Comparing Evolutionary Strategies on a Biobjective Cultural Algorithm

Carolina Lagos,1 Broderick Crawford,1,2 Enrique Cabrera,3 Ricardo Soto,1,4 José-Miguel Rubio,1,5 and Fernando Paredes6

1 Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile
2 Universidad Finis Terrae, 7500000 Santiago, Chile
3 CIMFAV Facultad de Ingeniería, Universidad de Valparaíso, 2362735 Valparaíso, Chile
4 Universidad Autónoma de Chile, 7500138 Santiago, Chile
5 Departamento de Computación e Informática, Universidad de Playa Ancha, 33449 Valparaíso, Chile
6 Escuela de Ingeniería Industrial, Universidad Diego Portales, 8370109 Santiago, Chile

Correspondence should be addressed to Carolina Lagos; carolina.lagos.c@mail.pucv.cl

Received 9 April 2014; Accepted 27 June 2014; Published 31 August 2014

Academic Editor: Xin-She Yang

Copyright © 2014 Carolina Lagos et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Evolutionary algorithms have been widely used to solve large and complex optimisation problems. Cultural algorithms (CAs) are evolutionary algorithms that have been used to solve both single and, to a less extent, multiobjective optimisation problems. In order to solve these optimisation problems, CAs make use of different strategies such as normative knowledge, historical knowledge, circumstantial knowledge, and among others. In this paper we present a comparison among CAs that make use of different evolutionary strategies; the first one implements a historical knowledge, the second one considers a circumstantial knowledge, and the third one implements a normative knowledge. These CAs are applied on a biobjective uncapacitated facility location problem (BOUFLP), the biobjective version of the well-known uncapacitated facility location problem. To the best of our knowledge, only few articles have applied evolutionary multiobjective algorithms on the BOUFLP and none of those has focused on the impact of the evolutionary strategy on the algorithm performance. Our biobjective cultural algorithm, called BOCA, obtains important improvements when compared to other well-known evolutionary biobjective optimisation algorithms such as PAES and NSGA-II. The conflicting objective functions considered in this study are cost minimisation and coverage maximisation. Solutions obtained by each algorithm are compared using a hypervolume S metric.

1. Introduction

Evolutionary algorithms (EAs) are an effective alternative to (approximately) solve several large and complex optimisation problems, as they are able to find good solutions for a wide range of problems in acceptable computational time. Although less studied, EAs for multiobjective optimisation (MO) problems, called evolutionary multiobjective optimisation (EMO), have demonstrated to be very effective. In fact, during the last two decades, several authors have focused their efforts on the development of several EMO algorithms to solve a wide range of MO problems. For instance, Maravall and de Lope [1] use a genetic algorithm (GA) to solve the multiobjective dynamic optimisation for automatic parking system. In [2] the authors propose an improvement to the well-known NSGA algorithm (called NSGA-II) based on an elitist approach. In [3] the author presents an EMO algorithm applied on a specific variation of the well-studied capacitated vehicle routing problem (CVRP), where the author includes in the EMO algorithm an explicit collective memory method, namely, the extended virtual loser (EVL). Other well-known EMO algorithms developed during the last two decades are PAES [4] and MO particle swarm optimisation [5]. More recently, hybrid techniques have been also applied to a large
number of optimisation problems (see [6]). A comprehensive literature review related to EMO algorithms can be found in [7].

