

## Single-objective and multi-objective optimization using the HUMANT algorithm

Marko Mladineo<sup>1,\*</sup>, Ivica Veža<sup>1</sup> and Nikola Gjeldum<sup>1</sup>

<sup>1</sup> Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture,  
University of Split  
Ruđera Boškovića 32, 21000 Split, Croatia  
E-mail: [marko.mladineo](mailto:marko.mladineo@fesb.hr), [iveza](mailto:iveza@fesb.hr), [ngjeldum](mailto:ngjeldum@fesb.hr)

**Abstract.** When facing a real world, optimization problems mainly become multi-objective i.e. they have several criteria of excellence. A multi-criteria problem submitted for multi-criteria evaluation is a complex problem, as usually there is no optimal solution, and no alternative is the best one according to all criteria. However, if a metaheuristic algorithm is combined with a Multi-Criteria Decision-Making method then, instead of submitting all solutions, only near-optimal solutions are submitted for multi-criteria evaluation, i.e. compared and ranked using *a priori* decision-maker preferences. It is called an *a priori* approach to multi-objective optimization. This paper presents this approach using a specially designed HUMANT (HUMANoid ANT) algorithm derived from Ant Colony Optimization and the PROMETHEE method. The preliminary results of this optimization algorithm are presented for the Single-Objective Traveling Salesman Problem (TSP), Shortest Path Problem (SPP) and the Multi-Objective Partner Selection Problem (PSP). Additionally, the multi-objective approach of the HUMANT algorithm to single-objective optimization problems is presented using the Shortest Path Problem (SPP).

**Key words:** single-objective optimization, multi-objective optimization, HUMANT algorithm, PROMETHEE method, ant colony optimization

Received: September 10, 2014; accepted: October 11, 2015; available online: October 31, 2015

DOI: 10.17535/cro.2015.0035

---

### 1. Introduction

For several years, the idea has existed of combining metaheuristic algorithms, such as the Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) or Simulated Annealing (SA) with the Multi-Criteria Decision-Making (MCDM) method (like AHP, ELECTRE, MAUT, PROMETHEE, or TOPSIS) to scalarize multi-objective problems into single-objective problems [27].

---

\* Corresponding author.

Generally, MCDM methods use some form of *a priori* knowledge concerning decision-maker preferences, such as criteria weights. Using these *a priori* decision-maker preferences, alternatives are compared and ranked i.e. they solve complex multi-criteria problems subject to a limited number of alternatives. However, if this limited number of alternatives becomes unlimited or very large ( $10^{20}$  alternatives or more), then practically applying these methods is impossible, as the computational time becomes too long. Consequently, these methods cannot be applied to complex multi-objective optimization problems because number of possible solutions (alternatives) is too large, and sometimes an infinite number of possible solutions exist.

Thus, if a metaheuristic algorithm is combined with a Multi-Criteria Decision-Making method, then only near-optimal solutions are submitted for multi-criteria evaluation i.e. compared and ranked using these *a priori* decision-maker preferences. This paper presents this particular approach using a specially designed HUMANT (HUMAnoid ANT) algorithm derived from a metaheuristic algorithm utilizing Ant Colony Optimization and the MCDM method PROMETHEE.

## 2. State-of-the-Art

Metaheuristic optimization deals with optimization problems in different areas and applications, from engineering design to economics and everyday life (telecommunication routing, trip planning, etc.) [32]. Most real-world optimizations are very complex and are accompanied by numerous constraints where such optimizations are solved using a highly efficient metaheuristic algorithm. Different objectives are often conflicting, so finding an optimal solution or even near-optimal solutions can be a hard task [27].

There are many applications of metaheuristic algorithms to different multi-objective optimization problems, but no known universal multi-objective metaheuristic optimization algorithm is able to solve most of the multi-objective optimization problems [27].

Generally, multi-objective optimization methods are classified based on three different approaches (Figure 1) [27]:

- ***A priori* approach** – decision-maker provides preferences before the optimization process.
- ***A posteriori* approach** – the optimization process determines a set of Pareto solutions, and then the decision-maker chooses a solution from the set of solutions provided by the algorithm.
- **Interactive approach** – there is progressive interaction between the decision-maker and the solver, i.e. knowledge gained during the optimization process helps the decision-maker to define the preferences.

