond to minimizing the travel distances based on two distinct cost matrices (e.g., representing distance and time). Various algorithm configurations were evaluated, each configuration being run 20 times, keeping balance between statistical significance and the ability to execute the extensive parameter grid search. Some of these results were previously reported in [28]. Based on these findings, the selected configurations were then cross-validated on a range of additional instances, including problems with routes of up to 300 points.

Table 1 presents the parameters used for the algorithm configuration. The first part lists the parameters that were common for all versions of the algorithm and had a single value used. The second part is a list of parameters used for the configurations both with the Basic and the MultiDim pheromone tables with a set of possible values. The third part presents the additional parameters used only for the configurations with the MultiDim pheromone table, also with a set of possible values. Based on our learnings from single-objective experiments [29], we have decided to focus only on three aggregation types. The experiments included all possible combinations of the possible values of the parameters (grid search).

The top results obtained in the experiments are presented in Table 2. The tested configurations were sorted by the average value of the hypervolume-based indicator for the global Pareto front after the last iteration. Then the top 15 of them were listed, followed by the first two occurrences of configurations with the Basic pheromone table. It should be noted that the execution time of the algorithm using the MultiDim pheromone table did not exceed more than 30% above what was measured for the algorithm using the Basic pheromone table. In addition, since the configured number of iterations allows the algorithm to stagnate, running additional iterations would not change the results.

As we can see, the best-performing configurations use the MultiDim pheromone. The first two

configurations using the Basic pheromone table do not appear until positions 44 and 45. They both use a 0.05 pheromone delta value and 10 solutions for pheromone update. They differ in the SolutionSelectionStrategy – one of them selects updating solutions from all feasible solutions found in the iteration (LastIterationAll), and the other one only from the Pareto optimal solutions from the iteration (LastIterationPareto).

In fact, none of the top configurations uses the GlobalPareto strategy, which probably causes the algorithm to fall into premature convergence in some local optimum. Most of them use more than 10 solutions for the pheromone update process (up to a maximum of 100). A general tendency can be observed that usually selecting more solutions is combined with a larger pheromone size (e.g., pheromone size 4 is combined only with up to 30 updating solutions).

We can interpret it as follows: the more solutions we use in the learning process, the more capacity we need to extract useful information from them. The most common combination of update and aggregation type is ExponentialRandomMax with PartFromEvaluation. However, we can also see ExponentialRandom and PartFromIndex occurring – no sign of the PairingCombination, though, although it performed reasonably well for single-objective problems [29].

Figure 2 presents the average value from 20 repeats of the hypervolume-based indicator obtained by three selected configurations in subsequent iterations. Those three configurations are the best configuration and two top configurations with the Basic pheromone table (referenced by their ordinal number from Table 2). As we can see, the configurations using the Basic pheromone table obtained better results in the early iterations, but were outperformed by the algorithm with the MultiDim pheromone table around iterations 55 and 80 respectively. The MultiDim pheromone makes the learning process slower at the beginning and converges later, but allows for better final results.

From the best-performing configurations (see Table 2), we have chosen a subset with the aim of diversifying the parameters' values. Figure 3 is a box-and-whiskers chart for the hypervolume-based indicator for them, where the numbers are the ordinals from Table 2. The chart was already presented

Table 1. Grid search - configuration parameters and their values

Parameter	Value(s)
Common const parameters	
ants	100
alpha	2.0
beta	3.0
min pheromone	0.001
max pheromone	0.999
iterations	300
Common varying parameters	
update ants	1, 10, 30, 60, 100
pheromone delta – increase and expiration	0.05, 0.1
solution selection strategy	LastIterationAll LastIterationPareto GlobalPareto
MultiDim specific parameters	
pheromone size	4, 10, 20
update type	PartFromEvaluation PartFromIndex
aggregation type	ExponentialRandom ExponentialRandomMax PairingCombination

Table 2. Average and standard deviation of the hypervolume-based indicator value from 20 repeats (kroAB100). Top configurations all use the multi-dimensional pheromone with various combinations of other parameters. The basic pheromone table does not appear until the 44th position.

#	PhType	PhDelta	SSS	UpdAnts	PhSize	AT	UT	I Avg.	I Std Dev.
1	MultiDim	0.05	LIP	60	20	ERM	PFE	0.176323	0.001376
2	MultiDim	0.05	LIP	100	20	ERM	PFE	0.176564	0.000724
3	MultiDim	0.05	LIP	30	20	ERM	PFE	0.177278	0.001094
4	MultiDim	0.05	LIP	100	10	ERM	PFI	0.177436	0.001167
5	MultiDim	0.05	LIP	60	10	ERM	PFI	0.177562	0.001330
6	MultiDim	0.05	LIP	10	4	ERM	PFE	0.177688	0.000979
7	MultiDim	0.05	LIA	30	4	ER	PFE	0.177737	0.001105
8	MultiDim	0.05	LIA	60	20	ER	PFE	0.177915	0.001095
9	MultiDim	0.05	LIA	30	4	ERM	PFE	0.178075	0.000901
10	MultiDim	0.05	LIA	30	4	ER	PFI	0.178101	0.001129
11	MultiDim	0.05	LIP	30	10	ERM	PFE	0.178130	0.000942
12	MultiDim	0.1	LIP	60	20	ERM	PFE	0.178164	0.001481
13	MultiDim	0.05	LIA	30	4	ERM	PFI	0.178217	0.001105
14	MultiDim	0.05	LIA	60	20	ERM	PFE	0.178374	0.001141
15	MultiDim	0.05	LIP	60	10	ERM	PFE	0.178415	0.001084
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
44	Basic	0.05	LIP	10	-	-	-	0.180390	0.001380
45	Basic	0.05	LIA	10	-	-	-	0.180404	0.001023

