A Proposal of Good Practice in the Formulation and Comparison of Meta-heuristics for Solving Routing Problems

E. Osaba, R. Carballedo, F. Diaz, E. Onieva, and A. Perallos

Deusto Institute of Technology (DeustoTech), University of Deusto,
Av. Universidades 24, Bilbao 48007, Spain
[e.osaba, roberto.carballedo,fernando.diaz, enrique.onieva, perallos]@deusto.es

Abstract. Researchers who investigate in any field related to computational algorithms (defining new algorithms or improving existing ones) find large difficulties when evaluating their work. Comparisons among different scientific works in this area is often difficult, due to the ambiguity or lack of detail in the presentation of the work or its results. In many cases, a replication of the work done by others is required, which means a waste of time and a delay in the research advances. After suffering this problem in many occasions, a simple procedure has been developed to use in the presentation of techniques and its results in the field of routing problems. In this paper this procedure is described in detail, and all the good practices to follow are introduced step by step. Although these good practices can be applied for any type of combinatorial optimization problem, the literature of this study is focused in routing problems. This field has been chosen due to its importance in the real world, and its great relevance in the literature.

Keywords: Meta-heuristics, Routing Problems, Combinatorial Optimization, Intelligent Transportation Systems, Good Practice Proposal

1 Introduction

Nowadays, the combinatorial optimization is a widely studied field in artificial intelligence and operations research, which is subject of a large number of articles and works every year [1, 2]. There are a lot of problems of this type, such as the Traveling Salesman Problem (TSP) [3] or the Vehicle Routing Problem (VRP) [4]. These problems have a great interest, thanks to their complexity, simple formulation and easiness of application to real world. Being NP-Hard [5], the scientific interest of the problems arising in combinatorial optimization make that many studies focus on their resolution, using a wide variety of algorithms.

The techniques used to solve this kind of problems can be divided into two groups: heuristics and meta-heuristics. A heuristic is an optimization technique that solves a problem using specific information and knowledge of that problem. This way, heuristics explore the space of feasible solutions intensifying the search in the most promising areas in order to achieve good optimization results quickly. However, they are only used to solve well-known problems of very simple formulation, such as standard TSP or the basic VRP, due to the difficulty of finding appropriate heuristics to real problems with complex objective functions and constraints.

On the other hand, a meta-heuristic is an optimization technique that solves a specific problem using only general information and knowledge common to all optimization problems. Meta-heuristics explore a larger area of the solution space in order to achieve good optimization results with independence of the problem. Thus, meta-heuristics are more appropriate to solve real problems with complex formulation, since they do not use any specific information of the problem in the exploration of the space of feasible solutions. Meta-heuristics can be applied in a wide range of fields, such as transport [6–9], medicine [10], or industry [11, 12]. Some of these algorithms are based on a single search, such as Simulated Annealing [13] and Tabu Search [14], and some others are based on a multiple search (population based algorithms), such as genetic algorithm [15, 16], ant colony systems [17], particle swarm optimization [18, 19], or imperialist methods [20]. Besides these, in the last years many new population techniques, and strategies, have been proposed [21–24]. Meta-heuristics also can be classified in trajectory algorithms and constructive algorithms. Trajectory algorithms start from an initial complete solution or an initial set of complete solutions that are modified until reaching a final optimal solution, while constructive algorithms start from a partial solution or a set of partial solutions that are built until reaching an optimal complete solution.

Using the TSP as example, many specific heuristics applied to it and its variants can be found in the literature. In [25], for example, three different contructive heuristics for the problem can be found. An example of a meta-heuristic for this problem can be found in [26]. In this work many variants of the genetic algorithm (GA) are explained, with many different crossover operators and mutation operators. Furthermore, in [27], a variable neighborhood heuristic for one variant of the TSP, the Traveling Deliveryman Problem, is presented.

Thus, heuristics focus on the resolution of problems of simple formulation, trying to reach its optimal solution. Meta-heuristics, however, can be applied to a wide variety of real problems whose complexity prevents developing appropriate heuristics. In this sense, the comparison among heuristics is more simple than the comparison among meta-heuristics, since they are implemented for a specific problem. No matter which the nature of heuristics is, or the parameters and features utilized, the best heuristic will be the one that obtains the best results in a reasonable time. Despite this, some problems arise when comparing heuristics, if the results are not displayed properly. This fact can be seen in [28].