EMO algorithms have some problems that must be taken into account, though. For instance, they tend to fall into premature convergence with low evolution efficiency [8]. This is because of implicit information embodied in the evolution process and domain knowledge corresponding to optimisation problems which are not fully included in the solution approach [9]. To overcome these problems, one can make use of implicit evolution information. Reynolds [10] proposes an EA, called cultural algorithm (CA), which is inspired from human culture evolution process and that makes use of the implicit evolution information generated at each iteration. The CAs have a dual evolution structure which consists of two spaces: population and belief space. On the one hand, the population space works as in any other EA, that is, using evolutionary operators such as mutation and crossover. On the other hand, in the belief space, implicit knowledge is extracted from selected individuals in the population and stored in a different way. Then, they are used to guide the whole evolution process in the population space such that they can induce the population to escape from local optimal solutions. It has been proved that CAs can effectively improve the evolution performance [9]. Although less studied, CAs have been also used to solve MO problems. Coello et al. [11] and Coello et al. [12], two remarkable surveys on CAs, only mention Coello and Landa [13] work as example of a CA application solving MO problems. More recently, Zhang et al. [14] present a CA which is enhanced by using a particle swarm optimisation algorithm. This enhanced CA is applied to fuel distribution MO problem. Srinivasan and Ramakrishnan [15] present a MO cultural algorithm that is applied on data mining domain. In [16] the authors applied a CA to a biobjective portfolio selection problem using normative knowledge in the belief space. In [17] authors present a formal framework to implement MO cultural algorithms. To the best of our knowledge, [18] is the only article that uses CAs to solve the biobjective uncapacitated facility location problem (BOUFLP). Furthermore, we did not find any article which compares the performance of CAs using different evolutionary strategies at the belief space level. Thus, in this paper we present an extension of the biobjective cultural algorithm (BOCA) developed in [18]. We use two different strategies at the belief space level and compare the performance of our new algorithms with the performance of the previous one. We also compare its results with other well-known EMO algorithms such as PAES and NSGA-II. Obtained solutions were compared using the hypervolume S metric proposed in [19].

The remaining of this paper is organised as follows. Section 2 shows an overview on MO focused on EMO algorithms. Section 2.1 presents briefly the BOUFLP and some of its distinctive features. In Section 3, we describe our implementation for the BOCA algorithm. We describe the main differences between our implementation and the one in [18]. Section 3.2 presents the BOCA algorithm applied to a set of well-known instances from the literature. Finally, Section 4 presents the conclusions of this work.

2. Overview

In this section we show an overview of topics related to this paper. In Section 2.1, MO concepts are presented, emphasizing EMO algorithms and its state of art. More specifically, we focus on the development of EMO algorithms for MO combinatorial optimisation (MOCO) problems. In Section 2.2 we present the BOUFLP formulation based on a cost-coverage approach. Finally, in Section 2.3 we present an overview of CAs and its multiobjective extension. Details of our CA implementation are also presented at the end of this section.

2.1. (Evolutionary) Multiobjective Optimisation. In this section we briefly introduce the main principles of MO problems and, particularly, MOCO problems. For a comprehensive review of this topic see [20, 21]. In this paper we will make use of the following notation for the comparison of vectors (solutions). Let $y$ and $y' \in \mathbb{R}^p$. We say that $y \leq y'$ if $y_k \leq y'_k \forall k = 1, \ldots, p$. Similarly, we will say that $y \leq y'$ if $y \leq y'$ but $y \neq y'$. Finally, we say that $y < y'$ if $y_k < y'_k \forall k = 1, \ldots, p$. A solution $\hat{x} \in \mathbb{R}^n$, with $n$ being equal to the number of decision variables, is called an efficient solution and its image $f(\hat{x})$, with $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, a nondominated point of the MO problem if there is no $x \in \mathbb{R}^n$, with $x \neq \hat{x}$, such that $f(x) \leq f(\hat{x})$. In [22] the author describes several excellence relations. These relations establish strict partial orders in the set of all nondominated points related to different aspects of their quality. Previously, in [23, 24] the authors consider several outperformance relations to address the closeness of the set of nondominated points found by an algorithm to the actual set of nondominated points, called Pareto Frontier (PF). In [25] a comprehensive explanation of the desirable features of an approximation to the PF is presented. In this paper, we choose the $S$ metric which is properly explained in [24]. The $S$ metric calculates the hypervolume of a multidimensional region [19] and allows the integration of aspects that are individually measured by other metrics.

An advantage of the $S$ metric is that each algorithm can be assessed independently of the other algorithms involved in the study. However, the $S$ values of two sets $A$ and $B$, namely, $S_A$ and $S_B$, respectively, cannot be used to derive whether either set entirely dominates the other. Figure 1(a) shows a situation where $S_A > S_B$ and set $A$ completely dominates set $B$. Figure 1(b) shows a situation where $S_A > S_B$, but neither $A$ dominates $B$ nor $B$ dominates $A$.