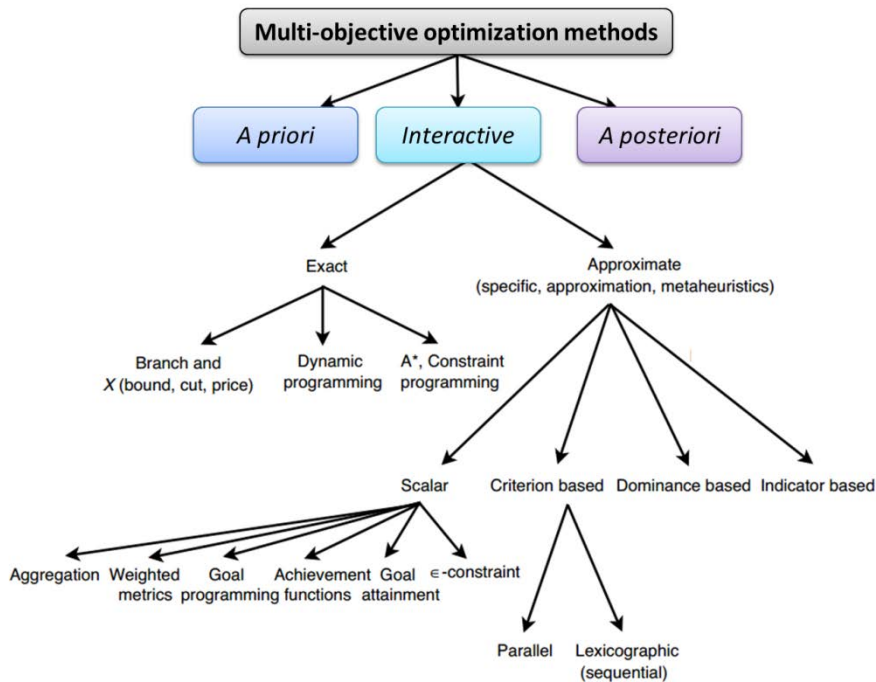


Figure 1: Classification of multi-objective optimization algorithms [27]

There are evolutionary *a posteriori* approach multi-objective optimization algorithms with excellent results in solving problems with two or three criteria, such as: Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4], S Metric Selection Evolutionary Multiobjective Optimisation Algorithm (SMS-EMOA) [1], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [34] and Scatter Tabu Search Procedure for Non-Linear Multiobjective Optimization (SSPMO) [19]. There are also some successful interactive approaches ([5], [15], and [21]) to multi-objective optimization.

However, in this paper the focus is on an *a priori* approach to multi-objective optimization, meaning that a particular single-objective metaheuristic algorithm is combined with a MCDM method, such as Multi Objective Ant Colony Optimization (MOACO) [8] or Progressive Multi-Objective Optimization (PMOO) [24].

Furthermore, when considering the *a priori* approach, any of the popular metaheuristic algorithms can be used for multi-objective optimization (Ant Colony Optimization (ACO) [4], Cuckoo Search (CS) [31], Firefly Algorithm (FA) [30], Genetic Algorithm (GA) [11], Particle Swarm Optimization (PSO) [13] and Simulated Annealing (SA) [14]), but has to be combined with a MCDM method in the appropriate manner.

Multi-Criteria (Multi-Attribute or Multi-Objective) Decision-Making (MCDM) consists of selection of “best” alternative, comparison and ranking of

alternatives, or comparison of alternatives with some referent points (sorting of alternatives). Generally, MCDM methods can be divided into following groups based on their characteristics: based on utility functions – MAUT [10], outranking methods – AHP [23], ELECTRE [22], PROMETHEE [2], TOPSIS [33], and interactive methods – VIMDA [20]. The main disadvantage of all these MCDM methods is that their scope is Multi-Criteria problems with finite number of alternatives, usually 10-30 alternatives, and very rare more than 100. So, in combination with metaheuristic algorithm, a limited number of near-optimal solutions found by metaheuristic algorithm should be submitted to some MCDM method. According to [3], few papers describe methods with *a priori* approach in which the user's preferences are integrated into a multi-objective metaheuristic algorithm.

### 3. Methodology

There is no single optimal solution to the multi-objective problem, but instead a set of solutions defined as the Pareto optimal solutions [27]. A solution is Pareto optimal if a given objective (criterion) cannot be improved without degrading other objectives (criteria). This set of solutions represents a compromise between different conflicting objectives (criteria). The main goal is to obtain the Pareto optimal set and, subsequently, the Pareto frontier. However, there can theoretically be a single solution to some combinatorial optimization problems. There is no best solution based on each criterion, but an overall best solution is based on decision-maker preferences, obtained using a MCDM method. Such approach is called the MCDM *a priori* approach to solving Multi-Objective optimization problems.

This *a priori* approach to Multi-Objective optimization combines the PROMETHEE method and Ant Colony Optimization (ACO) in a metaheuristic manner into a specialized single-objective and multi-objective metaheuristic algorithm to solve complex optimization problems.