The comparison between meta-heuristics is more complex, as many factors must be taken into account. This fact creates a lot of controversy and can lead to much confusion and bad practices. Despite this, there is still no methodology or procedure that helps researchers to describe and compare their meta-heuristics in a reliable manner. This way, the aim of this paper is to propose a procedure to facilitate an accurate comparison between different meta-heuristics. Although good practices proposed can be used for any type of combinatorial optimization problem, the literature of this study is focused in routing problems. This field has been chosen due to its importance in the real world, and its great relevance in the literature.

The structure of this paper is as follows. In the following section, the steps to follow in the description of the meta-heuristics in comparison are explained. In section 3, how the results should be accurately presented is explained. This paper ends with the conclusions of the study and its utility (section 4).

2 Good practices about the implementation and presentation of the meta-heuristic

In relation to the implementation of a meta-heuristic, a good practice would be as follows:

- A detailed specification of the problem constraints, classified in hard constraints and soft constraints.
- A detailed specification of the objective function, which should include the soft constraints if necessary.
- In the presentation of the work, the type of meta-heuristic technique being used must be precisely specified in the title or abstract of the paper, mentioning also heuristics, if used.
- A detailed description of all the operators used in the implementation. If they have been developed by the author, they have to be explained. If they are not originally developed by the author, they have to be correspondingly referenced. If the used operators are not described or referenced, the replicability of the results displayed is impossible.

The first step in the design of a technique for the resolution of routing problems is to define clearly the constraints and the objective function of the problem. Specifically, the objective function is an important issue related to the implementation of a technique. In problems like the TSP, this is not a problem, since the objective function is the distance of the route and the aim is to minimize it. For more complex problems, such as the Capacitated Vehicle Routing Problem (CVRP) [4] or Vehicle Routing Problem with Time Windows (VRPTW) [29], the function may vary depending on the objectives to be achieved. For the CVRP, for example, there are studies that prioritize the minimization of vehicles used [30], while others are focused on reducing the distance traveled [31]. For this reason, to avoid confusion, a good practice is to describe in detail the objective function. Otherwise, it is considered a bad practice.

When the problem and its characteristics have already been introduced, it is important to present adequately the meta-heuristic. One practice that should be avoided would be the confusing denomination of the techniques. An example of this confusing naming can be found in [32]. In this work, the authors present their approach to solve the heterogeneous fleet vehicle routing problems with two-dimensional loading constraints as a meta-heuristic, but the technique used is a simulated annealing based in a heuristic local search. This approach should have been described more precisely as a meta-heuristic in combination with a heuristic specific of the problem.

When the problem and its objective function is already defined and the type of meta-heuristic to develop is decided, the next step is to decide how it will be implemented and what kind of operators will be used. Although it seems simple, this fact could be controversial. As known, meta-heuristics use different types of operators to modify and transform the available solutions, in order to improve them. Thereby, the first point to be considered is the following: To test the optimization hability of a meta-heuristic to solve a routing problem, it is necessary to use neutral operators throughout the implementation. In other words, operators that use characteristics of the problem and optimize by themselves have to be avoided.

As an example of this fact, the initialization process of a GA can be mentioned. The most appropriate way to prove the optimization quality of a meta-heuristic is to use a random initialization process, instead of using optimizing initialization functions to create individuals, such as those proposed in [33] for the VRPTW. If any of these initialization functions is used, the individuals will pass through an optimization process before the execution of the core of the GA. Therefore, it may not be known exactly what the capacity of optimization of a meta-heuristic is when final results are obtained. In this case, it has to say that a heuristic has being implemented, because specific information of the problem is used.