In this paper we use EAs to solve the BOUFLP. An EA is a stochastic search procedure inspired by the evolution process in nature. In this process, individuals evolve and the fitter ones have a better chance of reproduction and survival. The reproduction mechanisms favor the characteristics of the stronger parents and hopefully produce better children guaranteeing the presence of those characteristics in future generations [26]. EAs have been successfully applied to a large number of both single- and multiobjective optimisation problems. Comprehensive reviews of EMO algorithms are presented in [11, 24] and more recently in [12]. A more general
Two main (conflicting) objectives can be identified in the FLP:

(i) minimise the total cost associated with the facility installation and customer allocation and

(ii) maximise the customers rate coverage.

Several works on single-objective optimisation have been carried out considering these two objectives separately. On the one hand, uncapacitated FLP (UFLP) is one of most studied FLPs in the literature. In the UFLP the main goal is to minimise the location-allocation cost of the network. On the other hand, \( p \) median FLPs are one of the most common FLPs among those that are focused on coverage maximisation. Most important FLP models are well described and formalised in [29]. MO FLPs have been also well studied in the literature during the last two decades. A survey on this topic can be found in [31].

As we mentioned before, in this paper we solve the BOUFLP. BOUFLP has been modelled with minisum and maxisum objectives (cost and coverage). The following model formulation is based on [32]. Let \( I = \{1, \ldots, m\} \) be the set of potential facilities and \( J = \{1, \ldots, n\} \) the set of customers. Let \( FC_i \) be the fixed cost of opening facility \( i \) and \( d_j \) the demand of customer \( j \). Let \( c_{ij} \) be the cost of assigning the customer \( j \) to facility \( i \) and \( h_{ij} \) the distance between facility \( i \) and customer \( j \). Let \( D_{\text{MAX}} \) be the maximal covering distance; that is, customers within this distance to an open facility are considered well served. Let \( Q^j = \{ i \in I : h_{ij} \leq D_{\text{MAX}} \} \) be the set of facilities that could serve customer \( j \) within the maximal covering distance \( D_{\text{MAX}} \). Let \( x_i \) be 1 if facility \( i \) is open and 0, otherwise. Let \( y_{ij} \) be 1 if the whole demand of customer \( j \) is served by facility \( i \) and 0, otherwise. Consider

\[
\begin{align*}
\text{minimise} & \quad f(x, y) \\
\text{s.t.} & \quad \sum_{j \in J} y_{ij} \quad \forall j \in I \\
& \quad y_{ij} \leq x_i \quad \forall j \in J, i \in I \\
& \quad y_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I \\
& \quad x_i \in \{0, 1\} \quad \forall i \in I
\end{align*}
\]

with \( f : \mathbb{R}^n \rightarrow \mathbb{R}^p \) and \( p = 2 \). Objective functions \( f_1 \) and \( f_2 \) are as follows:

\[
\begin{align*}
f_1(x, y) &= \sum_{i \in I} FC_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}, \\
f_2(x, y) &= -\sum_{j \in J} \sum_{i \in Q^j} y_{ij}
\end{align*}
\]

Equation (6) represents total operating cost; the first term corresponds to location cost, that is, the sum of the fixed costs of all the open facilities, and the second term represents the allocation cost, that is, the cost of attending customer demand by an open facility. Equation (7) measures coverage as the sum of the demand of customers attended by open facilities within the maximal covering distance. Equations (2) and (3)
ensure that each customer is attended by only one facility. Equation (3) also forces customer to be assigned to an open facility. Finally, equations (4) and (5) set decision variables as binary.

