*Invited paper*

# A Many-objective Ant Colony Optimization applied to the Traveling Salesman Problem

Francisco Riveros[1], Néstor Benítez[1], Julio Paciello[1], and Benjamín Barán[1]

[1] *Facultad Politécnica, Universidad Nacional de Asunción, San Lorenzo, Paraguay*
{friverosn,nestorbdietrich}@gmail.com, {julio.paciello,bbaran}@pol.una.py

## Abstract

Evolutionary algorithms present performance drawbacks when applied to Many-objective Optimization Problems (MaOPs). In this work, a novel approach based on Ant Colony Optimization theory (ACO), denominated ACO $\lambda$ base-p algorithm, is proposed in order to handle Many-objective instances of the well-known Traveling Salesman Problem (TSP). The proposed algorithm was applied to several Many-objective TSP instances, verifying the quality of the experimental results using the Hypervolume metric. A comparison with other state-of-the-art Multi Objective ACO algorithms as MAS, M3AS and MOACS as well as NSGA2 evolutionary algorithm was made, verifying that the best experimental results were obtained when the proposed algorithm was used, proving a good applicability to MaOPs.

**Keywords:** Ant Colony Optimization, Traveling Salesman Problem, Many-objective optimization, Hypervolume, NSGA2

## 1. Introduction

Traditionally, Multi-objective Ant Colony Optimization (MOACO) algorithms are used to solve Multi-objective Optimization Problems (MOP), and they are considered as one of the best method for solving the well known Traveling Salesman Problem (TSP) [1].

Related works [2, 3, 4], use MOACO algorithms to effectively solve real life MOP's, for instances with 2 or 3 objective functions. However, this effectiveness is reduced when the number of objectives grows, as it happens with most evolutionary algorithms [5]. The main objective of this work is to improve the effectiveness of MOACO algorithms when applied to Many-objective Optimization Problems - MaOPs (typically, with more than 3 objective functions). In this context, a novel MOACO algorithm is proposed in this work to handle MaOPs.

## 2. Many-objective TSP

The well known TSP [6] can be represented as a fully-connected weighted graph $G = (N, A)$, where $N$ represents a set of $c = |N|$ nodes and $A$ a set of edges that interconnect the nodes in $N$. In a mono-objective approach, each edge has a unique cost function $d_{i,j}$ associated, typically representing the distance between nodes $i$ and $j$. TSP consists in finding the minimal cost Hamiltonian cycle, a tour that minimizes the traveled distance from a source node, traversing each node exactly once, and returning to the initial node [1]. To mathematically formulate the TSP, a dichotomous variable $x_{i,j}$ for all $(i, j) \in A$ can be considered, taking the value 1 if the edge $(i, j)$ belongs to the Hamiltonian cycle and 0 otherwise. Then, the TSP may be formulated as:

$$minimize \sum_{(i,j)\in A} d_{i,j}x_{i,j} \quad (1)$$

For the Many-objective TSP, $k$ cost functions ($k > 3$) are consired, having for each edge $(i, j)$ a set of $k$ cost functions or distances, $d^1_{i,j}, d^2_{i,j}..., d^k_{i,j}$. The problem consists in simultaneously minimizing the $k$ cost functions [7], i. e.:

$$min\ y = \begin{bmatrix} \sum_{(i,j)\in A} d^1_{i,j}x_{i,j} \\ . \\ . \\ . \\ \sum_{(i,j)\in A} d^k_{i,j}x_{i,j} \end{bmatrix} \quad (2)$$

where $y \in \mathbb{R}^k$.

## 3. MOACO algorithms

ACO algorithms [8] are inspired in the natural behaviour of real ant colonies to solve combinatorial optimization problems. They use an artificial ant colony, which can be described as a computational agent colony, working in cooperation and indirectly communicating through artificial pheromone trails. The ants build solutions traveling graph $G = (N, A)$, from an initial node. For each visited node, an ant selects the next city to be visit considering the visibility ($\eta_{i,j}$) and pheromone ($\tau_{i,j}$) parameters of each edge, applying a probabilistic policy given by the following equation:

$$p_{i,j} = \begin{cases} \frac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_{x\in J_i}\tau_{i,x}^{\alpha}\eta_{i,x}^{\beta}} & \text{if } j \in J_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $J_i$ represents all unvisited and reachable nodes from node $i$; while $\alpha$ and $\beta$ are parameters a priori defined, that weigh the relevance of the pheromones with respect to visibilities. The rule to update the pheromone is given by:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \rho\Delta\tau \quad (4)$$

where $\rho$ represents the pheromone evaporation coefficient $(0 < \rho < 1)$ and $\Delta\tau$ is the ammount of pheromones to be increased at every edge that belongs to a selected Hamiltonian tour.