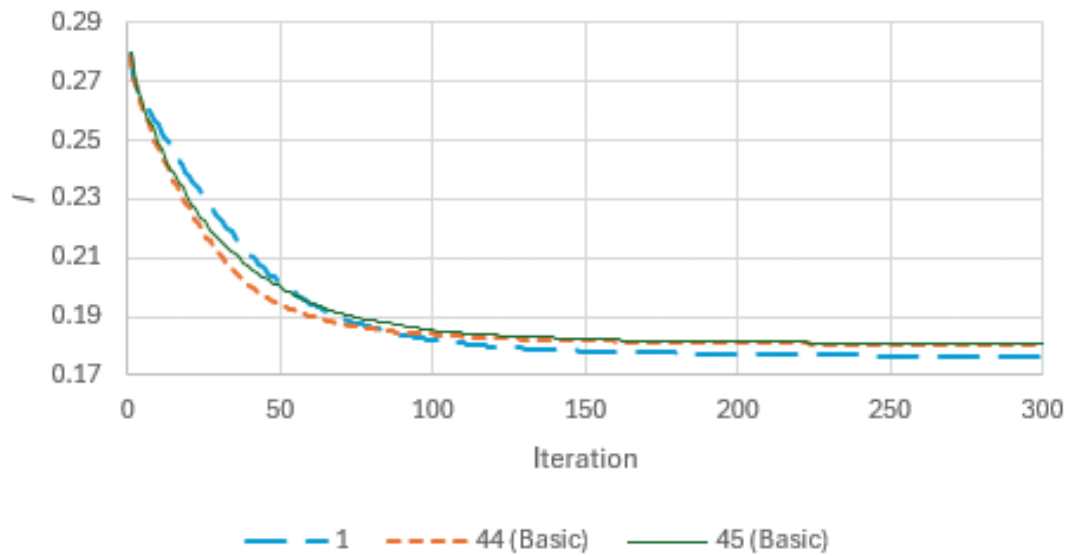


Figure 2. Hypervolume-based indicator value (average from 20 repeats) per iteration obtained for kroAB100 by selected configurations (ordinal numbers from Table 2). Top configuration using multidimensional pheromone (1) converges slower but gives better final results.

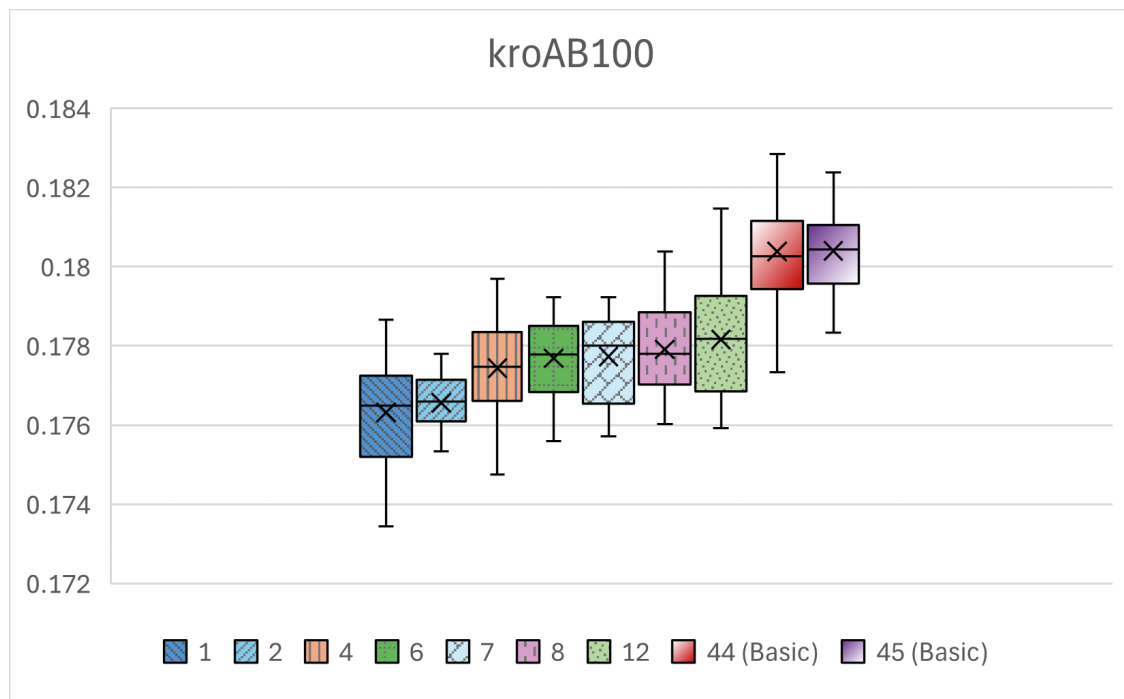


Figure 3. Hypervolume-based indicator value obtained by selected configurations for kroAB100 across 20 repeats. The numbers on the x-axis correspond to the configuration ordinals in Table 2. Configurations utilizing the MultiDim pheromone generally achieve better and more consistent performance compared to the Basic variants (44 and 45).

in [28], but we included it here again for easier reference for cross-validation experiments, in which we have used the selected subset of the configurations to solve a number of different bi-objective TSP instances.

Table 3. Average and standard deviation of the hypervolume-based indicator value from 20 repeats (kroCD100)

Configuration	<i>I</i> Avg.	<i>I</i> Std Dev.
1	0.173772	0.001024
2	0.173685	0.001100
4	0.174644	0.001180
6	0.174228	0.001452
7	0.175390	0.001031
8	0.175380	0.001206
12	0.175482	0.001350
44	0.176471	0.001533
45	0.177666	0.001300

In the first cross-validation experiment, the selected configurations were tested against an instance based on the kroC100 and kroD100 instances from TSPLIB, also referred to as kroCD100. The results obtained are presented similarly to those for kroAB100, listing the mean final hypervolume-based indicator value and its standard deviation in Table 3 and using a box-and-whiskers chart in Figure 4. Even though the configurations do not preserve order if sorted by the average value of the hypervolume-based indicator, they are fairly close to that. More importantly, configurations utilizing the MultiDim pheromone table consistently outperform those relying on the Basic pheromone table. This is especially noticeable if we look at both the value *I* and the repeatability of the results obtained.

