25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1400
MULTIOBJECTIVE OPTIMIZATION OF HEAT EXCHANGER DESIGN THROUGH HEURISTIC KALMAN ALGORITHM

Pedro Henrique Leite da Silva Pires Domingues
Pontifical Catholic University of Rio de Janeiro - Rua Marquês de São Vicente, 225, Zip code 22451-900, Rio de Janeiro, RJ
phd.engmec@gmail.com
Leandro dos Santos Coelho
Pontifical Catholic University of Paraná - R. Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR
Federal University of Paraná - Av. Coronel Francisco Heráclito dos Santos, 100, Zip code 80215-901, Curitiba, PR
leandro.coelho@pucpr.br
Roberto Zanetti Freire
Pontifical Catholic University of Paraná - R. Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR, 80215-901
roberto.freire@pucpr.br
Ravipudi Ventaka Rao
Sardar Vallabhbhai National Institute of Technology - Ichchhanath Surat- Dumas Road & Keval Chowk, Zip code 395007, Surat, GJ
ravipudirao@gmail.com
Helon Vicente Hultmann Ayala
Pontifical Catholic University of Rio de Janeiro - Rua Marquês de São Vicente, 225, Zip code 22451-900, Rio de Janeiro, RJ
helon@puc-rio.br

Abstract. *Due to the constant presence of heat exchangers in a wide range of industries, the design optimization of heat exchangers draws the scientific community's attention, which has explored the use of metaheuristics in the development of this type of equipment. The heat exchanger design optimization generally considers effectiveness and cost as objectives, which recommends the use of a multiobjective approach due to the conflicting aspect of the objectives. A recent multiobjective metaheuristic for non-convex constrained optimization problems is the Multiobjective heuristic Kalman Algorithm (MOHKA), which is attractive by its ease pf implementation and the use of few adjustable parameters, among them are the slowdown factor (α) and the number of solutions (N_ξ) to calculate the measure ξ, an important variable for variance calculus and process evolution. In order to improve the MOHKA performance, this work proposes new MOHKA versions with i) N_ξ suppression for randomness addition, ii) new architectures with population creation directly dependent of ξ, iii) diversity preservation mechanism alteration for the niching procedure and iv) α varied in a range of [0.4,0.9], considering the area under the Inverted Generational Distance (IGD) evolution curve and the final IGD values as the comparison metrics. Final results showed that the α did not impact the performance, the niching procedure inclusion decreased the optimization efficiency, and the MOHKA with the new architecture and addition of randomness presented a potential for improvement.*
Keywords: *Multiobjective Optimization, Multiobjective Heuristic Kalman Algorithm, Niching Procedure, Heat Exchanger*

1. INTRODUCTION

Heat exchangers are common devices used in various industry fields, with the main purpose of transferring thermal energy through two or more fluids or also a fluid and a solid surface (Shah and Sekulic, 2003). In order to improve the heat transfer, it could be used treated, rough or extended contact surfaces, coiled tubes or fluid vibration (Fabbri, 2000; Bintoro *et al.*, 2005). However, this design features affect the heat exchanger economical cost and its efficiency, which is higher with lesser energy resource consumption.

In the recent years, the academic community has been studying the use of metaheuristics on the heat exchanger design optimization, such as in Patel and Rao (2010); Ayala *et al.* (2016); Mohanty (2016); Tharakeshwar *et al.* (2017); Esfe *et al.* (2018) and Zarea *et al.* (2018). One of the recent proposed optimization metaheuristics are the Heuristic Kalman Algorithm (HKA) (Toscano and Lyonnet, 2012) and its multiobjective version (MOHKA) (Ayala *et al.*, 2017), that sought inpiration in the Kalman filter philosophy, considering that the solution generated has its noisy corruption iteratively removed by Kalman equations, ensuring an optimal solution (Ayala *et al.*, 2017). The idea of MOHKA is to add an external archive to store nondominated solutions, which is ordered first by Pareto dominance and then by crowding distance (Deb *et al.*, 2002), a diversity preservation mechanism.

The MOHKA method is attractive by the use of few adjustable parameters and the ease of implementation. Seeking to potentialize these characteristics, it would be interesting to i) evaluate new MOHKA versions with the substitution of a parameter for adding randomness and ii) having a simplified architecture.

In Deb and Jain (2014) it is proposed a diversity preservation mechanism called niching procedure to replace the crowding distance operator, which tends to become computationally expensive for many-objective problems. The main advantages of the niching procedure is that the mechanism adapts to the number of objectives and normalizes them, which proved effective in Seada and Deb (2016) and Seada *et al.* (2019).

Therefore, in the present work, the design optimization of a plate-fin and a shell-tube heat exchangers considering a multiobjective approach, as well as the biobjective ZDT1 test problem, presented by Zitzler *et al.* (2000) are proposed for the study of new MOHKA versions optimization performance. The proposed algorithms considered addition of randomness, architecture and diversity preservation mechanism change, as much as the variation of the slowdown factor α, which is used to avoid rapid convergence to a local minima. The Inverted Generational Distance (IGD) is a performance metric that quantifies both convergence and diversity of the Pareto Front (PF) (Zhang *et al.*, 2009). IGD based metrics were used to evaluate the results of each MOHKA version.

The remainder of this paper is structured as follows: theoretical concepts involving multiobjective optimization, comparison metrics, niching procedure diversity preservation mechanism and architecture of the MOHKA versions are presented in Section 2. The objective functions and heat exchangers design problem are depicted in Section 3; and the results are reported in Section 4. Finally, the paper is concluded in Section 5.

2. METAHEURISTIC MULTIOBJECTIVE OPTIMIZATION

In the present section, the concept of optimum for multiobjective optimization problems is stated in Section 2.1. The comparison metrics assumed to evaluate the solutions is introduced in Section 2.2. The Section 2.3. presents the niching procedure mechanism that is adapted to replace the crowding distance operator in the solutions selection. Finally, the MOHKA's architecture and its versions are described in Section 2.4.

2.1 Multiobjective optimization approach

In optimization, the concept of optimum for single objective problems is not suitable to multiobjective problems, since the response offered by the multiobjective approach is not unique, but a set of solutions selected in the eyes of 'Pareto Optimality Theory', which provides the concepts of i) 'Pareto Optimal', where a solution x presents better results for all objectives, when compared with other solutions in the search space; ii) 'Pareto Dominance', where a solution x_1 dominates a solution x_2 ($x_1 \succ x_2$) if at least in one objective, the value presented by x_1 is lower than the handed by x_2, while the other objective values of x_1 are lower or equal in comparison with x_2; iii) 'Pareto Set', which is a set composed by nondominated solutions; iv) 'Pareto Front' (PF), which is the representation of the Pareto Set in the objective space. The formally definitions of the presented concepts can be found on Coelho *et al.* (2007).