A Multi-objective Optimization ACO (MOACO) is an extension of the ACO metaheuristic used to solve multi-objective optimization problems. In general, the equations are modified as follows [4]:

$$p_{i,j} = \begin{cases} \frac{\tau_{i,j}^{\alpha}(\eta_{i,j}^{1}...\eta_{i,j}^{k})^{\beta}}{\sum_{x\in J_i}\tau_{i,x}^{\alpha}(\eta_{i,x}^{1}...\eta_{i,x}^{k})^{\beta}} & \text{if } j \in J_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\Delta\tau = \frac{1}{\sum_{l=1}^{k}\sum_{(i,j)\in A} d_{i,j}^{l} x_{i,j}} \quad (6)$$

where $k$ is the number of objective functions. Algorithm 1 presents the pseudocode of a generic MOACO:

*Algorithm 1. Pseudocode of a generic MOACO.*

```
procedure MOACO()
 initializeParameters()
 while not stopCriteria()
  generation = generation + 1
  //m is the ant count
  for ant = 1 to m
   buildSolution()
   evaluateSolution()
   updateParetoSetAndPheromones() //(4)
  end for
 end while
return ParetoSet
end procedure

procedure buildSolution()
 sol = {}
 while UnvisitedStatesExist()
  next = getNextState() //(5)
  sol = sol U next
  markAsVisited(next)
  if (onlineUpdate)
   updateOnlinePheromones() //(4)
  end if
 end while
end procedure
```

Today, different MOACO implementations exist [9, 10]. Each implementation includes different pheromone update rules, number of pheromone values, number of ant colonies, etc. For this work, 3 state-of-the-art MOACO algorithms were selected: Multi-objective Ant System (MAS), Multi-objective Ant Colony System (MOACS) and Multi-objective Max-Min Ant System (M3AS), based on previous experimental results comparing several MOACOs, while solving different multiobjective TSPs [4]. In a MOACO context, each ant builds a solution using equation (5). In the specific case of MOACS, it also uses pseudo-random selection given by equation (7) and the transition probability given by equation (8), also used by M3AS and MAS.

$$j = \begin{cases} max_{j\in J_i}\{\tau_{i,j}^{\alpha}(\eta_{i,j}^{1\lambda_1}...\eta_{i,j}^{k\lambda_k})^{\beta}\} & \text{if } q < q_0 \\ \pi & \text{otherwise} \end{cases} \quad (7)$$

where $j$ is the next city to be visited while variable $\pi$ is calculated as:

$$\pi = \begin{cases} \frac{\tau_{i,j}^{\alpha}(\eta_{i,j}^{1\lambda_1}...\eta_{i,j}^{k\lambda_k})^{\beta}}{\sum_{x\in J_i}\tau_{i,x}^{\alpha}(\eta_{i,x}^{1\lambda_1}...\eta_{i,x}^{k\lambda_k})^{\beta}} & \text{if } j \in J_i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The variables $\lambda = [\lambda_1, .., \lambda_k]$ are parameters that weight the relative importance of each objective. Typically, each algorithm can use a different set of parameters $\lambda$ to guide the search of each ant to different regions of the objective space. This strategy was first applied successfuly by MOACS as explained in [4]. Several experimental evaluations using generated TSP instances of 2, 4, 8, and 10 objectives were performed in this work with all the implemented MOACO algorithms. The evaluations were executed simultaneously minimizing all $k$ objective functions. Based on previous analysis of MOACO algorithms applied to the multiobjective TSP, it could be observed that: (1) the Hypervolume metric [11] decreases when increasing the number of objectives; and (2) MOACS obtained a slightly better performance than the other MOACOs. For this reason, MOACS was selected to be adapted with the aim of improving its performance for MaOP TSP.