Further cross-validation consisted in running the same set of selected configurations on several more instances, including some of the larger sizes: kroEF100, package-euclidAB100, kroAB150, kroAB200, and kroAB300, again 20 times each. Figure 5 contains box-and-whiskers graphs, analogous to the previous ones for kroAB100 and kroCD100 presenting the results of these experiments.

As we can see, as long as the instance size is 100, the configurations behave fairly consistently. Nevertheless, based on the other instances, it seems that the larger the instance, the more differences can

be seen in the performance of the algorithm depending on the values of the parameters. Two MultiDim configurations: 4 and 6 seem to handle the increasing size of the problem worst. Others do not exhibit such a tendency, at least in comparison to each other. However, even though the Basic configuration with LastIterationAll solution selection strategy (45) clearly remains the worst, the other Basic configuration (44) with LastIterationPareto relatively improved its performance with increasing size of the problem, finally taking the lead among all the tested configurations for the biggest one (kroAB300).

4.3 Automatic configuration of the algorithm

The second part of the experiments focused on an approach to algorithm evaluation based on an automated search of the parameter space, using the *irace* package [17]. For this purpose, we have used 4 bi-objective TSP instances during the training part, i.e., when looking for the best-performing configurations: kroAB100, kroCD100, package-euclidAB100, and kroAB300. The found configurations were later used to solve 5 other bi-objective TSP instances: kroEF100, kroAB150, kroAB200, euclidAB300, and euclidAB500, during the testing/evaluation part of the experiments.

Since the algorithm's behavior is primarily influenced by the type of pheromone table and, in the case of the MultiDim table, the aggregation type, we conducted six *irace* scenarios. The first one uses the Basic pheromone table and five use the MultiDim pheromone table—one for each aggregation type: ExponentialRandom, ExponentialRandomMax, WeightedCombination, PairingCombination, and ExpectedCombination. The remaining parameters were set as shown in Table 4. From each scenario, the best-performing configuration was selected and used for subsequent evaluation.

The settings of the *irace* software were mostly set to default with a few exceptions. The performance of the configurations in the grid search experiments depended on the solved problem (especially problem size), and variance of the results was fairly high compared to the differences in the configurations' performance. Therefore, we increased the number of instances evaluated before the first elimination test from default 5 to 12 (which also in-

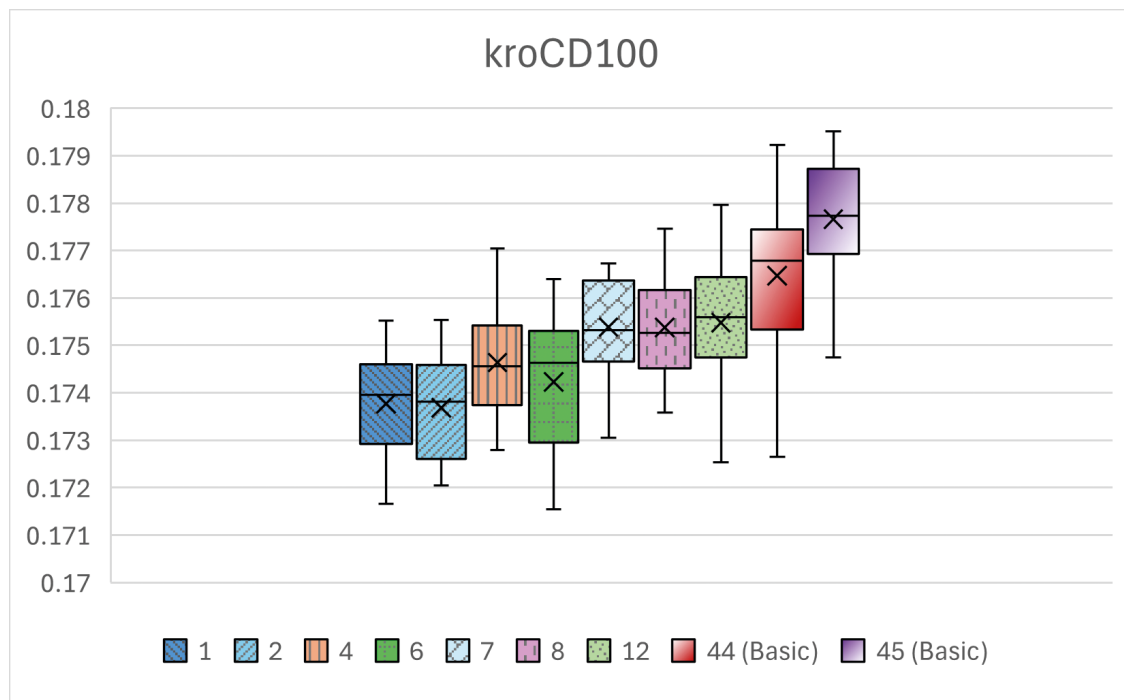


Figure 4. Hypervolume-based indicator value obtained by selected configurations (ordinal numbers from Table 2) for kroCD100 across 20 repeats. The results are fairly consistent with those for kroAB100 (see Fig. 3).

Table 4. irace - configuration parameters and their values

Parameter	Value(s)
Common const parameters	
ants	100
min pheromone	0.001
max pheromone	0.999
iterations	300
Common varying parameters	
alpha	real range (0.1, 5)
beta	real range (0.1, 5)
update ants	integer range (1, 100)
pheromone delta – increase and expiration	real range (0.04, 0.1)
solution selection strategy	LastIterationAll LastIterationPareto GlobalPareto
MultiDim specific parameters	
pheromone size	even integer range (2, 20)
update type	PartFromEvaluation PartFromIndex