#### 3.1. PROMETHEE method

The PROMETHEE method is well accepted by decision-makers, because it is comprehensive and has the ability to present results using simple ranking [2]. The input for the PROMETHEE method is a matrix consisting of a set of potential alternatives (actions)  $A$ , where each  $a$  element of  $A$  has its  $f(a)$  representing evaluation of a criteria. Each evaluation  $f_j(a_i)$  must be a real number. The PROMETHEE I method ranks actions in partial pre-order, with the following dominance flows [2], for leaving flow:

$$\phi^+(\alpha) = \frac{1}{n-1} \sum_{b \in A} \Pi(\alpha, b) \quad (1)$$

and for entering flow:

$$\phi^+(\alpha) = \frac{1}{n-1} \sum_{b \in A} \Pi(\alpha, b) \quad (2)$$

where  $a$  represents action from a set of actions  $A$ ,  $n$  is the number of actions and  $\Pi$  is the aggregated preference index defined for each pair of actions. A similar calculation is used to calculate entering flow. These two flows give a partial relation, and subsequently a net outranking flow is obtained using the PROMETHEE II method, which ranks the actions in total pre-order and calculates net flow. In terms of assessing priority, net outranking flow represents the synthetic parameter based on defined criteria and priorities among criteria. Usually, criteria are weighted using criteria weights  $w_j$  and the normal pondering technique.

### 3.2. Ant Colony Optimization

Ant Colony Optimization (ACO) is a population-based metaheuristic algorithm [4]. It is based on a set of virtual agents called artificial ants. They search for solutions to a given optimization problem. However, the optimization problem must be presented as a mathematical graph, and the ants build solutions by moving across the graph and leaving a pheromone trail. The solution to the construction process is stochastic and is affected by a pheromone update system. Hence, the main algorithm procedures are construction of solutions by ants and a pheromone update.

The ant construct solution is derived by traversing the graph from node to node until all nodes are visited or some condition has been met, i.e. until the solution has been constructed. Every single node is chosen using a particular probabilistic rule, like the one from the Ant System [4]:

$$(p_{ij})_k = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_k [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta} \quad (3)$$

where  $p_{ij}$  represents the probability that ant  $k$  will move from node  $i$  to node  $j$ , and where  $\eta$  is the reciprocal weight of an edge connecting two nodes,  $\tau$  is the level of pheromone along the edge of connecting two nodes,  $\alpha$  and  $\beta$  are algorithmic control parameters.

The pheromone level is updated after each iteration  $n$ , using rule [4]:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m (\Delta \tau_{ij})_k \quad (4)$$

where  $\tau_{ij}(t)$  represents the level of pheromone along an edge connecting nodes  $i$  and  $j$  at time  $t$ , and  $\rho$  represents the pheromone evaporation rate. Furthermore, the  $\Delta \tau_{ij}$  value depends on the quality of the solution [4]:

$$(\Delta \tau_{ij})_k = \frac{Q}{L_k} \quad (5)$$

where  $L_k$  represents the solution found by ant  $k$  (e.g. total length) and  $Q$  is a constant parameter.

Generally, at the beginning of a search, the pheromone level along each edge is set to a particular minimal value, and is increased over time along the edges connecting good solutions, evaporating on unvisited edges (bad solutions). Eventually, there is a convergence toward optimal or near-optimal solution [4]. The convergence of ACO was proven when modified into the so-called MAX-MIN ant system (MMAS) [25]. It is different from the original ACO, in that it limits the pheromone level along the edges using parameters  $\tau_{min}$  and  $\tau_{max}$ , and initially the pheromone level along each edge is set to a maximal value ( $\tau_{max}$ ). Accordingly, MMAS exhibits a convergence based on the evaporation of bad solutions, instead of encouraging good solutions. It achieves better results than all other ACO algorithms for numerous optimization problems [25].

### 3.3. Multi-Objective Ant Colony Optimization

Since the first application of ACO to multi-objective optimization [12], several approaches to Multi-Objective Ant Colony Optimization (MOACO) have been developed [9]:

- **Evaluation of objectives** – the decision-maker preferences are known *a priori* [8], hence the solutions can be scalarized, i.e. problem can be converted from multi-objective to single-objective [16]. This is the same approach as the generic *a priori* approach to multi-objective optimization.
- **One or several pheromone matrices** – each objective is represented by its own pheromone matrix in order to retain as much information as possible on each criterion. Finally, scalarization must be carried out in order to compare solutions.
- **One or several colonies** – this usually means that each colony constructs a solution only for a single objective. Finally, the Pareto-set of optimal solution is constructed. This is the same approach as the generic *a posteriori* approach to multi-objective optimization.
- **Different pheromone update approaches** – some specific rules for pheromone updates are based on a set of ants or set of colonies.