2.2 Multiobjective optimization algorithms comparison

Seeking to make a performance comparison among two or more algorithms, evaluation functions are used to quantify the qualities of the PF found by each method. Therefore, for the performance analysis and the comparison of the optimization methods two IGD-dependent metrics were assumed. The IGD calculates the mean square of the shortest distances d_i, from each true Pareto-optimal solution i and the nearest computed solution, for that reason, the IGD is a metric that consider not only the convergence but the distribution of the resulting PF (Zhang *et al.*, 2009). Being n the total number of solutions in the true PF, the IGD is mathematically expressed as (Sierra and Coello, 2005):

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{1}$$

Thus, the two algorithms comparison metrics are i) the integral of the IGD x evaluations curve (IGD_a), where, for each iteration the IGD and the number of evaluations reached is archieved in order to set up a convergence curve to the true PF at the end of the optimization. The axes are normalized and the smaller the area under the curve, greater is the convergence of the method to the real PF and ii) final IGD value (IGD_f) at the end of the optimization, which indicates how nearest is the computed PF to the true PF, with the lowest values being the best. For a better comprehension, Figure 1 illustrates both the IGD_a and IGD_f metrics.

2.3 Niching procedure mechanism

The niching procedure diversity preservation mechanism was adapted to replace the crowding distance operator in the MOHKA versions can be dismembered in three steps, as follows (Deb and Jain, 2014):

1. Determination of reference points on a hyper-plane. The reference points are placed in the objective space before

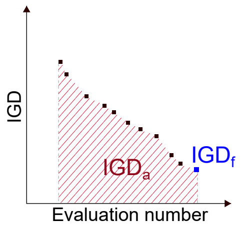

Figure 1. IGD_a and IGD_f metrics representation on a IGD evolution curve

the optimization process starts, being preferably supplied by the optimization designer. In case of lack of preference information, the reference points are allocated in a structured manner as in Deb and Jain (2014), where it is considered a systematic approach (Das and Dennis, 1998), which widely distributes H reference points on a $(M-1)$-dimentional normalized hyper-plane, where M is the number of objectives. Considering p niches or divisions, the number H of reference points is described by:

$$H = \binom{M+p-1}{p} \quad (2)$$

2. Adaptive normalization of population members. For each iteration, the population $X(k)$ is composed by N solutions, each one evaluated according to M objectives, generating the set $Z(k)$ of fit values $z_i^k = \left(z_{i,l}^k, ..., z_{i,M}^k\right)$, where i denotes the position of the fit value in the set $Z(k)$, l indicates the dimention of the i-th solution and k is the iteration number. This notation is maintained until the end of the present subsection. As the reference points are arranged in a normalized hyper-plane, the normalization of solution's fit value z_i^k is important for the subsequent solution-reference point association, as well as contributing to a better Pareto front coverage (Mandal *et al.*, 2018). In order to perform the normalization, for each objective the minimum value presented in the population (z_l^{min}) is identified and an ideal point is defined by $\bar{z}^k = \left(z_1^{min}, z_2^{min}, ..., z_M^{min}\right)$. Then, the solutions of $Z(k)$ set are translated through the subtraction operation $z_{i,l}^{k'} = z_{i,l}^k - \bar{z}_l^k$ and an extreme point for each l objective, denoted by $z^{l,max}$, is identified by finding the solution which yields a minimum for the following scalarizing function (Deb and Jain, 2014):

$$\zeta_l^k = \frac{z_i^{k'}}{w_l} \quad (3)$$

where ζ_l is the result of the scalarizing function when applied to the objective l of the translated solution $z_i^{k'}$, w_l is a weight vector close to the l-th objective axis, which can be contructed as follows $w_l = \mathbf{e}_l + \epsilon \sum_{h=1,h\neq l}^{M} \mathbf{e}_m$, being ϵ a tolerance, $\mathbf{e}_l$ a unit vector in the l-th axis direction and the same for $\mathbf{e}_m$.

The M extreme vectors defined among the $z^{l,max}$ points constitute a linear hyper-plane. For the distribution of Das and Dennis' reference points, the generated hyper-plane is coincident with that of the reference points and the interception of such hyper-plane and the l-th objective axis is a_l. Therefore, the fit values can be normalized, as follows (Deb and Jain, 2014):

$$z_{i_{norm}}^k = \frac{z_i^{k'}}{a_l} \quad (4)$$

3. Association Operation. In the third step, reference lines are drawn considering the reference points and the objective space origin. Then, the perpendicular distance among the reference lines and each solution is calculated. Therefore, the solutions are associated with the reference point whose reference line yields the shortest distance to the solution. Thereby, it is counted the number ρ_h of solutions associated to each niche or reference point h and for solutions with same nondominated rank, the preference is given to the ones associated to niches with lower ρ_h.

2.4 Multiobjective heuristic Kalman algorithm

The MOHKA was proposed by Ayala *et al.* (2017), on top of the HKA formally stated by Toscano and Lyonnet (2012). For a better comprehension of MOHKA, it can be divided in five steps, namely as:

1. Initialization. Where the mean m_k and the standard deviation Σ_k is calculated based on the decision variables bounds, as follows:

$$m_k = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}, \mu_j = \frac{\bar{x}_j + \underline{x}_j}{2}, j = 1, 2, \ldots, n \tag{5}$$

$$\Sigma_k = \begin{bmatrix} \sigma_1^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_n^2 \end{bmatrix}, \sigma_j = \frac{\bar{x}_j + \underline{x}_j}{2}, j = 1, 2, \ldots, n \tag{6}$$

where $\underline{x}_j$ and $\bar{x}_j$ are the lower and upper bounds for the j-th decision variable $x \in \Omega$, being Ω the solution space; and k is the iteration number, initially null.

2. Gaussian generator. Where a set of N new solutions are generated, derived from a normal distribution above the m_k and Σ_k, thus comprising the solutions set $X(k) = \{x_1^k, x_2^k, \ldots, x_N^k\}$.