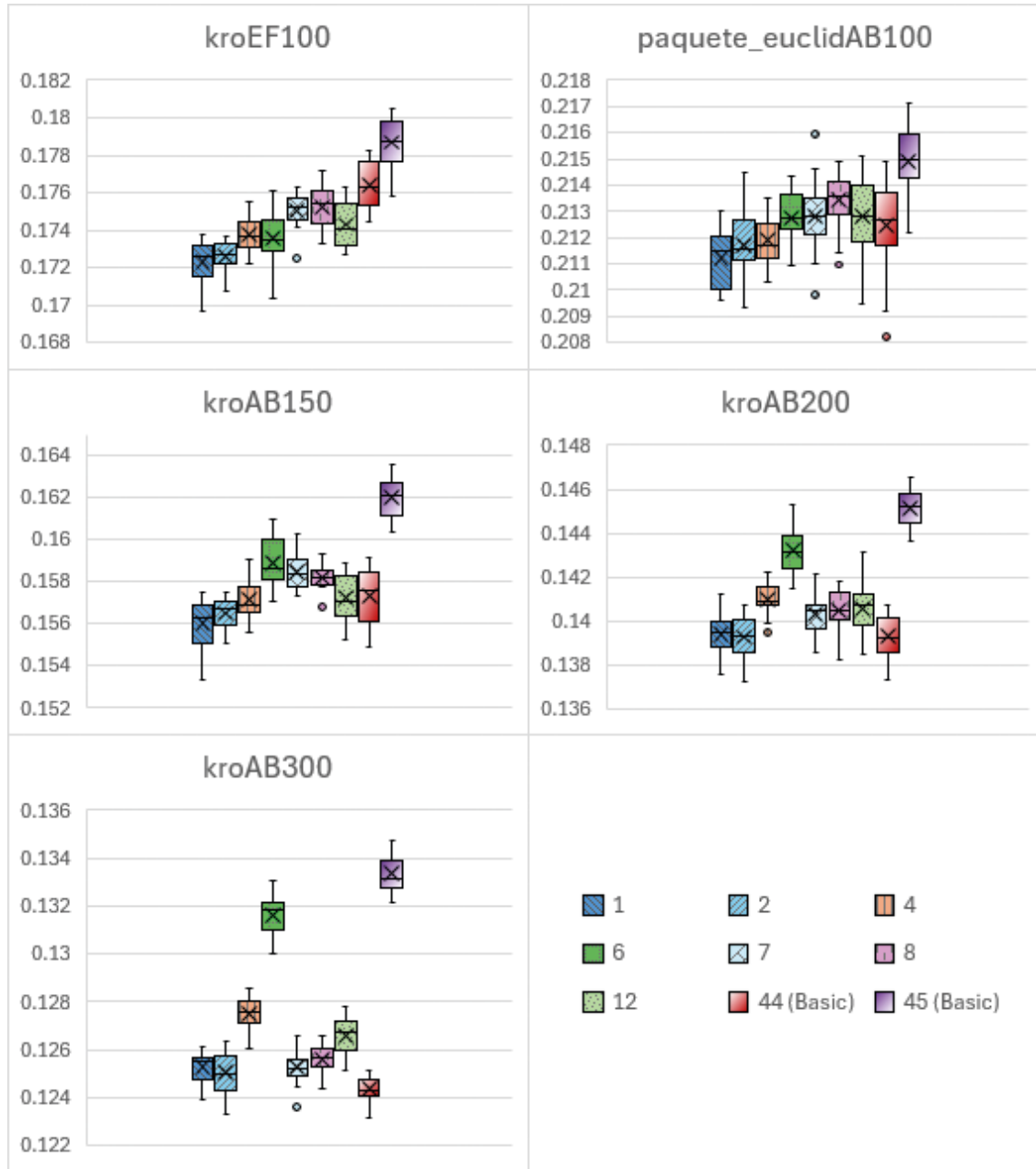


Figure 5. Hypervolume-based indicator value obtained by selected configurations (ordinal numbers from Table 2) for various MOTSP instances across 20 repeats. Relative effectiveness of the configurations starts to differ, especially with increasing problem size. One of the Basic configurations (44) takes the lead for the largest instance.

creased the number of configurations sampled and evaluated at each iteration to the same value). For the same reason, we also increased the number of instances evaluated between the elimination tests from default 1 to 4. The budget was set to 100,000 seconds for the Basic scenario and to 200,000 for the MultiDim scenarios (where the parameter space has higher dimensionality due to additional two parameters - pheromone size and update type).

Table 5 presents the best configurations found by *irace* for each scenario. In most cases, *irace* selected the LastIterationPareto solution selection strategy together with the PartFromEvaluation update type (if applicable). The size of the pheromone table was at least 12 and, as expected, MultiDim configurations used at least 35 solutions (and up to 94) for the pheromone update, extracting knowledge from much more than just 10 solutions (as selected for the Basic configuration).

Figures 6, 7 and 8 are box-and-whiskers graphs of the hypervolume-based indicator for the *irace*-selected configurations: Figure 6 for the training instances and Figures 7 and 8 for the testing instances. In order to allow for easier reference, comparison, and evaluation of the results obtained, the graphs also include scores for two representative configurations from the grid search experiments (section 4.2: 1 – the best among all configurations tested in those experiments (and in particular MultiDim configurations), and 44 – the best one with the Basic pheromone table.

As we can see, the ExponentialRandomMax aggregation type, which was the most promising based on the previous experiments, is mostly better than the Basic configuration (with the exception of *paquete-euclidAB100*, for which it was comparable). For smaller problem instances with size 100 the ExpectedCombination configuration is generally best (or comparable to ExponentialRandomMax) and consistently better than the Basic configuration. For larger instances, it performs worse; however, other MultiDim configurations start to overpower the Basic one, with the PairingCombination configuration taking the clear lead. Also, the WeightedCombination configuration becomes the second-best for instances larger than 150 points. As it seems, the larger the instance, the more evident the gap between the configurations becomes, sug-

gesting that the MultiDim pheromone gains more advantage being given a harder problem to solve.

Note that the above observations are consistent for both training and test instances. Another important remark to make is that the execution time of all configurations is fairly comparable. Based on the number of algorithm executions during *irace* runs, we can tell that even though the Basic configuration is the fastest. The execution time for the MultiDim configurations does not degrade too noticeably: ExponentialRandom is comparable, ExponentialRandomMax and PairingCombination are worse by about 10%, Expected by 15-20%, and Weighted by 20-25%.