As has been already mentioned, this paper uses the *a priori* approach to multi-objective optimization, hence objectives are evaluated and solutions scalarized using the MCDM method.

### 3.4. The HUMANT algorithm

This paper presents an interesting approach to single-objective and multi-objective optimization based on the idea of combining Ant Colony Optimization and the multi-criteria PROMETHEE method for solving multi-objective optimization problems. This combination allows artificial ants to use multi-criteria decision-making, which is actually a human attribute.

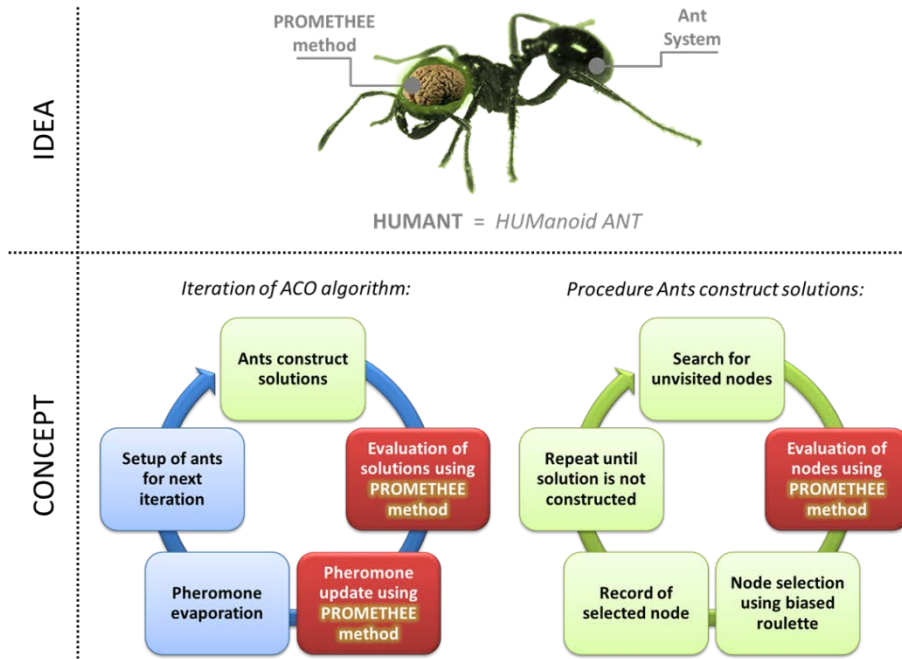


Figure 2: Idea and concept of HUMANT algorithm

Originally, the algorithm was designed to solve the multi-criteria Partner Selection Problem (PSP) and was called HUMANT (HUMANoid ANT) algorithm [18]. The idea and concept behind the HUMANT algorithm is presented in Figure 2. To realise presented concept, it is necessary to change the ACO equation (3) for evaluation of nodes and solutions, and the equation (4) for pheromone update. The PROMETHEE method must be implemented here. Also, the probabilistic rule used to evaluate nodes is changed in the procedure of constructing solution from (3) into (6):

$$p_{ij} = \frac{[\tau_{ij}(t)]^\alpha \cdot [\Phi'_{ij}]^\beta}{\sum_{k=1}^n [\tau_{ik}(t)]^\alpha \cdot [\Phi'_{ik}]^\beta} \tag{6}$$

where  $\Phi'_{ij}$  represents the modified PROMETHEE II score for edge connecting nodes  $i$  and  $j$ . It is calculated using following equation:

$$\Phi'_{ij} = \frac{1}{n-1} \frac{\sum_{k=1}^n (\pi(X_{ij}, X_{ik}) + (1 - \pi(X_{ik}, X_{ij})))}{2} \tag{7}$$

where  $X_{ij}$  represents the weight of edge connecting nodes  $i$  and  $j$ .  $\Phi'_{ij}$  is the PROMETHEE II score modified to produce the output in  $[0, 1]$  interval instead in  $[-1, 1]$  interval.

The evaluation of solutions needed for the pheromone update process is carried out using the following equation:

$$\Delta\tau_{ij} = 2 \left( \Phi'(x) \right) = 2 \left( \frac{\Pi(x, s^{id}) + (1 - \Pi(s^{id}, x))}{2} \right) \quad (8)$$

where  $x$  is the solution and  $s^{id}$  is the known optimal solution (for previously solved optimization problems). If the optimal solution is unknown, then the ideal solution is used. For instance, the ideal solution in the Shortest Path Problem (SPP) is the Euclidean distance between the origin and destination. In the Traveling Salesman Problem (TSP), the ideal solution is the sum of each row's minimal distance (or cost). Usually, this is not an achievable solution, i.e. it is better than the optimal solution.