2.3. Biobjective Cultural Algorithm. The experience and beliefs accepted by a community in a social system are the main motivations for the creation of the CAs. Originally proposed by Reynolds [10], CAs model the evolution of cultural systems based on the principles of human social evolution. In this case, evolution is seen as an optimisation process [10]. The CAs guide the evolution of the population based on the knowledge. Knowledge acquired during previous iterations is provided to future generations, allowing accelerating the convergence of the algorithm to good solutions [33]. Domain knowledge is modelled separately from the population, because there is certain independence between them which allows us to work and model them separately, in order to enhance the overall algorithm performance. Figure 3 shows this interaction.

CAs are mainly characterised by presenting two inheritance systems: one at population level, called population space, and the other at knowledge level, called belief space. This key feature is designed to increase the learning rates and convergence of the algorithm and thus to do a more responsive system for a number of problems [34]. Moreover, it allows us to identify two significant levels of knowledge: a microevolutionary level (represented by the population space) and macroevolutionary level (represented by the space of beliefs) [35].

CAs have the following components: population space (set of individuals who have independent features) [35]; belief space (stored individuals acquired in previous generations) [34]; computer protocol, connecting the two spaces and defining the rules on the type of information to be exchanged between them by using the acceptance and influence functions; and finally knowledge sources which are described in terms of their ability to coordinate the distribution of individuals depending on the nature of a problem instance [35]. These knowledge sources can be of the following types: circumstantial, normative, domain, topographic, and historical.

The most distinctive feature of CAs is the use of the belief space which through an influence function affects future generations. For this reason, in this paper we focus on the effect on the algorithm performance of changes in such an influence function. To do this, we have considered results obtained previously in [18], where the authors used an influence function based on historical knowledge, and we compare those results with our BOCA implementation which considers two influence functions: the first one based on circumstantial knowledge and the second one based on normative knowledge. Algorithm 1 shows the general procedure of our BOCA algorithm.
Belief space
Population space
Update ()
Accept ()
Influence ()

New generation ()
Best individuals ()

2 2 2
1 1 1
n−1 n
n−1 n
n−1 n

Figure 3: CA general diagram.

To initialise the population, we use a semirandom function. In its first phase, this function defines in a stochastic way the set of facilities that will be opened (selected facilities). Then, we allocate each customer to a selected facility minimising the cost function $f_1$ while avoiding minimising the coverage function $f_2$. This strategy provides better results than using completely random initial populations, and its computational time additional cost is marginal.

To obtain the next generation, two parents are used in a recombination process. To avoid local optimal values, we do not overuse the culture. Thus, a parent is selected from the population to obtain diversity and the other parent is selected from the belief space to influence the next generation. The belief space keeps a list of all the individuals which meet some criteria. These criteria depend on what knowledge the algorithm implements. In this paper the circumstantial knowledge selects the best individuals found so far for each objective function. Thus, one individual will give us information on the best value found for $f_1$ and the other will do the same for $f_2$. The historical knowledge stores a list of individuals with the best fitness value found so far. The fitness value is calculated as the hypervolume $S$ that is covered by an individual. Finally, normative knowledge considers a list of individuals which are pairwise nondominated with respect to the other individuals of their generation.

Let $|J|$ be the number of available facilities and let $|I|$ be the number of customers of our BOUFLP. In this paper decisions variables $x$ and $y$ are represented by a binary $|J|$-length vector and $|J| \times |I|$ matrix, respectively. Comparing two different solutions (individuals) needs an evaluation criterion. In this paper we use the same criterion explained in [18].

3. Computational Experiments

In this section we present the set of instances that are used in this study as well as results obtained by our BOCA implementation.

3.1. Instances Presentation. The instances that were used in this paper correspond to random instances using a problem generator that follows the methodology from UflLib [36]. Previous works in the literature have also used this problem generator to create their test instances [18, 26].

The BOCA algorithm has several parameters that need to be set. As in [18], the number of generations considered in this paper is equal to 100. Population size $L$ is set equal to 100, mutation probability in the population space $P_{ps}$ is equal to 0.2, and probability of mutation in the belief space $P_{bs}$ is 0.04. Both $L$ and $P_{ps}$ values are different from the values used in [18]. These values are chosen as they all together yield to the best performance of the algorithm given some test instances. Thus, although resulting values are different from that used in [18], the method we use to set them is the same as that used in that work. This is important in order to fairly compare the different BOCA algorithms.