Figure 9 presents the global Pareto front found for the *euclidAB500* test instance by the PairingCombination and Basic configurations selected by *irace*. The x and y coordinates of the points presented are solutions costs of objective functions determined by the *euclidA500* and *euclidB500* cost matrices respectively. The figure clearly shows that the algorithm with multidimensional pheromone table overpowers the other one in the central part of the Pareto front; i.e., it performs especially better in the most important task related to the multi-objective optimization as it finds significantly better solutions balanced between both optimization targets. Additionally, during this run, the multidimensional version took only around 4.5% more time than the other one.

5 Conclusions

In this paper, we have widely evaluated a new approach to solving multi-objective optimization problems using a modified version of the ACO algorithm. The concept of introducing a multidimensional pheromone to this algorithm was first presented in the single-objective area in [27] and [29], and was tested preliminarily in the multi-objective environment in [28]. It is based on the idea that using a single value per edge to store the collective information extracted from a created multitude of solutions might not be enough and thus proposes to store a set of values per edge instead.

We have broadly tested how this idea performs applied to multi-objective optimization problems. The experiments conducted on MOTSP instances

Table 5. *irace*-selected configurations for different scenarios

Scenario	α	β	PhDelta	SSS	Up.Ant.	PhSize	UT
ERM	1.6406	4.7953	0.0433	LIP	36	14	PFE
ER	1.369	4.8534	0.0711	LIP	94	20	PFE
PC	1.7993	4.9801	0.0573	LIP	75	14	PFE
EC	3.1165	4.9069	0.0805	GP	35	12	PFI
WC	0.6865	4.7997	0.0416	LIP	83	20	PFE
Basic	2.691	4.8517	0.0891	LIP	10	-	-

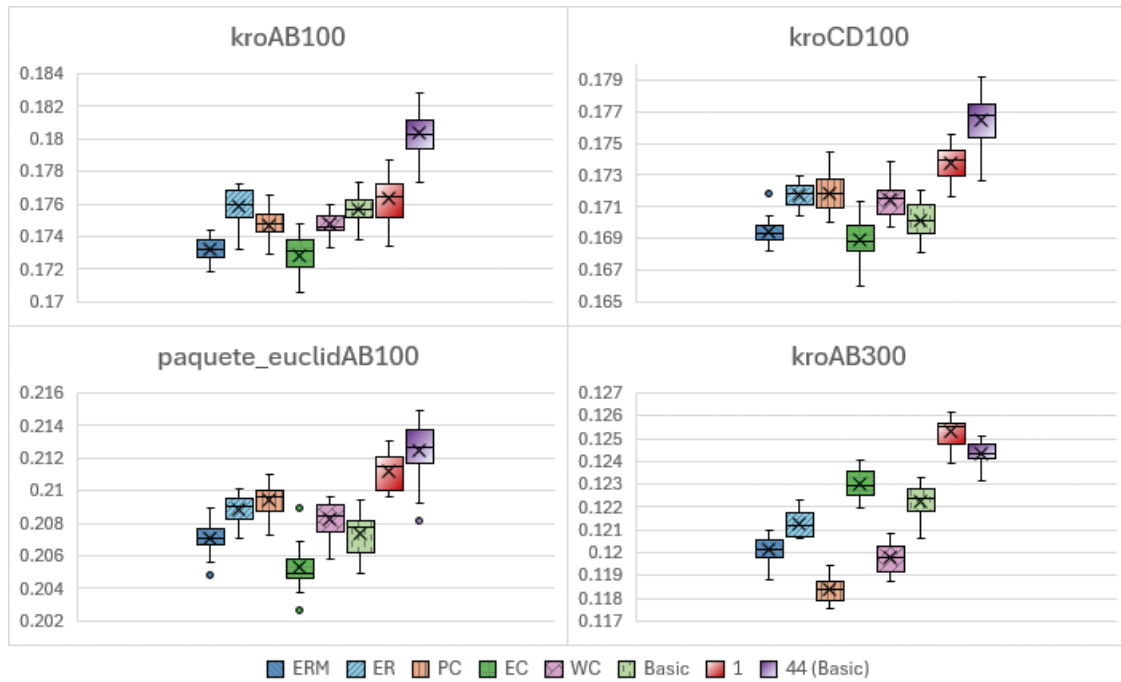


Figure 6. Hypervolume-based indicator value obtained by *irace*-selected configurations for training instances across 20 repeats. Top MultiDim (1) and Basic (44) configurations from section 4.2 are included for reference. The *irace* EC configuration is leading for the smaller instances and the PC configuration for the larger instance.