## 4. Performance metrics

The Hypervolume [11] was used as the evaluation metric for experimental results, considering that it is a widely adopted metric by researchers of the multi-objective optimization area [5, 12]. The Hypervolume considers the size of the dominated region in the objective space combining distance, distribution, and extension metrics in one empirical value. Given a set $Z = \{z_1, z_2, ..., z_R\}$ of $R$ solutions $z_i \in \mathbb{R}^k$, and a reference point $y_{ref} \in \mathbb{R}^k$, the hypervolume $H$ of set $Z$ is calculated as [11]:

$$H(Z) = H(Z, y_{ref}) = (\bigcup_{i=1}^{R} H(z_i)) \quad (9)$$

where $H(z_i)$ is the hypervolumen (area in 2D, volumen in 3D) of the hypercube defined by $z_i$ and $y_{ref}$ (selected as solution point worse than any $z_i$).

In other words, the hypervolume of a Pareto front approximation $Z$ is defined as the union of the portions (hypervolumes) limited by a chosen reference point $y_{ref} \in \mathbb{R}^k$, and each points $z_i \in \mathbb{R}^k$ of the approximated Pareto front. For example, considering a bi-objective problem with 2 known Pareto fronts $Z = \{a, b, c, d, e, f, g\}$, $Z' = \{a', b', c', d'\}$ and the reference point $y_{ref}$ (see Figure 1 ), if the hypervolume of $Z$ is greater than hypervolume of $Z'$, then it can be said that, Pareto front $Z$ is not worse than Pareto front of $Z'$ [13]; therefore, in practical application $Z$ is prefered over $Z'$ .

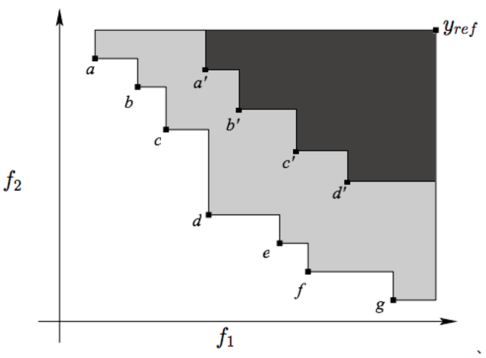


Figure 1: Hypervolume for Pareto fronts $Z, Z'$ and the reference point $y_{ref}$.

## 5. Proposed implementation

It can be seen in equations (7) and (8) that the parameters $\lambda$ are used for weighting the relative importance of each objective visibility. Typically, the sum of all parameters $\lambda$ is a constant. With this strategy, each ant may specialize into a specific region of the search space. For example, for bi-objective problems [2], values for $\lambda_1$ and $\lambda_2$ can be chosen using the following equation:

$$\lambda_2 = m - \lambda_1 + 1 \quad (10)$$

where $m$ represents the number of ants while $\lambda_1$ may take values between 1 and $m$. For a problem with $k$ objective functions $f_1, f_2, ..., f_k$, each ant uses a set of parameters $\lambda$ in such a way that for each $f_i$ there is a $\lambda_i$. In general, each $\lambda_i$ takes one of $m$ possible values; therefore, $k^m$ ants are needed to cover all possible permutations, resulting in a non-viable strategy when the number of objectives $k$ is large. To avoid this problem, this work proposes a new strategy denominated $\lambda$ base-p assignment.

### 5.1. $\lambda$ base-p assignment

This strategy applies a restriction to the possible values that $\lambda_i$ can take, restricted by a parameter $p \in \mathbb{N}$. The basic idea is that each ant chooses a random number $\tilde{n} \in [0, (p^k - 1)]$, that is converted to base $p$, obtaining a number with up to $k$ digits, where each digit corresponds to a value of $\lambda = [\lambda_1, \lambda_2, ..., \lambda_k]$, with $\lambda_i \in \mathbb{N}$, in the range $[0, (p-1)]$.

This work uses a value $p = 3$ for experimental porposes, so each $\lambda_i$ takes values: 0 (null weight), 1 (medium weight) or 2 (height weight). As an example, considering a problem with $k = 8$ objectives and $p = 3$; an ant may choose a random integer value $\tilde{n}$ between 0 and $(3^8 - 1)$. Let's suppose that $\tilde{n} = 4589$, a conversion of $\tilde{n}$ to a representation in base 3 gives the value 20021222; so, each digit represents the value of each $\lambda_i$, in this example, $\lambda_1 = 2, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 2, \lambda_5 = 1, \lambda_6 = 2, \lambda_7 = 2$ $y$ $\lambda_8 = 2$. It can be seen that the strategy of $\lambda$ base-p assignment is able to scale to a large number of objectives, as needed for MaOP, maintaining its simplicity and understandability.