3. Measurement process. Where the solutions in the $X(k)$ set are evaluated, generating the $Z(k)$ set of same size, but arranged in ascending order. All the nondominated solutions are archieved in a temporary repository until it reaches the total file size of N_a solutions. In case of file fully complete, the nondominance and crowding distance concepts are used to select the solutions that remain in the archive (Deb *et al.*, 2002). In the next iteration, the i-th solution from $X(k)$ that was added to archive has its mean started as x_i^k and the same standard deviation from the solution in the archive which generated it (Ayala *et al.*, 2017). Also, the measurement ξ^k is calculated by the mean of the N_ξ best fit solutions in the archive, and from ξ^k, the variance V^k is obtained, as described below (Ayala *et al.*, 2017):

$$V^k = \frac{1}{N_\xi}\left[\sum_{i=1}^{N_\xi}(x_{i,1}^k - \xi_{1,1}^k)^2, \cdots \sum_{i=1}^{N_\xi}(x_{i,n}^k - \xi_{1,n}^k)^2\right] \tag{7a}$$

$$\xi^k = \frac{1}{N_\xi}\sum_{i=1}^{N_\xi} x_i^k \tag{7b}$$

where N_ξ is the number of candidates chosen from the gaussian generator, $x_{i,j}^k$ denotes the j-th dimension or decision variable of the i-th solution at the instant k.

4. Kalman estimator. Where the Kalman gain L_k and slowdown factor α are calculated through Kalman filter equations depicted below:

$$L_k = \Sigma_k(\Sigma_k + diag(V_k))^{-1} \tag{8a}$$

$$W_k = \left[vec^d\left[(I_k - L_k)\,\Sigma_k\right]\right]^{1/2} \tag{8b}$$

$$a_k = \frac{\alpha \min\left(1, \left(\frac{1}{n}\sum_{i=1}^n \sqrt{v_i^k}\right)^2\right)}{\min\left(1, \left(\frac{1}{n}\sum_{i=1}^n \sqrt{v_i^k}\right)^2\right) + \max\limits_i\left(w_i^k\right)} \tag{8c}$$

where the gaussian generator standard deviation vector is denoted by $S_k = \left(vec^d\left(\Sigma_k\right)\right)^{1/2}$, being $vec^d(.)$ an operator that returns the diagonal components of a matrix as vector, v_i^k and w_i^k are the i-th component of V_k and W_k, respectively. Finally, $\alpha \in (0, 1]$ is a scalar known as slowdown coefficient.

5. Mean and standard deviation update. Finally, in possession of the Kalman gain L_k and slowdown factor α, the mean and standard deviation vectors are updated as follows:

$$m_{k+1} = m_k + L_k\left(\xi_k - m_k\right) \tag{9a}$$

$$S_{k+1} = S_k + a_k\left(W_k - S_k\right) \tag{9b}$$

After the steps 2-5 are done for all solutions in the native archive, the iteration ends and the temporary repository becomes the new native archive. For a better understanding of the algorithm, its pseudocode is presented in Ayala *et al.* (2017).

In order to evaluate if the increase of randomness (subscript 'r'), the new architecture with number of evaluations fixed per iteration (subscript 'p') and the crowding distance replacement by niching procedure (subscripts 'n') have positive impacts on the MOHKA optimization performance, it were proposed the versions depicted next:

1. **Multiobjective heuristic Kalman algorithm random (MOHKAr)**: this version has the same architecture as the MOHKA, but the measure ξ becomes a randomly selected solution from the archive, no longer being an average of the N_ξ best solutions;

2. **Multiobjective heuristic Kalman algorithm random proportional (MOHKArp)**: this version selects the measure ξ in the same way as MOHKAr. However, it is used the mean and standard deviation of the solution selected as ξ measure to initialize a new population in the next iteration, making the iterations have a fixed number of evaluations;

3. **Multiobjective heuristic Kalman algorithm niching (MOHKAn)**: similar to MOHKA, but with the adapted niching procedure being used at the locus of the crowding distance as diversity preservation mechanism;

4. **Multiobjective heuristic Kalman algorithm random niching (MOHKArn)**: similar to MOHKAr, but with the crowding distance operator replaced by the niching procedure as the diversity preservation mechanism;

5. **Multiobjective heuristic Kalman algorithm random proportional niching (MOHKArpn)**: similar to MOHKArp version, but with the niching procedure replacing the crowding distance operator as the diversity preservation mechanism.

3. HEAT EXCHANGER DESIGN

The present section is dedicated to introduce the objective functions used to evaluate the solutions and also to depict both plate-fin (PFHE) and shell-tube (STHE) heat exchangers design. For a better comprehension, the Figure 2 (Sanaye and Hajabdollahi, 2010b,a) presents a schematic representation of both PFHE (left) and STHE (right).

3.1 Objective functions

For the heat exchanger optimization, two objectives are proposed. The first objective is to maximize the effectiveness ϵ and the second is to minimize the total annual cost C_{tot}. Considering a minimization configuration, the first and the second objectives, respectively f_1 and f_2, can be described as follows (Ayala *et al.*, 2016):

$$f_1(x) = 1/\left(1 + \epsilon(x)\right) \tag{10}$$

$$f_2(x) = C_{tot}(x) \tag{11}$$

where x is the solution, constituted of n decision variables and ϵ is effectiveness of the heat exchanger.

The effectiveness and total annual cost are distinctly calculated for PFHE and STHE, the formulation for obtaining these values is briefly presented in Ayala *et al.* (2016).

3.2 Plate-Fin heat exchanger

The PFHE design optimization problem considered a stainless steel material with thermal conductivity $k_w = 18$ W/mK and operating temperature condition of 620 K, pressure of 180 kPa and mass flow rate of 1.45 kg/s on the hot side, and operating temperature condition of 315 K, pressure of 120 kPa and mass flow rate of 1.35 kg/s for the cold side. Other constant values indispensable for the objective functions evaluation as electrical energy price $k_{el} = 20$ \$/MWh, price per unit of area $C_A = 90$ \$/m^2, expoent which provides non-linear growth relative to the area $n = 0.6$, operation time $\tau = 5000$ h/yr, compressor efficiency $\eta = 0.6$, depreciation time $q = 10$ years and interest rate $r = 0.1\%$, as in Sanaye and Hajabdollahi (2010b).