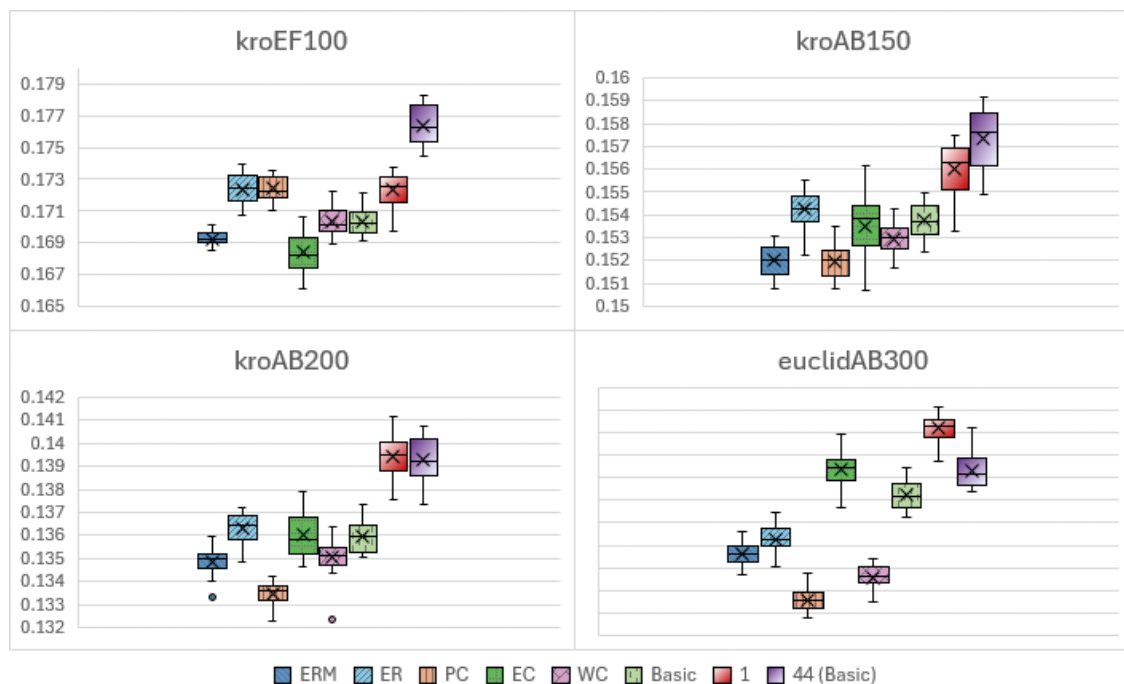


Figure 7. Hypervolume-based indicator value obtained by *irace*-selected configurations for test instances across 20 repeats. Top MultiDim (1) and Basic (44) configurations from section 4.2 are included for reference. Similarly as for the training instances, the *irace* PC configuration is taking the lead over EC and ERM with increasing problem size, followed by the WC configuration.

have first shown that selecting the algorithm's parameters based on the results of a single instance has a limited success rate. The top chosen configurations with multidimensional pheromone mostly or partly outperformed those using a basic pheromone table as long as the instance size remained fairly small, similar to the one used for initial evaluation. However, this superiority did not sustain as the solved problem increased in size.

Nevertheless, further experiments including automatic algorithm configuration proved that once properly configured, the multidimensional pheromone gives better results than a basic pheromone table. In fact exhibiting a bigger advantage with increasing problem size, introducing an acceptable degradation of execution time (e.g. only 4.5% for the best configuration and the largest instance). More importantly, the biggest difference in the obtained Pareto front is located in the central part of it. This means that it shows particularly strong performance in the key task of multi-objective optimization, discovering significantly superior solutions that balance both optimization targets.

Overall, this proves that the ACO algorithm can effectively benefit from extending the pheromone model storing the collective information in a more capacious structure and obtain better final results extracting knowledge from a larger number of feasible solutions.

As a perspective, we aim at designing and implementing a modification that would be more strongly coupled with multi-objective optimization, e.g., evaluating the solutions based on the Pareto front and the relation of domination instead of simply numerical value of individual objectives. Experiments with other families of multi-objective optimization problems and with more than two objectives will be investigated.

Acknowledgments

The research presented in this paper has received partial financing from the National Science Centre, Poland (NCN) project 2021/41/N/ST6/01776 (GS, MS) and from the funds assigned to AGH University of Krakow by Polish Ministry of Science and Higher Ed-

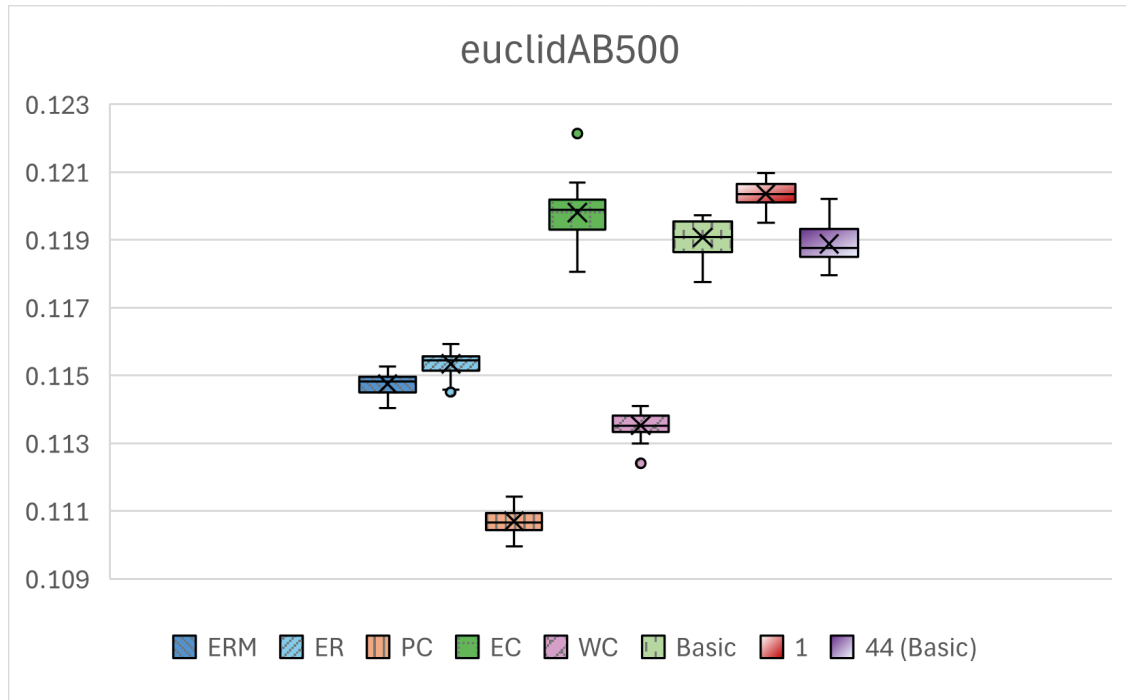


Figure 8. Hypervolume-based indicator value obtained by *irace*-selected configurations for the largest test instance across 20 repeats. Top MultiDim (1) and Basic (44) configurations from section 4.2 are included for reference. The differences in obtained results among the configurations become more evident. The *irace* PC and WC configurations are still leading while the *irace* Basic configuration falls behind, losing to most MultiDim configurations.