3.2. Results and Discussion. In this section we compare the results obtained by the previous BOCA algorithm (see [18] for further details) and our approach. Moreover, a comparison between results obtained by well-known EMO algorithms such as NSGA-II and PAES and our BOCA algorithm is also presented in this section.

Tables 1 and 2 show the results obtained by the BOCA implementations using historical [18], circumstantial, and normative knowledge, respectively. In the same way, Tables 3 and 4 present the results obtained by the well-known NSGA-II and PAES algorithms. For each algorithm $S$ value (%), time $t$ (in seconds), and the number of efficient solutions $|\bar{X}|$ have been included in these tables. As we mentioned before,
We then compare our BOCA implementations with the one presented in [18]. Tables 5 and 6 show a comparison between those algorithms. As we can see, when compared in terms of its $S$ value (the bigger, the better), BOCA algorithm using historical knowledge ($BOCA^H$) performs consistently better than the ones using circumstantial ($BOCA^C$) and normative ($BOCA^N$) knowledge. In fact $BOCA^H$ obtains a $S$ value that is, in average, 5.8% bigger than the one obtained by $BOCA^C$ and 6.5% bigger than the $S$ value obtained by $BOCA^N$. When compared in terms of the CPU time needed to reach the number of iterations (generations), $BOCA^H$ is, in average, faster than both $BOCA^C$ and $BOCA^N$ algorithms.
We can note that for $B$ instances, times required by BOCA$^H$ and BOCA$^C$ are, in average, quite similar (only 1.6% of difference). Finally, when we look at the number of efficient solutions found by each algorithm ($|\overline{X}|$), we can see that, again, BOCA$^H$ outperforms both BOCA$^C$ and BOCA$^N$ algorithms. In this case, the average number of efficient solutions found by the BOCA$^H$ algorithm is about 20% bigger than the one obtained by the other two approaches.

Results above are consistent with the good performance observed by the BOCA$^H$ approach in [18]. Moreover, results show that performance of the BOCA algorithm depends largely on the selected knowledge and it can make the difference in terms of $S$ value, time, and number of efficient solutions found by the algorithm. This is an important finding.
as it points out the relevance of the choice of a specific type of knowledge.

We now compare BOCA$^C$ and BOCA$^N$ algorithms to the well-known NSGA-II and PAES algorithms. Tables 7 and 8 show a comparison between our BOCA$^C$ algorithm and the NSGA-II and PAES algorithms. As we can see, although BOCA$^C$ obtains, in average, a $S$ value 6.8% lower than the one obtained by the NSGA-II algorithm, it is more than three times faster. Moreover, when BOCA$^C$ is compared to PAES algorithm, the obtained $S$ values are, in average, equivalent while BOCA$^C$ is around 30% faster than PAES. PAES obtains, in average, more efficient points than BOCA$^C$ though (9.32%).

Finally, Tables 9 and 10 show a comparison between BOCA$^N$ and NSGA-II and PAES algorithms. BOCA$^N$ performs quite similar to PAES algorithm with respect to both $S$ value and the number of obtained efficient solutions. However, BOCA$^N$ is faster than PAES. Similar situation occurs when BOCA$^N$ is compared to NSGA-II algorithm. Although NSGA-II obtains better values for both $S$ and $|\bar{x}|$, BOCA$^N$ is much faster than NSGA-II. This situation can be explained by the very fast performance that our BOCA$^N$ algorithm obtains for the set of small instances. When we look further at the results, we can note that if we only consider both medium and large size instances, execution times obtained by both algorithms are quite similar to each other. This result confirms what is outlined in [18] in the sense of the good performance that the BOCA algorithm shows. Furthermore, our results confirm this good performance with respect to other well-known EMO algorithms does not depend on which type of knowledge is considered. However, as we mentioned before, the choice of the knowledge used on the BOCA algorithm is an important issue and it has an impact on the algorithm performance.