## 6. Experimental Results

This section presents experimental results obtained after comparing the proposed MOACS base-p with the 3 already mentioned state-of-the-art algorithms (MOACS, M3AS, MAS) using the hypervolume as comparison metric. All the experiments were executed in a computer with the following characteristics: processor Genuine Intel 2.3 GHz, 64 bits architechture, 15 GB of RAM memory, and Ubuntu 12.04.2 LTS operating system.

Given that there is no Many-objective instances of the TSP already published in the specialized literature, random instances were generated for $k = 2, 4, 8$ and 10 objectives, taking care that no correlation between each pair of elements of cost matrices exceeds 0,1 using the Pearson correlation coefficient. Thus, the $k$ objectives are maintained reasonable contradictories or at least with a small correlation.

For the execution of the experiments, the original MOACS was updated to include the $\lambda$ base-p assignment, thus creating a new MOACO algorithm denominated in what follows MOACS base-p, representing the main contribution of this work. This variant was compared to the best MOACO algorithms that represent the state-of-the-art according to [4] (MOACS, M3AS and MAS), and also with respect to the NSGA2, considered as the referential algorithm for multi-objective optimization problems [5, 12, 14].

In order to analyze the evolution of the hypervolume, 10,000 generations was used as stop criterion, and the hypervolume was calculated for the generations 2.500, 5.000, 7.500 and 10.000.

Considering the parameters proposed in [4], the following values were used: $m = 10$ ants, $\alpha = 1$, $\beta = 2$, $\rho = 0{,}3$, $\tau_0 = 0{,}1$, $\tau_{max} = 0{,}9$, $\tau_{min} = 0{,}1$ and $q_0 = 0{,}5$. For the NSGA2, the following parameter were also used, based on [14]: a population size of 10, mutation probability of 0,8 and crossover probability of 0,98.

The presented results correspond to the average of 4 executions of each above mentioned algorithms, using TSP instances of 2, 4, 8 and 10 objectives, with 3 different problem size: 50, 75 and 100 nodes (cities).

Figure 2 shows the evolution of the hypervolume for the calculated Pareto fronts for each compared algorithm, when resolving an instance of 75 nodes, with 2 objectives (this instance is not considered as many-objective). It can be seen that MOACS base-p obtains a slightly larger hypervolume than the other alternatives along all generations, although MAS obtains a close result.

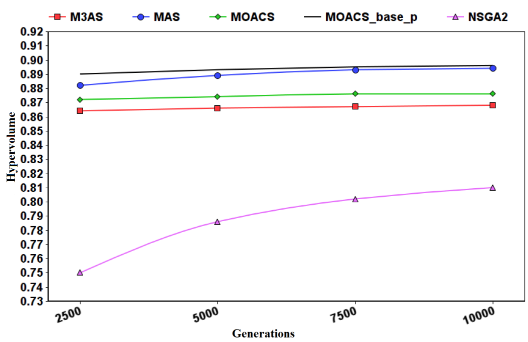


Figure 2: Obtained hypervolume solving instance with 2 objectives and 75 nodes

Figure 3 shows the evolution of the hypervolume for each compared algorithm, when solving an instance of 75 nodes, considering 4 objectives (considered already as a many-objective problem). It can be seen that the proposed algorithm is clearly better than the other algorithms along all generations, thus verifying the benefit of using the proposed $\lambda$ base-p assignment.

Figures 4 and 5 show the evolution of the hypervolume for each compared algorithm, when solving an instance of 75 nodes, considering 8 and 10 objectives respectively. Again, it can be seen that the proposed algorithm clearly outperforms to other algorithms along all generations, thus reinforcing the benefit of using the proposed $\lambda$ base-p assignment with increasing number of objectives.