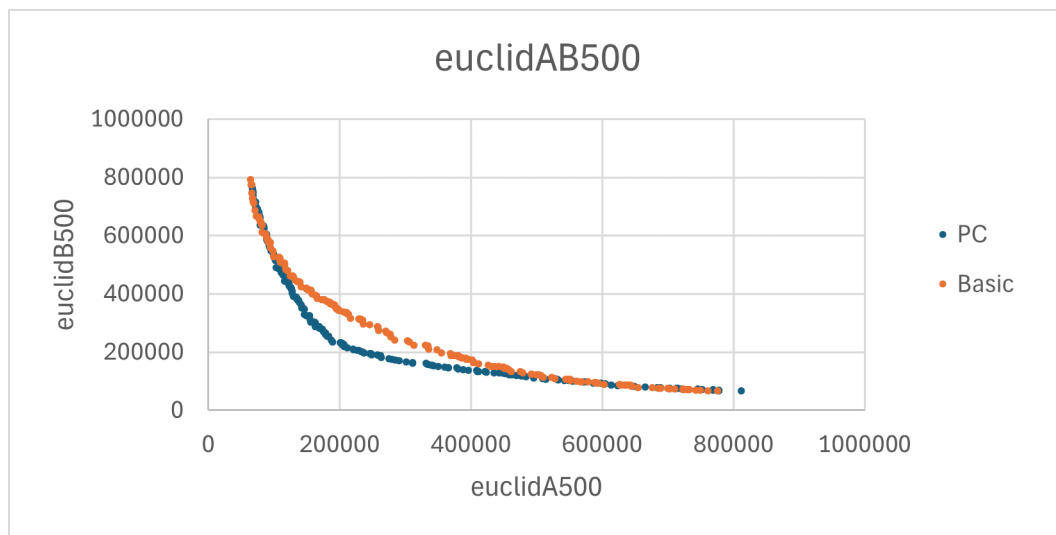


Figure 9. Pareto front found by *irace*-selected PC and Basic configurations for euclidAB500 test instance. PC configuration using multi-dimensional pheromone exhibits its advantage over the basic pheromone table in the central part of the Pareto front.

ucation (MZ). ARTIQ project: National Science Center: DEC-2021/01/2/ST6/00004, National Center for Research and Development, DWP/ARTIQI/426/2023 (AB).

References

- [1] Adrdor, R.: The power of intelligence emerging from swarms. *Computer Science* 26(1) (Apr 2025). DOI: 10.7494/csci.2025.26.1.6306, <https://journals.agh.edu.pl/csci/article/view/6306>
- [2] Alaya, I., Solnon, C., Ghedira, K.: Ant colony optimization for multi-objective optimization problems. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). vol. 1, pp. 450–457. IEEE (2007)
- [3] Angus, D., Woodward, C.: Multiple objective ant colony optimisation. *Swarm Intelligence* 3, 69–85 (2009)
- [4] Barán, B., Schaerer, M.: A multiobjective ant colony system for vehicle routing problem with time windows. In: *Applied Informatics*. pp. 97–102 (2003)
- [5] Bezerra, L.C., López-Ibáñez, M., Stützle, T.: Automatic generation of multi-objective ACO algorithms for the bi-objective knapsack. In: *International Conference on Swarm Intelligence*. pp. 37–48. Springer (2012)
- [6] van der Blom, K., Boonstra, S., Hofmeyer, H., Bäck, T., Emmerich, M.T.: Configuring advanced evolutionary algorithms for multicriteria building spatial design optimisation. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 1803–1810. IEEE (2017)
- [7] Brasileiro, I., Santos, I., Soares, A., Rabelo, R., Mazullo, F.: Ant colony optimization applied to the problem of choosing the best combination among m combinations of shortest paths in transparent optical networks. *Journal of Artificial Intelligence and Soft Computing Research* 6(4), 231–242 (2016). DOI: 10.1515/jaiscr-2016-0017, <https://doi.org/10.1515/jaiscr-2016-0017>
- [8] Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C.: Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research* 131(1), 79–99 (2004)
- [9] Dorigo, M.: Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano (1992)
- [10] Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406). vol. 2, pp. 1470–1477. IEEE (1999)
- [11] El-Samak, A.F., Ashour, W.: Optimization of traveling salesman problem using affinity propagation clustering and genetic algorithm. *Journal of Artificial Intelligence and Soft Computing Research* 5(4), 239–245 (2015). DOI: 10.1515/jaiscr-2015-0032, <https://doi.org/10.1515/jaiscr-2015-0032>
- [12] Falcón-Cardona, J.G., Leguizamón, G., Coello Coello, C.A., Castillo Tapia, M.G.: Multi-objective ant colony optimization: An updated review of approaches and applications. *Advances in Machine Learning for Big Data Analysis* pp. 1–32 (2022)
- [13] García-Martínez, C., Cordon, O., Herrera, F.: A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* 180(1), 116–148 (2007)
- [14] Green, S.: *Swarm Intelligence: Harnessing Collective Behavior for Innovation and Problem Solving*. Independently published (2024)
- [15] Guo, N., Qian, B., Na, J., Hu, R., Mao, J.L.: A three-dimensional ant colony optimization algorithm for multi-compartment vehicle routing problem considering carbon emissions. *Applied Soft Computing* 127, 109326 (2022)
- [16] Iredi, S., Merkle, D., Middendorf, M.: Bi-criterion optimization with multi colony ant algorithms. In: *International Conference on Evolutionary Multi-Criterion Optimization*. pp. 359–372. Springer (2001)
- [17] López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3, 43–58 (2016)
- [18] López-Ibáñez, M., Stützle, T.: Automatic configuration of multi-objective ACO algorithms. In: *International Conference on Swarm Intelligence*. pp. 95–106. Springer (2010)
- [19] Lopez-Ibanez, M., Stutzle, T.: The automatic design of multiobjective ant colony optimization algorithms. *IEEE Transactions on Evolutionary Computation* 16(6), 861–875 (2012)
- [20] Nebro, A.J., López-Ibáñez, M., Barba-González, C., García-Nieto, J.: Automatic configuration of NSGA-II with jMetal and irace. In: *Proceedings of*