In order to optimize the PFHE design, it was considered seven design variables and their respective bounds, namely as hot fluid flow length ($L_h = [0.2, 0.4]$ m), cold fluid flow length ($L_c = [0.2, 0.4]$ m), no flow length ($L_n = [0.7, 1.2]$ m), fin thickness to fin length ratio ($t_f/L_f = [0.012, 0.048]$), fin height ($H_f = [0.0015, 0.0080]$ m), fin length ($L_f = [0.0020, 0.0035]$ m) and fin pitch ($pt_f = [0.0010, 0.0025]$ m).

3.2 .1 Shell-Tube heat exchanger

The STHE design optimization problem is the same analyzed by Sanaye and Hajabdollahi (2010a) using GA, where is considered an oil cooler. In that case, the oil enters the shell with mass flow rate of 8.1 kg/s and temperature of 351.45 K (78.3°C), while fresh water enters in the tubes with mass flow rate of 12.5 kg/s and temperature of 303.15 K (30.0°C). It was considered a tube arrangement of 90° and, as in Sanaye and Hajabdollahi (2010a), electrical energy price $k_{el} = 0.15$ \$/kW h, life period $ny = 10$ years, rate of annual discount $i = 10\%$, operation time $\tau = 7500$ h/yr and pump efficiency $\eta_p = 0.6$ to evaluate the objective functions.

In order to optimize STHE design, it was considered six design variables and their respective bounds, namely as the inner tube diameter ($d_i = [0.0112, 0.0153]$ m), number of tubes ($N_t = [100, 600]$), length of tube ($L_t = [3, 8]$ m), tube pitch pt to outer tube diameter d_o ($pt/d_o = [1.25, 2.00]$), baffle cut ratio ($bc/D_s = [0.19, 0.32]$), baffle spacing ratio ($bs/D_s = [0.2, 2.4]$), where D_s is the shell diameter.

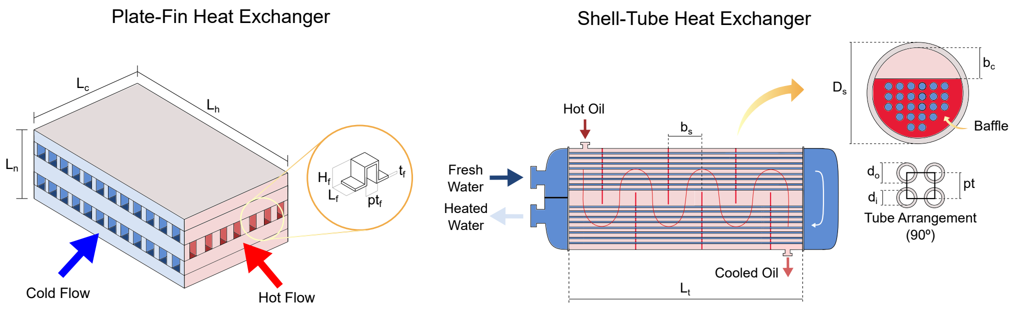

Figure 2. Schematic representation of a plate-fin and a shell-tube heat exchangers

4. RESULTS

In order to study the MOHKA parameter α and the MOHKA versions proposed above, the PFHE, STHE and ZDT1 Zitzler *et al.* (2000) problems are considered.

For the ZDT1 problem it was considered a maximum number of evaluations of 25,000 (Zitzler *et al.*, 2000), for PFHE and STHE problems it was performed 13 optimization tests using MOHKAr in order to study the ideal maximum number of evaluations. The tests considered 30 rounds of optimization, $\alpha = 0.40$, a population size of $N = 100$, $N_\xi = 0.6$ and 1,000, 2,000, 3,000, 4,000, 5,000, 7,500, 10,000, 20,000, 30,000, 40,000, 50,000, 75,000, 100,000 maximum number of evaluations, respectively, and as Figure 3 indicates, when the number o evaluations reached 30,000 there is practically no improvement in the generated PF for PFHE (left) and STHE (right) problems. Therefore, for all numbered tests proposed next the population size N, the maximum number of evaluation $Eval_{max}$ and the number of optimization rounds n_{seeds} were fixed in 100, 30000 and 30, respectively, also it was considered $N_\xi = 6$ for all N_ξ dependent versions.

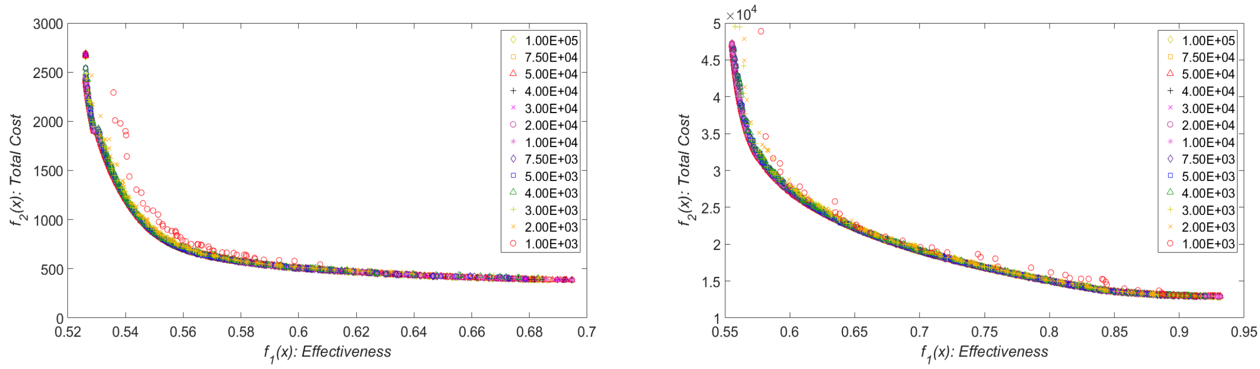

Figure 3. Iteration study for PFHE and STHE problems

For all studied methods the slowdown factor α was varied according to the typical values 0.40, 0.50, 0.60, 0.70, 0.80 and 0.90 (Toscano and Lyonnet, 2012). To facilitate the representation of results in tables, each combination of method and α value was called a test and numbered from 1 to 6 for the original MOHKA, 7 to 12 for MOHKAr, 13 to 18 for MOHKArp, 19 to 24 for MOHKAn, 25 to 30 for MOHKArn and 31 to 36 for MOHKArpn, totalizing 108 tests considering PFHE, STHE and ZDT1 problems.