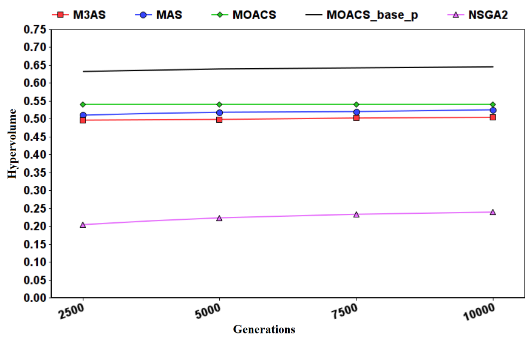


Figure 3: Obtained hypervolume solving instance with 4 objectives and 75 nodes

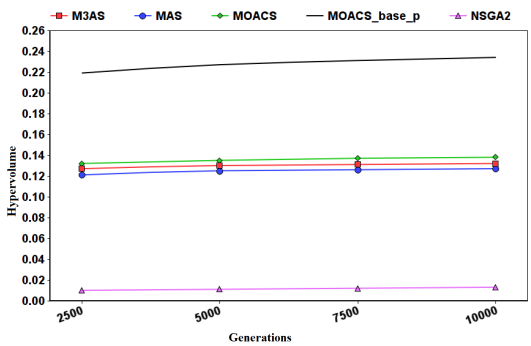


Figure 4: Obtained hypervolume solving instance with 8 objectives and 75 nodes

The same behaviour was observed when solving instances with 50 and 100 nodes. For instances with more than 2 objectives, MOACS base-p obtains a significantly better hypervolume than the other studied algorithms for all generations. Below, the experimental results for 50 nodes are presented in tables 1 to 4, summarizing the experimental results.

Table 1: Obtained hypervolume solving instance with 2 objectives and 50 nodes

| | Generations | | | |
|---|---|---|---|---|
| MOACO | 2.500 | 5.000 | 7.500 | 10.000 |
| M3AS | 0,893 | 0,895 | 0,896 | 0,897 |
| MAS | 0,903 | 0,911 | 0,912 | 0,912 |
| MOACS | 0,898 | 0,899 | 0,9 | 0,901 |
| MOACS_base_p | 0,915 | 0,916 | 0,917 | 0,917 |
| NSGA2 | 0,743 | 0,78 | 0,796 | 0,806 |

## 7. Conclusions and future works

This work presented a comparative analysis of MOACO algorithms applied to Many-objective Optimization Problems, for the first time in the

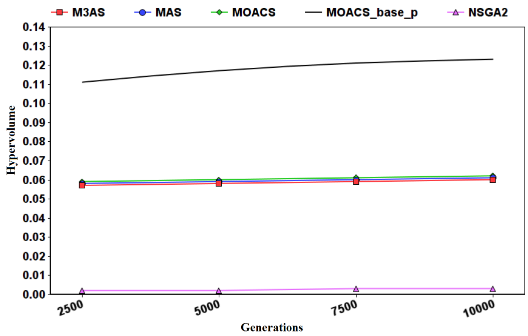


Figure 5: Obtained hypervolume solving instance with 10 objectives and 75 nodes

Table 2: Obtained hypervolume solving instance with 4 objectives and 50 nodes

| | Generations | | | |
|---|---|---|---|---|
| MOACO | 2.500 | 5.000 | 7.500 | 10.000 |
| M3AS | 0,643 | 0,648 | 0,65 | 0,651 |
| MAS | 0,633 | 0,638 | 0,645 | 0,656 |
| MOACS | 0,653 | 0,656 | 0,658 | 0,66 |
| MOACS_base_p | 0,743 | 0,748 | 0,751 | 0,753 |
| NSGA2 | 0,201 | 0,215 | 0,225 | 0,233 |

Table 3: Obtained hypervolume solving instance with 8 objectives and 50 nodes

| | Generations | | | |
|---|---|---|---|---|
| MOACO | 2.500 | 5.000 | 7.500 | 10.000 |
| M3AS | 0,289 | 0,293 | 0,295 | 0,296 |
| MAS | 0,284 | 0,288 | 0,292 | 0,294 |
| MOACS | 0,292 | 0,295 | 0,297 | 0,298 |
| MOACS_base_p | 0,411 | 0,421 | 0,426 | 0,431 |
| NSGA2 | 0,009 | 0,011 | 0,012 | 0,013 |

Table 4: Obtained hypervolume solving instance with 10 objectives and 50 nodes

| | Generations | | | |
|---|---|---|---|---|
| MOACO | 2.500 | 5.000 | 7.500 | 10.000 |
| M3AS | 0,178 | 0,181 | 0,182 | 0,183 |
| MAS | 0,181 | 0,186 | 0,188 | 0,19 |
| MOACS | 0,178 | 0,182 | 0,184 | 0,186 |
| MOACS_base_p | 0,279 | 0,289 | 0,294 | 0,298 |
| NSGA2 | 0,002 | 0,002 | 0,002 | 0,003 |

literature to the best of our knowledge. A new and simple variant of MOACS was proposed, denominated MOACS base-p, obtaining a better empirical performance when solving the MaOP TSP than other MOACO's (MOACS, M3AS, MAS) and the NSGA2, considered as the referential algorithm for multi-objective optimization in the specialized literature [5, 12]. All the results were compared considering the hypervolume as the performance evaluation metric.