Continuing with the GA and using the TSP as example, a heuristic crossover operator would be the Very Greddy Crossover (VGX) [34]. The VGX is an operator for the TSP that uses the distances between cities to generate the children. It is logical to think that using this operator the GA will get good results for the TSP, as the VGX makes by itself a small optimization on the resulting individuals. To implement a meta-heuristic, operators like Order Crossover [35], Half Crossover [36], Order Based Crossover [37] or Modified Order Crossover [38] should be chosen as the crossover function, since they are neutral operators. These operators only care to meet the constraints of the problem and they do not use any kind of information related to the problem.

Regarding this matter, the next point to consider is introduced. It should be avoided the comparisons in any work between meta-heuristic techniques with neutral operators and heuristic techniques with optimizing functions. Otherwise, the comparison would be unreliable, because of the different nature of the techniques. One example of this bad practice is shown in [39], where three techniques are compared solving the problem of clustering rectangles. In this work two of them are meta-heuristics, while the other one is a heuristic. Another example of this type of bad practice can be found in [40], which introduces a new heuristic crossover operator for the GA applied to the TSP, called Sequential Constructive Crossover (SCX). To check the quality of the new crossover operator, the results obtained by the SCX are compared with the results obtained by two GA, which use neutral crossover functions. Logically, the SCX gets much better results, but the comparison is not fair and valid. This same bad practice is performed in [41]. In that work, a new greedy mutation operator to solve the TSP is presented. To prove the quality of the new operator, the authors compare its performance with the one of 7 different mutation functions, being 6 of them neutral and only one greedy. In relation to the said above, the comparison with the 6 neutral function provides no relevant information, and it could be considered a bad practice.

It should be borne in mind that to make a completely reliable comparison between two meta-heuristics, is mandatory the use the same operators and parameters for both, as long as possible. If it is not possible, operators used in both techniques must have similar characteristics. For this reason, the points explained above are of vital importance, both to make the results easily reusable in other studies, and to give credibility to a comparison performed in a work.

3 Good practices showing of the results

Having described properly the characteristics of a meta-heuristic, it is appropriate to show the results it can get. This is a very important step, because according to the form that the results are presented, their replicability can increase, and other researchers can use them to compare their techniques. This is a very important issue for the relevance and impact of the study. In terms of showing of results, a good practice would be as follows:

- As long as the problem allows, the tests have to be performed with instances obtained in a benchmark. Obviously, the more instances are tested, the richer the study. Each instance that is used must be referenced, with its name and the benchmark it belongs to.
- It is vital to show the execution time, accompanied by its time unit and an explanation of the characteristics of the computer on which the tests were carried out.
- Apart of showing the runtime, to make a fair comparison between different techniques presented in different studies, it is highly recommended to show the number of iterations needed by the meta-heuristics to obtain the result of each execution.
- The more data displayed, the richer work. Thus, comparisons made with the meta-heuristic will be more reliable. For every instance of the problem in study, this information should include at least the number of executions carried out, and, both for objective and runtime, best and worst results, the average and the standard deviation.

The quality of a new technique must be checked applying the technique to several instances of the problem in study. The best option is to perform the tests with one of the benchmarks that can be found in the literature. Benchmarks are composed of instances of a particular problem, which researchers can try to solve to validate their new techniques [42]. Many of these instances have a known optimal solution. The effectiveness and efficiency of a meta-heuristic can be known by comparing its results with those offered by benchmarks. Taking into account this fact, it is much easier to contrast the quality of a technique if its results are compared with the results obtained by other techniques that have used the same benchmarks. Focusing on routing problems, there are a lot of benchmarks for a large number of problems, such as TSPLIB [43] or VRPWeb (*http://neo.lcc.uma.es/vrp*). Therefore, it has to be avoided, as far as possible, to do tests with unknown instances, as can be seen in [44], or [45].

At the time of showing results, one important point is the execution time. It can be considered a bad practice to show the results of a meta-heuristic without showing the execution times, as happens, for example, in [46] or [47]. Although it may be logical, it also must be specified in which unit the runtime is displayed, i.e., seconds, minutes, . . . Avoid this fact is considered a bad practice, as happens in [48]. Apart from showing in detail the runtimes of the technique, it is also important to note the characteristics of the computer used for the tests.