Seeking to produce a PF said real, the tests from 1 to 36 were concatenated, totalling 20998 and 20997 unique solutions that generated 2452 and 1741 nondominated solution for the PFHE and STHE problems, respectively. After producing the PF said real, the tests from 1 to 36 were redone to obtain the IGD values, allowing to evaluate the IGD_a and IGD_f metrics.

Table 1 presents the means and standard deviations ("Std") for all methods from 1 to 37, when applied to the PFHE, STHE and ZDT1 problems. The best IGD_a and IGD_f means for each method are highlighted in bold

Seeking to evaluate if any of the new versions of MOHKA brought performance increase, the Figures 4, 5 and 6 containing the violin plots for all tests' IGD_a and IGD_f metrics are presented, where the red and black points represent their mean and median, respectively. Also, the Table 2 was generated with the median obtained by each metric and method for the problems of PFHE, STHE and ZDT1, respectively.

Table 1. Mean and standard deviation of the IGD_a and IGD_f metrics for all methods and problems

	PFHE - IGD_a		PFHE - IGD_f		STHE - IGD_a		STHE - IGD_f		ZDT1 - IGD_a		ZDT1 - IGD_f	
Tests	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	0.08071	0.01921	7.11178	0.58126	0.09862	0.02965	87.75893	4.19107	0.32733	0.03384	0.26669	0.03012
2	0.08092	0.01946	7.09336	0.71280	0.09862	0.02965	87.75517	4.19216	0.32729	0.03390	0.26749	0.02773
3	0.08082	0.01943	**7.07986**	0.63271	0.09862	0.02965	87.75512	4.19212	0.32684	0.03399	0.26524	0.02867
4	0.08076	0.01935	7.08948	0.68651	0.09862	0.02965	87.72650	4.20782	0.32675	0.03369	0.26432	0.02917
5	**0.08066**	0.01921	7.10559	0.58307	0.09863	0.02967	87.95238	4.21336	**0.32658**	0.03386	**0.26043**	0.02572
6	0.08084	0.01941	7.14410	0.60368	**0.09862**	0.02965	**87.69463**	4.17887	0.32702	0.03403	0.26109	0.03035
7	0.08151	0.02340	7.32560	0.51024	0.10093	0.02807	87.87557	3.47522	0.32150	0.03026	0.20438	0.03235
8	0.08150	0.02339	7.27459	0.51849	0.10093	0.02808	87.85075	2.98962	**0.32095**	0.02864	**0.19721**	0.03352
9	0.08153	0.02337	7.42823	0.58410	0.10093	0.02807	87.94513	3.50065	0.32106	0.02934	0.20621	0.03567
10	0.08154	0.02349	7.35128	0.50415	**0.10092**	0.02807	87.78711	3.31150	0.32167	0.02834	0.20905	0.03161
11	0.08155	0.02345	7.34596	0.51031	0.10093	0.02809	**87.59678**	3.20180	0.32136	0.02953	0.20746	0.03504
12	**0.08138**	0.02334	**7.19221**	0.62525	0.10093	0.02809	87.71771	3.48119	0.32182	0.02922	0.20508	0.03274
13	0.07900	0.03487	7.34952	0.68981	**0.07952**	0.02265	87.66299	4.45028	0.12920	0.00907	0.13077	0.01213
14	0.07838	0.03771	**7.33197**	0.69534	0.08048	0.02298	88.75991	3.87375	0.12838	0.01161	0.12252	0.01289
15	0.07688	0.03409	7.70273	1.31958	0.07969	0.02231	**87.01330**	5.24734	0.12462	0.01176	0.11306	0.01006
16	0.07739	0.04001	7.38552	0.59545	0.08053	0.02284	88.26105	4.26554	0.12222	0.01215	0.10558	0.01219
17	**0.07611**	0.03645	7.56501	0.67937	0.07980	0.02275	88.26542	4.74387	0.12188	0.01142	0.10082	0.01137
18	0.07765	0.03624	7.36063	0.77870	0.08007	0.02274	88.96298	4.58372	**0.12060**	0.01069	**0.09591**	0.00919
19	0.12072	0.02440	32.87390	14.33120	0.14376	0.03657	430.71277	141.55400	0.33029	0.02938	0.27892	0.04214
20	0.12146	0.02425	36.40087	13.25994	**0.14376**	0.03657	430.71523	141.55175	0.33016	0.02940	0.27940	0.04327
21	0.12085	0.02435	**31.31559**	14.11465	0.14381	0.03650	430.89771	141.35766	0.32996	0.02945	0.27621	0.03895
22	0.12122	0.02376	33.68889	13.92103	0.14389	0.03656	**428.84724**	144.11652	0.32989	0.02929	0.27514	0.04104
23	0.11933	0.02295	34.00893	16.78557	0.14376	0.03657	430.91362	141.33028	**0.32988**	0.02943	**0.27504**	0.03763
24	**0.11921**	0.02285	31.74919	14.30598	0.14376	0.03657	430.91627	141.32728	0.33014	0.02857	0.27787	0.03501
25	0.12181	0.02101	31.92999	22.56989	0.14090	0.03532	405.69336	158.81975	0.32600	0.03062	0.20397	0.03207
26	0.12293	0.02112	35.30762	25.86354	0.14101	0.03519	407.81652	158.75043	**0.32467**	0.03066	0.20190	0.03105
27	0.12431	0.02179	35.57332	32.94169	0.14101	0.03519	407.81254	158.75279	0.32578	0.03054	0.20077	0.03727
28	**0.12142**	0.02286	34.02119	24.18223	0.14101	0.03519	407.81330	158.75202	0.32597	0.03117	0.20823	0.03615
29	0.12241	0.02431	**31.13583**	17.48641	0.14101	0.03519	412.07004	158.03089	0.32577	0.03178	0.20340	0.03896
30	0.12390	0.02321	35.82697	23.96487	**0.14085**	0.03521	**398.76217**	148.53808	0.32521	0.03181	**0.19830**	0.03464
31	**0.16384**	0.05011	68.48640	51.67784	0.23234	0.05710	846.66454	297.14001	0.12977	0.01280	0.12934	0.01500
32	0.16856	0.05909	59.16842	54.37152	**0.22902**	0.05703	**833.75323**	263.03309	0.12474	0.01317	0.11770	0.01030
33	0.17072	0.04534	55.87618	39.29132	0.22940	0.05625	916.21595	314.54644	0.12451	0.01199	0.11412	0.01194
34	0.18779	0.06080	70.08573	57.04002	0.22962	0.06167	868.93923	270.80321	0.12165	0.01275	0.11008	0.00894
35	0.16634	0.05111	**49.85450**	31.74101	0.23115	0.06066	859.19644	267.32397	0.12092	0.01240	0.10345	0.00941
36	0.19156	0.07599	72.32754	69.76732	0.23123	0.05811	875.97880	223.90925	**0.11825**	0.01120	**0.09880**	0.01089