Furthermore, the HUMANT algorithm was designed to use the MAX-MIN approach with pheromone trail limits set to  $\tau_{min} = 0$  and  $\tau_{max} = 1$ . If the constructed solution  $x$  is the same as the optimal (not ideal) solution  $s^{id}$ , then equation (8) will result in 1, i.e. the edges connecting the optimal solution will receive the maximal pheromone level. Hence, a convergence toward that solution is expected. If the optimal solution is unknown and the ideal solution is used, convergence will be a bit slower. A comparison of the HUMANT algorithm parameters and the MAX-MIN ant system parameters is shown in Table 1.

Parameter description	MAX-MIN ant system	HUMANT algorithm
Importance of pheromone trail along the edge	$\alpha$	$\alpha$
Importance of weight (cost) of edge	$\beta$	$\gamma$
Maximal level of pheromone trail	$\tau_{max}$	$\tau_{max}$
Minimal level of pheromone trail	$\tau_{min}$	$\tau_{min}$
Pheromone evaporation rate	$\rho$	$\rho$
Ideal solution	-	$s^{id}$

Table 1: Comparing the HUMANT algorithm parameters and the MAX-MIN ant system parameters.

Parameter  $\beta$  from equation (6) is missing in Table 1 due to the rule in the HUMANT algorithm rule:

$$\beta = n \quad (9)$$

where  $n$  represents the total number of evaluated nodes. This rule solves the problem associated with a non-dominating solution posed by PROMETHEE II methods, which occurs due to a large number of alternatives (more than 25-30 alternatives). One of the mandatory characteristics of the ACO is that one or few best alternatives are dominating in comparison with others. Controlling the importance of weight (cost) of edge is incorporated into the HUMANT algorithm using parameter  $\gamma$ . To simplify PROMETHEE II calculations, only a linear preference function is used, and parameter  $\gamma$  controls the linear preference



function parameters. The preference threshold  $p$  for the linear preference function is calculated using the following:

$$p = p_a \cdot Y \quad (10)$$

where  $p_a$  represents the automatically calculated linear preference threshold. Further, the indifference threshold  $q$  is calculated using the following:

$$q = \frac{q_a}{Y} \quad (11)$$

where  $q_a$  represents the automatically calculated linear indifference threshold. Automatic calculation of the parameters  $p_a$  and  $q_a$  means that  $q_a$  is a standard deviation of a set of mutual differences between actions, and  $p_a$  is  $q_a$  plus the average value of mutual differences between actions.

To conclude, the HUMANT algorithm is the MOACO algorithm with a MAX-MIN strategy, the global best ant pheromone update strategy, and a scalarization approach using simplified PROMETHEE II methods (i.e. using only the linear preference function where its parameters are automatically calculated). It was designed to incorporate a small number of parameters and uses fast calculation times. It can be applied to single-criteria and multi-criteria optimization problems.

Importantly, a similar approach was used to combine ACO and the PROMETHEE method by Eppe et al. [7]. However, they only presented a theoretical way of doing without experimental tests and practical algorithms. Their pheromone update strategy differs in the way that it does not include the problem of non-dominating solution of PROMETHEE II methods nor a solution to the problematic interval results of the PROMETHEE II method, i.e. the  $[-1, 1]$  interval.

## 4. Results

### 4.1. HUMANT algorithm and single-objective optimization

Traditionally, the Traveling Salesman Problem (TSP) was chosen to test experimentally the HUMANT algorithm for single-objective optimization problems. Problem instances *att48*, *eil51*, *kroA100* from TSPLIB were used and the HUMANT algorithm and its parameters were tested. A best combination of algorithm parameters was obtained:  $\alpha = 1$ ,  $\gamma = 1$ ,  $\rho = 0.4$ ,  $\tau_{min} = 0$  and  $\tau_{max} = 1$ . The algorithm was run 30 times with 500 iterations for each instance. The algorithm was written in *MATLAB* and tests were carried out using a 2.80 GHz CPU. The average run-time of 500 iterations for a 48-cities problem (*att48*) was 703 seconds. Code parallelization was not used.

The average result and average deviation from the optimum were compared with other ACO algorithms (Table 2). The results of other ACO algorithms are taken from [25].