- the Genetic and Evolutionary Computation Conference Companion. pp. 1374–1381 (2019)
- [21] Ning, J., Zhang, C., Sun, P., Feng, Y.: Comparative study of ant colony algorithms for multi-objective optimization. *Information* 10(1), 11 (2019)
- [22] Ostrowski, K., Starzec, M., Starzec, G.: The ant colony optimization algorithm applied in transport logistics. *Computer Science* 25(3) (Oct 2024). DOI: 10.7494/csci.2024.25.3.6360, <https://journals.agh.edu.pl/csci/article/view/6360>
- [23] Palakonda, V., Mallipeddi, R.: Pareto dominance-based algorithms with ranking methods for many-objective optimization. *IEEE Access* 5, 11043–11053 (2017)
- [24] Pérez Cáceres, L., Pagnozzi, F., Franzin, A., Stützle, T.: Automatic configuration of GCC using irace. In: *Artificial Evolution: 13th International Conference, Évolution Artificielle, EA 2017, Paris, France, October 25–27, 2017, Revised Selected Papers 13*. pp. 202–216. Springer (2018)
- [25] Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. *INFORMS J. Comput.* 3, 376–384 (1991), <https://api.semanticscholar.org/CorpusID:207225504>
- [26] Sekara, M., Kowalski, M., Byrski, A., Indurkha, B., Kisiel-Dorohinicki, M., Samson, D., Lenaerts, T.: Multi-pheromone ant colony optimization for socio-cognitive simulation purposes. *Procedia Computer Science* 51, 954–963 (2015). DOI: <https://doi.org/10.1016/j.procs.2015.05.234>, <https://www.sciencedirect.com/science/article/pii/S187705091501042X>, international Conference On Computational Science, ICCS 2015
- [27] Starzec, G., Starzec, M., Bandyopadhyay, S., Maulik, U., Rutkowski, L., Kisiel-Dorohinicki, M., Byrski, A.: Two-dimensional pheromone in ant colony optimization. In: *International Conference on Computational Collective Intelligence*. pp. 459–471 (2023)
- [28] Starzec, G., Starzec, M., Zajecka, M., Dobrowolski, G., Byrski, A.: Towards application of multi-dimensional aco pheromone for multi-criteria optimization. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) *Artificial Intelligence and Soft Computing*. pp. 271–279. Springer Nature Switzerland, Cham (2026)
- [29] Starzec, G., Starzec, M., Rutkowski, L., Kisiel-Dorohinicki, M., Byrski, A.: Ant colony optimization using two-dimensional pheromone for single-objective transport problems. *Journal of Computational Science* 79, 102308 (2024). DOI: 10.1016/j.jocs.2024.102308, <https://www.sciencedirect.com/science/article/pii/S1877750324001017>
- [30] Starzec, M., Starzec, G., Byrski, A., Turek, W.: Distributed ant colony optimization based on actor model. *Parallel Computing* 90, 102573 (2019)
- [31] Starzec, M., Starzec, G., Byrski, A., Turek, W., Pietak, K.: Desynchronization in distributed ant colony optimization in HPC environment. *Future Generation Computer Systems* 109, 125–133 (2020)
- [32] Stützle, T., Hoos, H.H.: Max–min ant system. *Future Generation Computer Systems* 16(8), 889–914 (2000)
- [33] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2), 117–132 (2003). DOI: 10.1109/TEVC.2003.810758
- [34] Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature — PPSN V*. pp. 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)



Grażyna Starzec received M.Sc. from the AGH University of Science and Technology in 2017. Currently she is finishing Ph.D. at the Faculty of Computer Science, AGH University of Krakow. Her research interests are simulation, distributed computing and metaheuristics with a stress on swarm-based metaheuristics.
<https://orcid.org/0000-0002-0813-798X>



Mateusz Starzec received M.Sc. from the AGH University of Krakow, Faculty of Computer Science, Electronics and Telecommunications in 2017 and PhD in 2021. His research interests are simulation, distributed computing and metaheuristics.
<https://orcid.org/0000-0001-8258-2443>



Malgorzata Zajecka-Bebel received Ph.D. in mathematics from Jagiellonian University in Krakow in 2013. Her research interests include machine learning, data science and metaheuristics.

<https://orcid.org/0009-0009-7914-2653>



Aleksander Byrski received Ph.D. in 2007, D.Sc. in 2013. Since 2020 he holds a title of professor. He works at AGH University of Krakow, Faculty of Computer Science. He is interested in parallel and distributed computing, metaheuristic computing and agent-based systems.

<https://orcid.org/0000-0001-6317-7012>



El-Ghazali Talbi received the Master and Ph.D degrees in Computer Science, both from the Institut National Polytechnique de Grenoble in France. Then he became an Associate Professor in Computer Sciences at the University of Lille (France). Since 2001, he is a full Professor at the University of Lille

and the head of the optimization team of the Computer Science laboratory (LIFL, CRISTAL). His current research interests are in the field of multi-objective optimization, parallel algorithms, metaheuristics, combinatorial optimization, cluster and grid computing, hybrid and cooperative optimization, and application to logistics/transportation and bio-medical.

<https://orcid.org/0000-0003-4549-1010>



Danuta Rutkowska received Ph.D. in automatics and D.Sc. in computer science, both from the Wroclaw University of Technology. In 2002 she was conferred the title of professor by the President of Poland. Her research interests are in the area of artificial intelligence, focusing on neural networks, fuzzy systems, genetic/evolutionary algorithms, and hybrid intelligent systems (neuro-fuzzy-genetic), as well as generative AI. After working for many years at the Czestochowa University of Technology, she is currently a professor in the Institute of Information Technology at the SAN University in Lodz, Poland. Since 2022 she has been serving as a vice rector for science at this university.

<https://orcid.org/0000-0003-0217-2589>