According to Figures 4, 5 and 6 and with the Table 2, it is observed that i) the slowdown factor α do not had a relevant impact on the IGD_a and IGD_f, perhaps for having used the typical α values (Toscano and Lyonnet, 2012), being large enough to avoid local minima (for both IGD_a and IGD_f) or because the iteration/evaluation number is large enough to reach PFs very close to the real ones (for IGD_f), ii) the MOHKA versions that used the niching procedure (tests from 19 to 36) yielded worse results overall, when compared with the versions that used crowding distance, however iii) for ZDT1, MOHKArn and MOHKArpn yielded best results than the original MOHKA and iv) the MOHKArp method (tests from 13 to 18) presented the best performance, for presenting median visually inferior in the IGD_a metric for PFHE and STHE problems and practically tied with MOHKArpn in ZDT1 problem, also a tied performance with the best IGD_f values for all problems.

In order to evaluate the best MOHKArp, it was separately normalized among 0 and 1 each column interval of the Table 2 that corresponds to the MOHKArp IGD_a and IGD_f medians, following by the sum of the row values found and the ranking of the sums, which allows us to state that the MOHKArp with $\alpha = 0.70$ (test 16) was the best method according to both IGD_a and IGD_f metrics. The same process was done with the other MOHKA versions, indicating the MOHKA with $\alpha = 0.90$ (test 6), MOHKAr with $\alpha = 0.50$ (test 8), MOHKAn with $\alpha = 0.60$ (test 21), MOHKArn with $\alpha = 0.60$ (test 27) and MOHKArpn with $\alpha = 0.80$ (test 35) as the best MOHKAs for each version. Therefore, it was generated the Figure 7, which presents the 30 rounds median IGD obtained in each evaluation number during the optimization problems PFHE, STHE and ZDT1, from left to right.

The analysis of Figure 7 considers the best of each version and confirms that for PFHE and STHE problems, the niching-procedure-dependent methods had a worse convergence in comparison with the methods that do not use this feature. This behavior changes in ZDT1 problem where MOHKA and MOHKAr lose, while MOHKArpn earn performance, being comparable to the MOHKArp IGD convergence performance. In short, for PFHE and STHE problems, the MOHKArp has a slightly better convergence than the best MOHKA and MOHKAr, being these two tied. Nonetheless, for ZDT1, MOHKArp has a tied performance to the MOHKArpn and for all problems, MOHKArp converges faster. Despite the best performance, Figure 7 reveals that the standard deviation of the MOHKArp with $\alpha = 0.70$ encompasses the MOHKA with $\alpha = 0.90$ IGD distribution for the PFHE problem, suggesting a statistically non-relevant result. The Wilcoxon rank sum test among the IGD_a and IGD_f medians obtained for tests 6 and 16 (Pohlert, 2014) for R (R Core Team, 2018) yielded p-values that corroborates the suggestion (0.2366 for IGD_a and 0.1294 for IGD_f). However, for

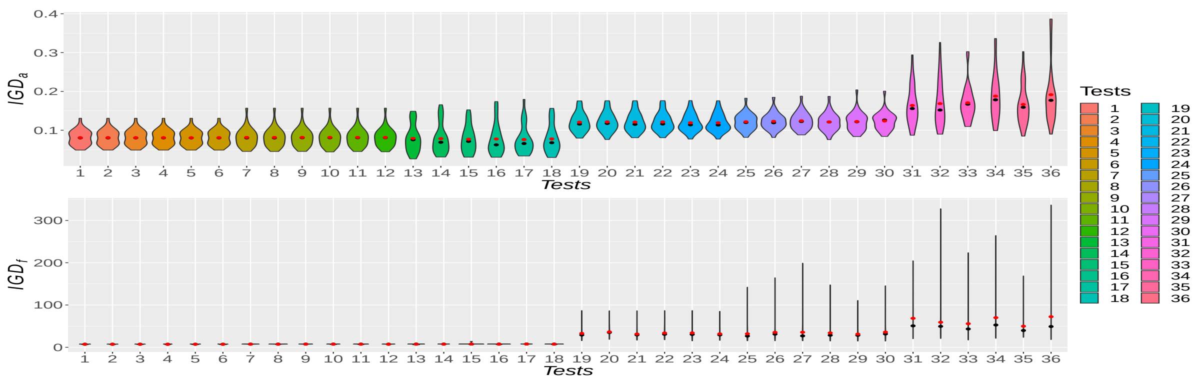

Figure 4. Violin plots for the IGD_a and IGD_f obtained on all tests in the PFHE problem

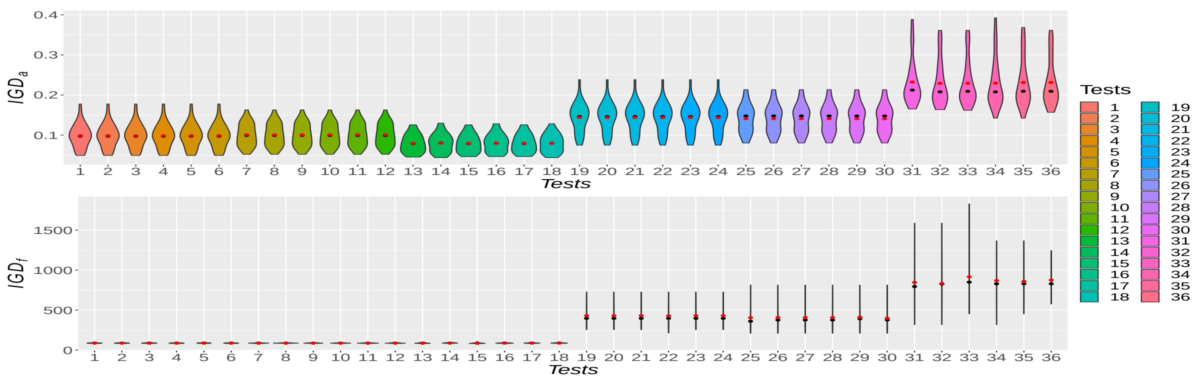

Figure 5. Violin plots for the IGD_a and IGD_f obtained on all tests in the STHE problem

ZDT1 problem the performance enhance was significant (p-value of 1.8626E-09 for both metrics). Finally, for STHE problem the performance improvement was relevant for the IGD_a, but not for IGD_f (p-values of 0.0234 and 0.9032, respectively).