### 4. Conclusions and Future Work

Evolutionary algorithms are a very good alternative to solve complex combinatorial optimisation problems. In this paper
Moreover, our BOCA algorithm can be used to solve other problems with the advantages of each kind of knowledge at the same time. Hybrid knowledge could be implemented in order to exploit the impact on the performance of the BOCA algorithm and the right knowledge implemented depending on the problem needed in order to find patterns that allow us to get the best value, time, and number of efficient solutions found by the corresponding algorithm. This is an important finding as it points out the relevance of the choice of a specific type of knowledge. Moreover, our results confirm the good performance showed by the BOCA algorithm with respect to other well-known EMO algorithms such as NSGA-II and PAES algorithms. The BOCA algorithm is very competitive when compared to those EMO algorithms independently of the type of knowledge implemented.

As a future work we think that more investigation is needed in order to find patterns that allow us to get the right knowledge implemented depending on the problem features. As we mentioned before, the knowledge choice has an impact on the performance of the BOCA algorithm and therefore it must be studied in depth. Also, as future work, hybrid knowledge could be implemented in order to exploit the advantages of each kind of knowledge at the same time. Moreover, our BOCA algorithm can be used to solve other interesting MOPs arising in the logistic field, such as routing or scheduling problems.

### Appendix

**Result Tables**

In this appendix section, obtained results are presented. Columns $S_i$ show the S value obtained by algorithm \{\} as %. Algorithms are indexed as follows. The original BOCA algorithm is indexed by \{1\}. BOCA algorithms using circumstantial and normative knowledge are indexed by \{2\} and \{3\}, respectively. Finally, the other EAs considered in this paper, namely, NSGA-II and PAES, are indexed by \{4\} and \{5\}, respectively. Columns $t_i$ show the time obtained by each algorithm in seconds. Columns $|\hat{X}_i|$ show the number of efficient solutions found by the corresponding algorithm. Finally operator $\Delta(\cdot)$ shows a value that is equivalent to $(\cdot) - 100 \times (\cdot)$.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### References

[1] D. Maravall and J. de Lope, “Multi-objective dynamic optimization with genetic algorithms for automatic parking,” *Soft Computing*, vol. II, no. 3, pp. 249–257, 2007.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: Nsga III” in *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph et al., Eds., vol. 1917 of Lecture Notes in Computer Science, pp. 849–858, Springer, Berlin, Germany, 2000.

[3] I. Borgulya, “An algorithm for the capacitated vehicle routing problem with route balancing,” *Central European Journal of Operations Research*, vol. 16, no. 4, pp. 311–343, 2008.

[4] P. J. Angeline, Z. Michalewicz, M. Schoenauer, Y. Yao, and A. Zalzala, Eds., *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation*, vol. 1, IEEE Press, 1999.

[5] C. A. Coello Coello and M. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization,” in *Proceedings of the Congress on Evolutionary Computation (CEC ’02)*, vol. 2, pp. 1051–1056, 2002.

[6] W. K. Mashwani, “Comprehensive survey of the hybrid evolutionary algorithms,” *International Journal of Applied Evolutionary Computation*, vol. 4, pp. 1–19, 2013.

[7] R. Bhattacharya and S. Bandyopadhyay, “Solving conflicting bi-objective facility location problem by NSGA II evolutionary algorithm,” *International Journal of Advanced Manufacturing Technology*, vol. 51, no. 1–4, pp. 397–414, 2010.

[8] B. Crawford, C. Lagos, C. Castro, and F. Pedreses, “A cultural algorithm for solving the set covering problem,” in *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, P. Melin, O. Castillo, E. Ramírez, J. Kacprzyk, and W. Pedrycz, Eds., vol. 41 of *Advances in Soft Computing*, pp. 408–415, Springer, Berlin, Germany, 2007.
[9] Y. Guo, J. Cheng, Y. Cao, and Y. Lin, “A novel multi-population cultural algorithm adopting knowledge migration,” Soft Computing, vol. 15, no. 5, pp. 897–905, 2011.