Although the runtime is helpful for comparing two techniques shown in the same study, for the comparison between techniques of different works the use of another parameter is more reasonable. This fact is given because it could be not fair to compare the runtime of different algorithms if they have been developed in different computers. It is logical that the more powerful the computer, the less time is needed to execute a meta-heuristic. Thus, a good measure to compare techniques is showing the number of iterations needed to obtain the resulting solution. This value will vary depending on the technique being used. For example, for a Tabu Search or Simulated Annealing, this value will be the number of iterations performed to reach the solution. For a GA, it could be the number of generations executed. Even though it is considered a good practice, nowadays very few studies show this parameter. [49] and [50] are two examples of this good practice. Besides this, a good practice to perform a more strict comparison of the results is the utilization of statistical tests. The well-known student's t-test, or the normal distribution z-test can be some of these statistical tests. In the literature, few articles perform an exam of this type, although it is a good practice [51, 52].

To provide richness and replicability to a study, it is highly recommended to display a complete set of results, showing important data as the mean, the best result or the standard deviation. As it is mentioned in [28], where several tips to compare heuristics are introduced, display only the best results of a heuristic, as is often done in the literature [53, 46], may create a false picture of the quality of the technique. This statement may be also applicable to the meta-heuristics, that is, to display only the best results in a comparison of techniques is considered a bad practice. Therefore, the average result based on multiple executions is considered the best basis for the comparison.

4 Conclusions and further work

Routing problems and meta-heuristics for their resolution are subject of a large number of studies annually. Every year, many novel techniques or modifications of existing ones are developed by researchers. For this reason, comparisons between techniques are widely used in many studies, since they are appropriate to check the quality of new proposed techniques. Despite this, there is no a specific methodology or procedure that helps researchers to compare different techniques, either within the same or different works. That is why in the literature can be found a lot of studies in which have been done bad practices presenting the results of their techniques, or comparing them with other works. This is what prompted us to do this work, in which a procedure of good practice to facilitate the comparison between meta-heuristics oriented to solve routing problems has been defined. With this procedure, researchers will be able to make comparisons easily and reliably.

The utility of this study could be very high. The future work related to this research can be very large, and is in the hands of all researchers working in the area of combinatorial optimization and routing problems. One of our proposals, is to modify the different benchmarks in the literature, so that not only the best results have to be shown for each instance. The details of the technique that has been used to achieve the best result should also be shown, mentioning the runtimes, iterations needed and details of implementation. This would facilitate the comparison between techniques and the replicability of the results.

In [42] we defined a methodological proposal in the showing of results in benchmarks to eliminate ambiguities in the comparison of VRP solving techniques. Now, in this paper, we extend that proposal introducing a procedure of good practice to present a meta-heuristic and its results properly, with the aim of facilitating the comparisons between different techniques. We have published some other papers related to good practices [54, 55]. As future work, we plan to perform a methodology to help the researchers to realize proper, detailed and objective analysis of the studies made. This way, we aim to facilitate the comprehension of the results and its capacity to be replicated and discussed. In addition, we want to extend our study to other fields inside the soft computing, where several interesting papers are published annually [56, 57].

References

1. Papadimitriou, C.: The new faces of combinatorial optimization. Journal of Combinatorial Optimization **7422**(1) (2012) 19–23
2. Korte, B., Vygen, J.: Combinatorial optimization: theory and algorithms. Volume 21. Springer-Verlag (2012)
3. Lawler, E., Lenstra, J., Kan, A., Shmoys, D.: The traveling salesman problem: a guided tour of combinatorial optimization. Volume 3. Wiley New York (1985)
4. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research **59**(3) (1992) 345–358