5. CONCLUSION

In this paper, we have suggested new architectures for the MOHKA metaheuristic in order to enhance the method's performance. The results had showed that i) the slowdown factor α did not impact the convergence of the methods, which may have been caused by the use of typical values that were effective in avoiding local minima or by the use of a maximum number of evaluations that allowed obtaining PFs very close to the real ones for PFHE and STHE problems, ii) with the excepetion of the ZDT1 problem, the adapted niching procedure was not effective in substitute the crowding distance operator as a diversity preservation mechanism and iii) the MOHKArp was the version that presented the fastest IGD convergence, although the median of the best MOHKArp ($\alpha = 0.70$) had presented a better performance in the IGD evolution curve when compared to the best MOHKA ($\alpha = 0.90$), the result is not statistically relevant for the PFHE problem for not having reached p-value lower than 0.05.

For future works, it would be interesting to look for ways to improve MOHKArp, one possibility would be to study ways to add randomness to MOHKArp as the optimization follows. Another interesting point would be to evaluate the performance of niching-procedure-dependent codes in many-objectives problems (four or more objectives).

REFERENCES

Ayala, H.V.H., Coelho, L.S. and Reynoso-Meza, G., 2017. "Heuristic Kalman algorithm for multiobjective optimization". *IFAC-PapersOnLine*, Vol. 50, No. 1, pp. 4460–4465.

Ayala, H.V.H., Keller, P., Morais, M.F., Mariani, V.C., Coelho, L.S. and Rao, R.V., 2016. "Design of heat exchangers using a novel multiobjective free search differential evolution paradigm". *Applied Thermal Engineering*, Vol. 94, pp. 170–177.

Bintoro, J.S., Akbarzadeh, A. and Mochizuki, M., 2005. "A closed-loop electronics cooling by implementing single phase impinging jet and mini channels heat exchanger". *Applied Thermal Engineering*, Vol. 25, No. 17-18, pp. 2740–2753.

Coelho, C.A.C., Lamont, G.B. and Veldhuizen, D.A.V., 2007. *Evolutionary algorithms for solving multi-objective problems*. Springer, New York, USA, 2nd edition.

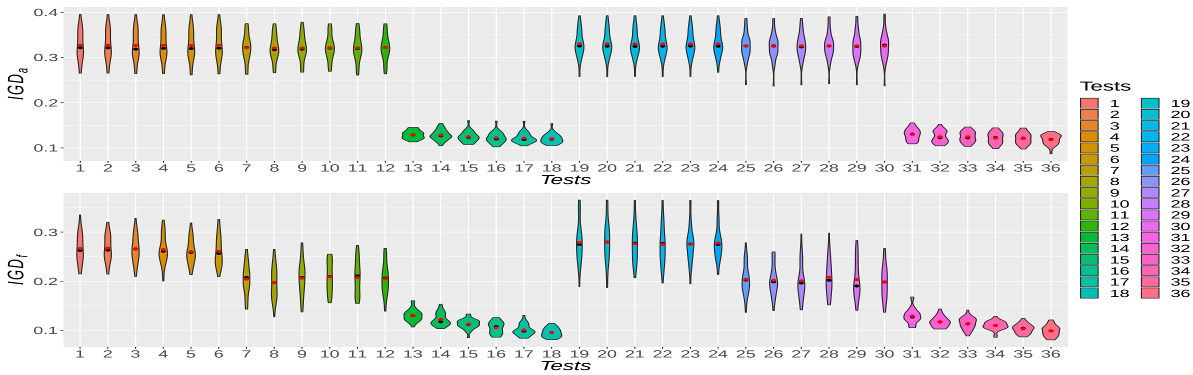

Figure 6. Violin plots for the IGD_a and IGD_f obtained on all tests in the ZDT1 problem

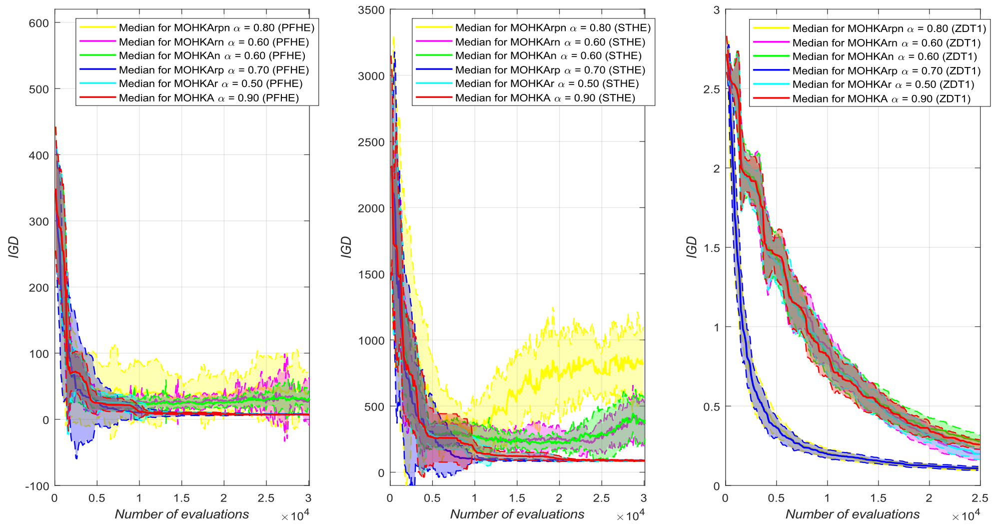

Figure 7. IGD evolution comparison for the best of each MOHKA version

Das, I. and Dennis, J., 1998. "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems". *SIAM Journal on Optimization*, Vol. 8, No. 3, pp. 631–657.

Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: Nsga-ii". *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197.

Deb, K. and Jain, H., 2014. "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints". *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 4, pp. 577–601.