Problem instance	HUMANT algorithm	MAX-MIN Ant System (MMAS)	Ant Colony System (ACS)	Ant System (AS)
48-cities problem ( <i>att48</i> )	10662 (0.32%)	-	-	-
51-cities problem ( <i>eil51</i> )	455.1 (6.83%)	427.6 (0.38%)	428.1 (0.49%)	437.3 (2.65%)
100-cities problem ( <i>kroA100</i> )	21358.3 (0.36%)	21320.3 (0.18%)	21420.0 (0.65%)	22471.4 (5.59%)

Table 2: A comparison of the HUMANT algorithm with other ACO algorithms with the aim of solving the Traveling Salesman Problem.

Table 2 is proof of a concept for the HUMANT algorithm, because the results are acceptable and interesting enough, that the HUMANT algorithm when applied to the 100-cities problem is better than the Ant Colony System or the original Ant System. The HUMANT algorithm had the worst performance for the 51-cities problem, because the problem instance *eil51* is a TSP problem incorporating numerous local optima where the HUMANT algorithm exhibits a very strong convergence.

#### 4.2. The HUMANT algorithm and a multi-objective approach to single-objective optimization

The Shortest Path Problem (SPP) was chosen to test experimentally the ability of the HUMANT algorithm to use a multi-objective approach to the single-objective optimization problem. The main aim of the SPP is to find shortest path between two nodes on a graph that represents a road network or something similar. It is not an NP-hard problem but there are known heuristic algorithms that can solve it, like Dijkstra's algorithm or its modified version, which incorporates certain constraints, and is called the A-star algorithm. However, if no constraints are used, the algorithm explores all edges or most of the edges of a graph. An algorithm that can find the shortest path without using constraints and explore only the nodes closest to the origin and destination nodes would be intelligent and efficient. However, such an algorithm can be designed only if a multi-objective approach is used. In solving the SPP, additional criteria (objectives) for identifying a solution could possibly be a deviation from of the Euclidean distance between the origin and destination. This is a bi-criteria approach to the SPP (Figure 3).

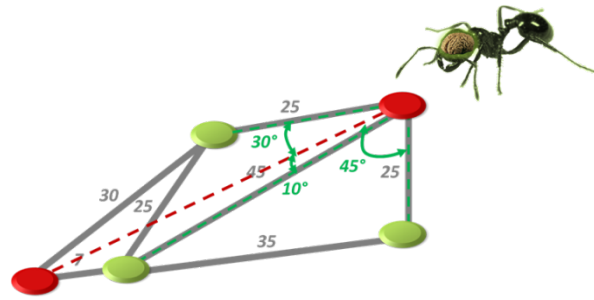


Figure 3: Multi-objective (bi-criteria) approach to the Shortest Path Problem

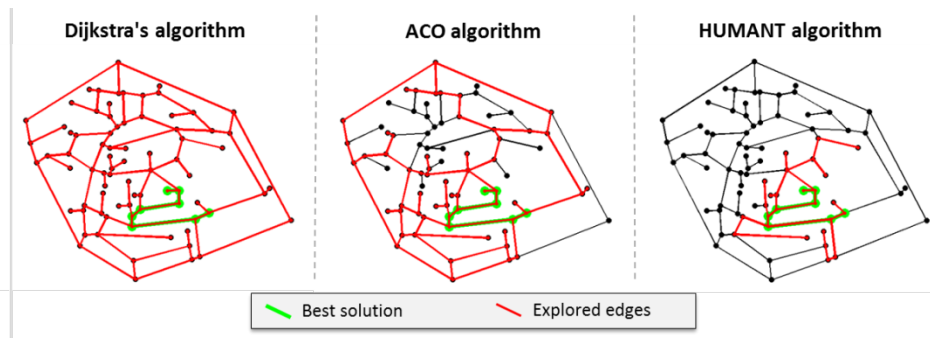


Figure 4: A comparison of the performance of Dijkstra's, ACO and the HUMANT algorithm in solving the Shortest Path Problem

In using a multi-objective approach to solving the single-objective SPP, the HUMANT algorithm was much more intelligent and efficient than the usual Dijkstra's algorithm and the ACO, as is shown in Figure 4.

#### 4.3. The HUMANT algorithm and multi-objective optimization

The Partner Selection Problem (PSP) was chosen to test experimentally the HUMANT algorithm for multi-objective optimization problems. The aim of the PSP is to select an optimal partner (enterprise) for each activity (technological process) that is part of a project or (manufacturing/ production) process. However, criteria for partner selection can be multiple: manufacturing cost [16], transportation cost [29], quality [17] or the business success of an enterprise [26], manufacturing time, etc.