[10] R. G. Reynolds, “An introduction to cultural algorithms,” in Proceedings of the 3rd Annual Conference on Evolutionary Programming, pp. 131–139, World Scientific, 1994.

[11] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), Springer, Secaucus, NJ, USA, 2006.

[12] C. A. Coello, C. Dhaenens, and L. Jourdan, Advances in Multi-Objective Nature Inspired Computing, Springer, 1st edition, 2010.

[13] R. Zhang, J. Zhou, L. Mo, S. Ouyang, and X. Liao, “Economic environmental dispatch using an enhanced multi-objective cultural algorithm,” Electric Power Systems Research, vol. 99, pp. 18–29, 2013.

[14] S. Srinivasan and S. Ramakrishnan, “A social intelligent system for multi-objective optimization of classification rules using cultural algorithms,” Computing, vol. 95, no. 4, pp. 327–350, 2013.

[15] G. G. Cabrera, C. Vasconcellos, R. Soto, J. M. Rubio, F. Paredes, and B. Crawford, “An evolutionary multi-objective optimization algorithm for portfolio selection problem,” International Journal of Physical Sciences, vol. 6, no. 22, pp. 5316–5327, 2011.

[16] R. Reynolds and D. Liu, “Multi-objective cultural algorithms,” in Proceedings of the IEEE Congress of Evolutionary Computation (CEC ‘11), pp. 1233–1241, June 2011.

[17] G. Cabrera, J. M. Rubio, D. Díaz, B. Fernández, C. Cubillos, and R. Soto, “A cultural algorithm applied in a BiObjective uncapacitated facility location problem,” in Evolutionary Multi-Criterion Optimization, R. Takanashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576 of Lecture Notes in Computer Science, pp. 477–491, Springer, Berlin, Germany, 2011.

[18] M. Ehrig, T. Kottmann, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” Evolutionary Computation, vol. 8, no. 2, pp. 173–195, 2000.

[19] J. G. Villagrasa, F. Palacios, and A. L. Medaglia, “Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example,” Annals of Operations Research, vol. 147, pp. 109–141, 2006.

[20] M. Ehrgott and X. Gandibleux, “Hybrid metaheuristics for multi-objective combinatorial optimization,” in Hybrid Metaheuristics, C. Blum, M. J. B. Aguiler, A. Roli, and M. Sampels, Eds., vol. 114 of Studies in Computational Intelligence, pp. 221–259, Springer, Berlin, Germany, 2008.

[21] J. Bramel and D. Simchi-Levi, The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management, Springer, New York, NY, USA, 1997.

[22] M. S. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, Wiley-Interscience, New York, NY, USA, 1st edition, 1995.

[23] Z. Drezner and H. Hamacher, Facility Location: Applications and Theory, Springer, Berlin, Germany, 2002.

[24] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, “Multiple criteria facility location problems: a survey,” Applied Mathematical Modelling, vol. 34, no. 7, pp. 1689–1709, 2010.

[25] C. S. Revelle and G. Laporte, “The plant location problem: new models and research prospects,” Operations Research, vol. 44, no. 6, pp. 864–874, 1996.

[26] R. G. Reynolds, New Ideas in Optimization, McGraw-Hill, Maidenhead, UK, 1999.

[27] R. Lande Becerra and C. A. Coello Coello, “A cultural algorithm with differential evolution to solve constrained optimization problems,” in Advances in Artificial Intelligence (IBERAMIA ’04), C. Lemaître, C. Reyes, and J. A. González, Eds., vol. 3315 of Lecture Notes in Computer Science, pp. 881–890, Springer, Berlin, Germany, 2004.

[28] C. Soza, R. Landa, M. Riff, and C. Coello, “A cultural algorithm with operator parameters control for solving timetabling problems,” in Foundations of Fuzzy Logic and Soft Computing, P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds., vol. 4529 of Lecture Notes in Computer Science, pp. 810–819, Springer, Berlin, Germany, 2007.

[29] M. Hoefner, ULLib, Benchmark Instances for the Uncapacitated Facility Location Problem,” 2014.