Esfe, M.H., Mahian, O., Hajmohammada, M.H. and Wongwises, S., 2018. "Design of a heat exchanger working with organic nanofluids using multi-objective particle swarm optimization algorithm and response surface method". *International Journal of Heat and Mass Transfer*, Vol. 119, pp. 922–930.

Fabbri, G., 2000. "Heat transfer optimization in corrugated wall channels". *International Journal of Heat and Mass Transfer*, Vol. 43, No. 23, pp. 4299–4310.

Mandal, J.K., Mukhopadhyay, S. and Dutta, P., 2018. *Multi-objective optimization: evolutionary to hybrid framework*. Springer Singapore, SG, 1st edition.

Mohanty, D.K., 2016. "Application of firefly algorithm for design optimization of a shell and tube heat exchanger from economic point of view". *International Journal of Thermal Sciences*, Vol. 102, pp. 228–238.

Patel, V.K. and Rao, R.V., 2010. "Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique". *Applied Thermal Engineering*, Vol. 30, No. 11-12, pp. 1417–1425.

Pohlert, T., 2014. *The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)*. URL `https://CRAN.R-project.org/package=PMCMR`. R package.

R Core Team, 2018. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Sanaye, S. and Hajabdollahi, H., 2010a. "Multi-objective optimization of shell and tube heat exchangers". *Applied*

Table 2. Median of the IGD_a and IGD_f metrics for all methods and problems

	PFHE		STHE		ZDT1	
Tests	IGD_a	IGD_f	IGD_a	IGD_f	IGD_a	IGD_f
1	0.08033	7.27768	0.09726	87.39741	0.32168	0.26276
2	0.08033	7.24413	0.09726	87.34070	0.32137	0.26322
3	0.08034	7.30294	0.09726	87.34071	0.31855	0.26574
4	0.08034	7.20299	0.09726	87.34071	0.32040	0.26092
5	0.08033	7.24402	0.09726	88.17209	0.31973	0.25835
6	0.08033	7.25578	0.09726	87.34142	0.32061	0.25655
7	0.08019	7.45012	0.09915	87.03768	0.32230	0.20775
8	0.08026	7.44444	0.09915	87.03771	0.31727	0.19718
9	0.08068	7.50721	0.09915	87.03774	0.31849	0.20767
10	0.08060	7.49192	0.09901	86.93513	0.32064	0.20976
11	0.08071	7.45964	0.09915	86.78924	0.31989	0.21099
12	0.08081	7.37480	0.09915	86.63860	0.32215	0.20653
13	0.07561	7.57068	0.07896	86.55314	0.12892	0.13035
14	0.06894	7.53459	0.08044	87.85575	0.12686	0.11759
15	0.07126	7.62879	0.07892	85.35071	0.12328	0.11230
16	0.06245	7.52240	0.08003	87.19535	0.11987	0.10779
17	0.06582	7.75752	0.07904	87.77567	0.11854	0.09853
18	0.06791	7.57623	0.07982	88.22505	0.11924	0.09578
19	0.11580	29.09568	0.14629	396.80555	0.32536	0.27492
20	0.11756	34.96224	0.14629	396.80160	0.32494	0.27979
21	0.11528	29.22926	0.14629	396.80171	0.32462	0.27757
22	0.11612	30.72122	0.14629	396.80182	0.32499	0.27660
23	0.11401	30.40658	0.14629	396.80193	0.32518	0.27581
24	0.11349	29.24911	0.14629	396.80610	0.32496	0.27483
25	0.11996	26.60234	0.14775	360.75719	0.32555	0.20248
26	0.11936	31.06591	0.14775	376.04319	0.32580	0.19883
27	0.12244	26.97527	0.14775	376.04323	0.32364	0.19650
28	0.12122	28.52293	0.14775	376.04327	0.32552	0.20208
29	0.12180	28.31938	0.14775	391.16039	0.32452	0.19044
30	0.12631	31.52410	0.14775	376.04336	0.32746	0.19847
31	0.15574	50.71148	0.21227	794.81287	0.13059	0.12744
32	0.15206	49.45806	0.20804	825.65304	0.12241	0.11731
33	0.16705	43.35480	0.20921	849.79440	0.12225	0.11348
34	0.17840	52.70905	0.20787	828.65877	0.12290	0.11000
35	0.15911	39.75508	0.20921	829.27436	0.12162	0.10439
36	0.17710	49.08957	0.20921	828.95251	0.11910	0.09923

Energy, Vol. 30, No. 14-15, pp. 1937–1945.

Sanaye, S. and Hajabdollahi, H., 2010b. "Thermal-economic multi-objective optimization of plate fin heat exchanger using genetic algorithm". *Applied Energy*, Vol. 87, No. 6, pp. 1893–1902.

Seada, H., Abouhawwash, M. and Deb, K., 2019. "Multiphase balance of diversity and convergence in multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, Vol. 23, No. 3, pp. 503–513.

Seada, H. and Deb, K., 2016. "A unified evolutionary optimization procedure for single, multiple, and many objectives". *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 3, pp. 358–369.

Shah, R.K. and Sekulic, D.P., 2003. *Fundamental of heat exchanger design*. John Wiley & Sons, New Jersey, USA, 1st edition.

Sierra, M.R. and Coello, C.A.C., 2005. "Improving pso-based multi-objective ooptimization using crowding, mutation and ϵ-dominance". In *International Conference on evolutionary multi-criterion optimization*. Springer, pp. 505–519.

Tharakeshwar, T.K., Seetharamu, K.N. and Prasad, B.D., 2017. "Multi-objective optimization using bat algorithm for shell and tube heat exchangers". *Applied Thermal Engineering*, Vol. 110, pp. 1029–1038.

Toscano, R. and Lyonnet, P., 2012. "A kalman optimization approach for solving some industrial electronics problems". *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 11, pp. 4456–4464.

Zarea, H., Kashkooli, F.M., Soltani, M. and Rezaeian, M., 2018. "A novel single and multi-objective optimization approach based on bees algorithm hybrid with particle swarm optimization (bahpso): Application to thermal-economic design of plate fin heat exchangers". *International Journal of Thermal Sciences*, Vol. 129, pp. 552–564.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W. and Tiwari, S., 2009. "Multiobjective optimization test instances for the cec 2009 special session and competition". *School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., and School Electr. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487*.

Zitzler, E., Deb, K. and Thiele, L., 2000. "Comparison of multiobjective evolutionary algorithms: empirical results". *Evolutionary Computation*, Vol. 8, No. 2, pp. 173–195.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.