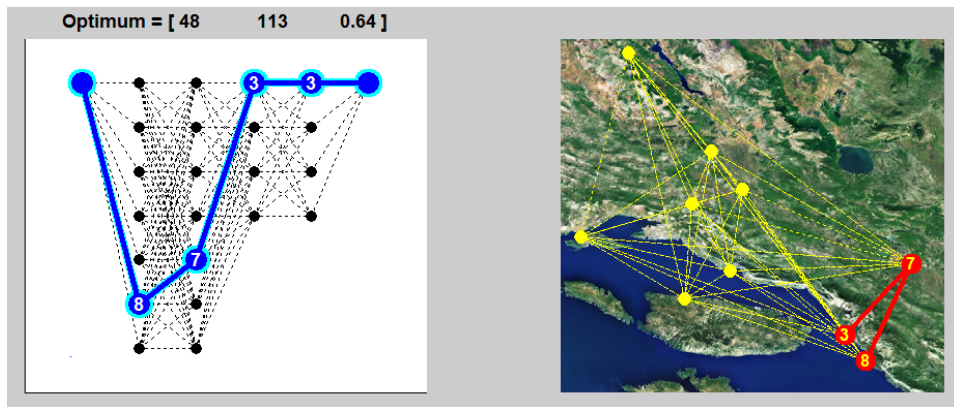


Figure 5: An optimal solution to the PSP instance, with 4 activities (technological processes) and 10 partners (enterprises), found by the HUMANT algorithm

For the problem taken from [28], the criteria are: manufacturing cost, transportation distances and manufacturing quality determined by the quality score of an enterprise. It is a PSP with 4 activities (technological processes) and 10 partners (enterprises) in total. The following parameters were used to solve this problem:  $\alpha = 1$ ,  $\gamma = 1$ ,  $\rho = 0.4$ ,  $\tau_{min} = 0$ ,  $\tau_{max} = 1$ ,  $w_{cost} = 0.15$ ,  $w_{transport} = 0.4$  and  $w_{quality} = 0.45$ . the HUMANT algorithm was run 30 times with 500 iterations each time it found an optimal solution to the problem: combination of enterprises [8, 7, 3, 3], resulting in costs of € 113,000, a 64% quality score, and 48 km of transport distance. According to [28], this is an optimal solution to this PSP instance (Figure 5).

Additional PSP instances were successfully solved using the HUMANT algorithm: a PSP instance from [16] and a PSP instance from [29]. This proved that the HUMANT algorithm can solve multi-objective optimization problems. Moreover, as already mentioned, the HUMANT algorithm was originally designed to solve these kinds of problems, i.e. multi-criteria PSP.

## 5. Conclusion

In this paper, a new metaheuristic algorithm for single-objective and multi-objective optimization is presented: namely, the HUMANT (HUMANoid ANT) algorithm. It is designed as a multi-objective optimization algorithm with an *a priori* approach, i.e. a combination of the metaheuristic algorithm for Ant Colony Optimization (ACO) and the multi-criteria decision-making method, PROMETHEE. Originally, it was designed to solve the multi-criteria Partner Selection Problem (PSP). This paper proves the concept of the HUMANT algorithm. Furthermore, it has presented how the HUMANT algorithm could be successful in solving different instances of the Traveling Salesman Problem (TSP). In solving TSP instances, on a 100-city problem provides even better results than the original ACO and AS. In addition, using the multi-criteria

approach for single-objective optimization, the HUMANT algorithm can become more intelligent and more efficient as demonstrated in solving the single-objective Shortest Path Problem (SPP). Finally, it is shown that HUMANT algorithm is able to solve a multi-criteria PSP instance with three objectives. However, additional research is needed to test the HUMANT algorithm on more TSP, SPP and PSP instances, and different multi-objective optimization problems, such as the Multi-Objective SPP, Multi-Objective Job Shop Scheduling problem, and the like. Finally, further research should cover differences between single-objective optimization (and multi-objective approach to single-objective optimization) with the multi-objective optimization highlighted.

## Acknowledgement

This work has been fully supported by the Croatian Science Foundation under project 1353 Innovative Smart Enterprise (INSENT).