The main contribution of this work consists in a new strategy for assigning the parameter $\lambda$ of MOACO algorithms for handling Many-objective Optimization Problems, denominated $\lambda$ base-p assignment. Experimental results confirm a superior performance of MOACS base-p when compared to other MOACO algorithms and the NSGA2, when solving instances from 2 to 10 objectives. As the number of objectives grows, the advantage of the MOACO base-p seems to increase.

As future work, experiments with a larger number of objectives can be performed, although the use of the Hypervolume metric should be reconsidered, given its asymptotical complexity that grows exponentially with the number of objectives [12]. Other extensions of this work can be: to apply the $\lambda$ base-p assignment strategy to other MOACO algorithms and to compare the proposed algorithm against other metaheuristics as Particle Swarm Optimization (PSO), solving other many-objective problems like the Quadratic Assignment Problem (QAP) or the Vehicle Routing Problem with Time Windows (VRPTW) to verify the performance of the proposed algorithm in other scenarios. Finally, other parameters of MOACO algorithms can also be adapted for a better performance in many-objective scenarios, for example, the use of groups of pheromones tables corresponding to groups of relatively similar objectives.

## References

[1] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The travelling salesman problem. Wiley, New York (1985)

[2] Barán, B., Schaerer, M.: A multiobjective Ant Colony System for Vehicle Routing Problems with Time Windows. Proc. Twenty first IASTED International Confe-rence on Applied Informatics, pp. 97–102. Insbruck, Austria (2003)

[3] Pinto, D., Barán, B.: Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach. In Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking, pp. 11-19, Cali, Colombia (2005)

[4] Paciello, J., Martínez, H., Lezcano, C., Barán, B.: Algoritmos de Optimización multiobjetivos basados en colonias de hormigas.

XXXII Conferencia Latinoamericana de Informática - CLEI'2006. Santiago de Chile (2006)

[5] vön Lücken, C., Barán, B., Brizuela, C.: A survey on multy-objective evolucionay algorithms for many-objective problems. Computational Optimization and Applications 58, no. 3, pp. 707–756, (2014)

[6] Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: The Travelling Salesman Problem. Wiley, New York (1985)

[7] Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In IEEE Congress on Evolutionary Computation CEC'2008, pp. 2419-2426 (2008)

[8] Barán, B., Gómez, O.: Reasons of ACO's Success in TSP, ANTS'2004 - Fourth International Workshop on Colony Optimization and Swarm Intelligence. Bruselas, Bélgica. (2004)

[9] Martínez, H., Paciello, J., Barán, B.: Equipo distribuido de algoritmos ACO multiobjetivos. VIII Argentine Symposium on Artificial Intelligence - ASAI, $35^{tz}$ Jornadas Argentinas de Informática e Investigación Operativa - 35 JAIIO. Mendoza, Argentina (2006)

[10] Paciello, J., Martínez, H., Barán, B.: Aplicación de un equipo de algoritmos ACO al VRPTW bi-objetivo. XIII Conferencia Latino-IberoAmericana de Investigación Operativa - CLAIO'2006. Montevideo, Uruguay (2006)

[11] Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Vol. 63. Ithaca: Shaker, (1999)

[12] Riquelme, N., vön Lücken, C., Barán, B.: Perfomance metrics in multi-objective optimization. XLI Latinoamerican Conference on Informatics - CLEI'2015. Arequipa Peru 2015

[13] While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. Evolutionary Computation, IEEE Transactions on, 10 no. 1, pp. 29–38 (2006)

[14] Optymalizacji, H. M., Kot, K., Pietraszko, A.: Heurystyczne metody wielokryterialnej optymalizacji kombinatorycznej. Akademia Górniczo-Hutnicza Thesis (2009)

## Acknowledgements

A gratitude to the Polytechnic Faculty of the National University of Asuncion for making easy the use of the research laboratories for realization of many experimental tests.