5. Lenstra, J., Kan, A.: Complexity of vehicle routing and scheduling problems. Networks **11**(2) (1981) 221–227
6. Onieva, E., Alonso, J., Pérez, J., Milanés, V., De Pedro, T.: Autonomous car fuzzy control modeled by iterative genetic algorithms. In: International Conference on Fuzzy Systems, IEEE (2009) 1615–1620
7. Zachariadis, E.E., Kiranoudis, C.T.: An effective local search approach for the vehicle routing problem with backhauls. Expert Systems with Applications **39**(3) (2012) 3174–3184
8. Simić, D., Simić, S.: Hybrid artificial intelligence approaches on vehicle routing problem in logistics distribution. In: Hybrid Artificial Intelligent Systems. Springer (2012) 208–220
9. Onieva, E., Milanés, V., Villagra, J., Pérez, J., Godoy, J.: Genetic optimization of a vehicle fuzzy decision system for intersections. Expert Systems with Applications **39**(18) (2012) 13148–13157
10. Stoean, R., Stoean, C.: Modeling medical decision making by support vector machines, explaining by rules of evolutionary algorithms with feature selection. Expert Systems with Applications **40**(7) (2013) 2677–2686
11. Gao, J., Sun, L., Gen, M.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research **35**(9) (2008) 2892–2907
12. Joo, C.M., Kim, B.S.: Genetic algorithms for single machine scheduling with time-dependent deterioration and rate-modifying activities. Expert Systems with Applications **40**(8) (2013) 3036–3043
13. Kirkpatrick, S., Gellat, C., Vecchi, M.: Optimization by simmulated annealing. Science **220**(4598) (1983) 671–680
14. Glover, F.: Tabu search, part i. INFORMS Journal on Computing **1**(3) (1989) 190–206
15. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional (1989)
16. De Jong, K.: Analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Michigan, USA (1975)
17. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science **344**(2) (2005) 243–278
18. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Volume 4., Perth, Australia (1995) 1942–1948
19. Garcia-Gonzalo, E., Fernandez-Martinez, J.: A brief historical review of particle swarm optimization (pso). Journal of Bioinformatics and Intelligent Control **1**(1) (2012) 3–16
20. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation. (2007) 4661–4667
21. Osaba, E., Diaz, F., Onieva, E.: Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. Applied Intelligence (2014) 1–22
22. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (abc) algorithm and applications. Artificial Intelligence Review (2012) 1–37
23. Osaba, E., Diaz, F., Onieva, E.: A novel meta-heuristic based on soccer concepts to solve routing problems. In: Proceeding of the fifteenth annual conference

companion on Genetic and evolutionary computation conference companion, ACM (2013) 1743–1744
24. Jiang, K., Song, B., Shi, X., Song, T.: An overview of membrane computing. Journal of Bioinformatics and Intelligent Control **1**(1) (2012) 17–26
25. Glover, F., Gutin, G., Yeo, A., Zverovich, A.: Construction heuristics for the asymmetric tsp. European Journal of Operational Research **129**(3) (2001) 555–568
26. Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S.: Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review **13**(2) (1999) 129–170
27. Mladenović, N., Urošević, D., Hanafi, S.: Variable neighborhood search for the travelling deliveryman problem. A Quarterly Journal of Operations Research **11**(1) (2013) 57–73
28. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Sciences **39**(1) (2005) 104–118
29. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: Vrp with time windows. The vehicle routing problem **9** (2002) 157–193
30. Chen, A.l., Yang, G.k., Wu, Z.m.: Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University Science A **7**(4) (2006) 607–614
31. Wang, C.H., Lu, J.Z.: A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. Expert Systems with Applications **36**(2) (2009) 2921–2936
32. Leung, S., Zhang, Z., Zhang, D., Hua, X., Lim, M.: A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. European Journal of Operational Research **225**(2) (2013) 199–210
33. Solomon, M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research **35**(2) (1987) 254–265
34. Julstrom, B.A.: Very greedy crossover in a genetic algorithm for the traveling salesman problem. In: ACM symposium on Applied computing. (1995) 324–328
35. Davis, L.: Applying adaptive algorithms to epistatic domains. In: Proceedings of the international joint conference on artificial intelligence. Volume 1. (1985) 161–163
36. Osaba, E., Onieva, E., Carballedo, R., Diaz, F., Perallos, A.: An adaptive multi-crossover population algorithm for solving routing problems. In: Proceedings of the VI International Workshop on Nature Inspired Cooperative Strategies for Optimization. Springer (2014) 113–124
37. Syswerda, G.: Schedule optimization using genetic algorithms. Handbook of genetic algorithms (1991) 332–349
38. Ray, S., Bandyopadhyay, S., Pal, S.: New operators of genetic algorithms for traveling salesman problem. In: Proceedings of the 17th International Conference on Pattern Recognition. Volume 2. (2004) 497–500
39. Burke, E., Kendall, G.: Comparison of meta-heuristic algorithms for clustering rectangles. Computers & Industrial Engineering **37**(1) (1999) 383–386
40. Ahmed, Z.H.: Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. International Journal of Biometrics and Bioinformatics **3**(6) (2010) 96
41. Albayrak, M., Allahverdi, N.: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Systems with Applications **38**(3) (2011) 1313–1320