## References

- [1] Beume, N., Naujoks, N. and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3), 1653–1669. doi:10.1016/j.ejor.2006.08.008
- [2] Brans, J. P., Mareschal, B. and Vincke, P. H. (1984). PROMETHEE - a new family of outranking methods in multicriteria analysis. *Operational Research IFORS 84*, Amsterdam.
- [3] Coello, C. A. C., Lamont, G. B. and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer.
- [4] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. doi:10.1109/4235.996017.
- [5] Deb, K. and Kumar, A. (2007). Interactive evolutionary multi-objective optimization and decision-making using reference direction method. *Proceedings of the 9th annual conference on Genetic and evolutionary computation GECCO '07*, 781–788.
- [6] Dorigo, M., Maniezzo, V. and Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26, 29–41. doi:10.1109/3477.484436.
- [7] Epe, S. (2009). Application of the Ant Colony Optimization Metaheuristic to multi-objective optimization problems. Technical report – ULB, Bruxelles.
- [8] Epe, S. (2009). Integrating the decision maker's preferences into Multi Objective Ant Colony Optimization. *Proceedings of the 2nd Doctoral Symposium on Engineering Stochastic Local Search Algorithms – SLS 2009*, Bruxelles.
- [9] Epe, S. and De Smet, Y. (2014). Approximating PROMETHEE II's net flow scores by piecewise linear value functions. *European Journal of Operational Research*, 233(3), 651–659. doi:10.1016/j.ejor.2013.08.043.

- [10] Farquhar, P. H. (1977). A survey of multiattribute utility theory and applications. *Multiple Criteria Decision Making - TIMS Studies in the Management Sciences*, 6, 59–90.
- [11] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- [12] Iredi, S., Merkle, D. and Middendorf, M. (1993). Bi-criterion optimization with multi colony ant algorithms. *Computer Science*, 359–372. doi:10.1007/3-540-44719-9\_25.
- [13] Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948.
- [14] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing, *Science*, 220, 671–680.
- [15] Luque, M., Caballero, R., Molina, J. and Ruiz, F. (2007). Equivalent information for multiobjective interactive procedures. *Management Science*, 53, 125–134. doi:10.1287/mnsc.1060.0595.
- [16] Mladineo, M., Veža, I. and Corkalo, A. (2011). Optimization of the selection of competence cells in regional production network. *Technical Gazette*, 18, 581–588.
- [17] Mladineo, M. and Veža, I. (2013). Ranking enterprises in terms of competences inside regional production network. *Croatian Operational Research Review (CRORR)*, 4, 65–75.
- [18] Mladineo, M. (2014). Phenomenological approach to the design of production networks. Ph.D. Thesis, University of Split, Split.
- [19] Molina, J., Laguna, L., Marti, R. and Caballero, R. (2007) SSPMO: A Scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, 19, 91–100. doi:10.1287/ijoc.1050.0149.
- [20] Olson, D. L. (1996). *Decision Aids for Selection Problems*. New York: Springer.
- [21] Phelps, S. and Koksalan M. (2003). An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12), 1726–1738. doi:10.1287/mnsc.49.12.1726.25117.
- [22] Roy, B. (1968). Classement et choix en présence de points de vue multiples (la méthode ELECTRE). (Ranking and choice in the presence of multiple points of view (the method ELECTRE). *La Revue d'Informatique et de Recherche Opérationnelle (The Journal of Computer Science and Operations Research) (RIRO)*, 8, 57–75.
- [23] Saaty, T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. New York: McGraw Hill.
- [24] Sörensen, K. and Springael, J. (2014). Progressive multi-objective optimization. *International Journal of Information Technology and Decision Making*, 13, 917–936. doi:10.1142/s0219622014500308.
- [25] Stützle, T. and Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16, 889–914.
- [26] Tadic, I. and Marasovic, B. (2013). Application of multicriteria decision making through financial, human resources and business process aspect in verification of companies' success. *Croatian Operational Research Review (CRORR)*, 4, 270–282.
- [27] Talbi, E. G. (2009). *Metaheuristics – From Design to Implementation*. New York: John Wiley & Sons.
- [28] Veža, I. and Mladineo, M. (2013). Sustainability through production networks. *Management and Production Engineering Review*, 4(4), 33–39. doi:10.2478/mper-2013-0035.

- [29] Wu, N., Mao, N. and Qian, Y. (1999). Approach to partner selection in agile manufacturing. *Journal of Intelligent Manufacturing*, 10, 519–529.
- [30] Yang, X. S. (2009). Firefly algorithms for multimodal optimization. 5th Symposium on Stochastic Algorithms, Foundation and Applications, LNCS, 5792, 169–178. doi:10.1007/978-3-642-04944-6\_14.
- [31] Yang, X. S. and Deb, S. (2010). Engineering optimization by cuckoo search, *Int. J. Math. Modelling Num. Optimisation*, 1, 330–343.
- [32] Yang, X. S. (2011). Metaheuristic Optimization. *Scholarpedia*, 6(8):11472.
- [33] Yu, C. X. and Wong, T. N. (2011). A TOPSIS-based pre-selection method supporting multiple products partner selection in a virtual enterprise. 21st International Conference on Production Research - Conference Proceedings, Stuttgart.
- [34] Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the Strength Pareto evolutionary algorithm for multiobjective optimization. Proceedings of the EUROGEN'2001, Athens.