42. Osaba, E., Carballedo, R.: A methodological proposal to eliminate ambiguities in the comparison of vehicle routing problem solving techniques. In: Proceedings of the 4th International Joint Conference on Computational Intelligence. (2012) 310–313
43. Reinelt, G.: Tsplib, a traveling salesman problem library. INFORMS Journal on Computing **3**(4) (1991) 376–384
44. Wang, L., Zhang, J., Li, H.: An improved genetic algorithm for tsp. In: International Conference on Machine Learning and Cybernetics. Volume 2. (2007) 925–928
45. Thamilselvan, R., Balasubramanie, P.: A genetic algorithm with a tabu search (gta) for traveling salesman problem. International Journal of Recent Trends in Engineering **1**(1) (2009) 607–610
46. Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., Liu, H.: Solve traveling salesman problem using particle swarm optimization algorithm. International Journal of Computer Science Issues **9**(2) (2012) 264–271
47. Sallabi, O.M., El-Haddad, Y.: An improved genetic algorithm to solve the traveling salesman problem. World Academy of Science, Engineering and Technology **52** (2009) 471–474
48. Nemati, K., Shamsuddin, S., Kamarposhti, M.: Using imperial competitive algorithm for solving traveling salesman problem and comparing the efficiency of the proposed algorithm with methods in use. Australian Journal of Basic and Applied Sciences **5**(10) (2011) 540–543
49. Tsubakitani, S., Evans, J.R.: Optimizing tabu list size for the traveling salesman problem. Computers & Operations Research **25**(2) (1998) 91–97
50. Ray, S., Bandyopadhyay, S., Pal, S.: New genetic operators for solving tsp: Application to microarray gene ordering. Pattern Recognition and Machine Intelligence **3776** (2005) 617–622
51. Nikolić, M., Teodorović, D.: Empirical study of the bee colony optimization (bco) algorithm. Expert Systems with Applications **40**(11) (2013) 4609–4620
52. Osaba, E., Onieva, E., Carballedo, R., Diaz, F., Perallos, A., Zhang, X.: A multi-crossover and adaptive island based population algorithm for solving routing problems. Journal of Zhejiang University SCIENCE C **14**(11) (2013) 815–821
53. Pullan, W.: Adapting the genetic algorithm to the travelling salesman problem. In: IEEE Congress on Evolutionary Computation. Volume 2. (2003) 1029–1035
54. Osaba, E., Onieva, E., Diaz, F., Carballedo, R., Perallos, A.: Comments on "albayrak, m., & allahverdy n.(2011). development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. expert systems with applications, 38 (3), 1313–1320": A proposal of good practice. Expert Systems with Applications **41**(4) (2014) 1530–1531
55. Osaba, E., Carballedo, R., Diaz, F., Perallos, A.: Discussion related to "wang, c.-h., & lu, j.-z.(2009). a hybrid genetic algorithm that optimizes capacitated vehicle routing problem. expert systems with applications, 36 (2), 2921–2936". Expert Systems with Applications **40**(14) (2013) 5425–5426
56. Corchado, E., Baruque, B.: Wevos-visom: An ensemble summarization algorithm for enhanced data visualization. Neurocomputing **75**(1) (2012) 171–184
57. Corchado, E., Wozniak, M., Abraham, A., C. P. L. F. de Carvalho, A., Snásel, V.: Recent trends in intelligent data analysis. Neurocomputing **126** (2014) 1–2