XLI CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU PARANÁ – BRAZIL

A Dynamic Treatment Criterion of Population Sizes in PSO

Samuel C.A. Basílio[1,3], Afonso C.C. Lemonge[2], Érica C.R. Carvalho[3], Patrícia H. Hallak[2]

[1] *Dept. of Computing and Mechanics, Federal Center for Technological Education of Minas Gerais*
558 José Péres street, Leopoldina, MG 36700-000, Brazil
[2] *Dept. of Applied and Computational Mechanics, School of Engineering, Federal University of Juiz de Fora*
[3] *Graduate Program of Computational Modeling, Federal University of Juiz de Fora*
José Lourenço Kelmer street, Juiz de Fora, MG 36036-900, Brazil
samuel@cefetmg.br; afonso.lemonge@ufjf.edu.br; ericacrcarvalho@gmail.com; patricia.hallak@ufjf.edu.br

Abstract. Populational algorithms are strongly dependent on parameters, and among them, the size of the population directly impacts the search for optimized solutions and computational cost. As the population grows, the slice of the inspected search space also tends to expand, allowing new optimums to be discovered. However, the increase in this population also implies an increase in the objective function calls, consequently increasing the algorithms' computational effort and execution time. When the objective function calls are the execution bottleneck, the number of individuals observed at each iteration is decisive for the weight given to exploration and exploitation. In general, the choice of population size happens empirically, through the user's experience. However, the dynamic treatment of population size can be a more interesting choice. In optimization problems that require a simulator, such as optimization in mechanical and structural engineering, the decrease in computational cost is very significant. Moreover, many simulations have high computational costs, motivating the study of a less empirical approach in population size choosing. Here we propose and study an approximation metamodel in the form of a criterion for dynamic treatment of the population size of a Particle Swarm Optimization algorithm applied to mechanical engineering optimization problems. This metamodel considers that particles that are very close and with similar speeds will have similar behavior, tending to the same solution, thus allowing one of the particles to be eliminated. Comparative results are presented using the proposed strategy, showing that it achieved the desired expectations.

Keywords: Structural Optimization Problems, Metamodel-Based Optimization, Particle Swarm Optimization

1 Introduction

In populational algorithms like PSO (Particle Swarm Optimization) [1], population size is crucial for an efficient search process. The larger the population, the greater the slice of the inspected search space, which leads to finding better solutions. However, population size growth also increases the number of fitness function calls, increasing the complexity of the algorithm and its execution time. With the fitness function max calls limitation, the population size growth, even if it does not influence the algorithm complexity, decreases the convergence capacity of the algorithm while increasing the exploration capacity, a fact that may or may not be desirable[2].

In general, the choice of population size happens empirically, through the user's experience, considering characteristics of the algorithm and the problem to be treated [3]. This negligent adjustment can lead to an underutilization of the algorithm both in results and in computational cost [4]. Thus, the dynamic treatment of population size may be a more interesting choice [5].

In this paper, we propose and study an approximation metamodel as a swarm optimization algorithm population size dynamic treatment criterion applied to structural optimization problems. This metamodel considers that close particles and with similar speed should have similar behaviors, tending to the same solution, thus allowing one of the particles to be eliminated.

To assess the performance of the proposed strategy, a set of five examples taken from the optimization literature were performed and compared. The results show that the use of such scheme for dynamic reduction of the population size is advantageous in most scenarios.

2 Proposed Criterion

The original PSO algorithm proposed by Kennedy and Eberhart [6], proposed a population of P individuals who move in the space R^n. Each individual i, also called a particle, represents a possible solution and is formed by two vectors x_j^i and v_j^i, with $i \in \{1, 2, ..., P\}$, $j \in \{1, 2, ..., n\}$, $x_j^i \in R$, $v_j^i \in R$, where x_j^i represents the position of the particle, v_j^i its speed and n the space dimension. Each particle also knows its best position, $pBest$ and the position of the best particle among all $gBest$. This knowledge is called social and cognitive knowledge, respectively. In this way, the position and speed vectors are calculated for each iteration t of the algorithm according to the following expressions:

$$v_j^i(t+1) = v_j^i(t) + c_1.r_1(x_{pBest}^i - x_j^i) + c_2.r_2(x_j^{gBest} - x_j^i) \tag{1}$$

$$x_j^i(t+1) = x_j^i(t) + v_j^i(t+1) \tag{2}$$

where c_1 and c_2 are coefficients that control the influence of social and cognitive knowledge on particles, and r_1 and r_2 are two random values that seek to add non-determinism to the model.

Here we propose a criterion for dynamically reducing the number of p particles during the execution of the PSO algorithm using the tournament system[7].

Let σ be a collision threshold for the position of two particles, a and b any two particles, if $|x^a - x^b| < \sigma$ and $|v^a - v^b| < \sigma$, we will randomly eliminate one of these particles.

The idea behind this modification is that if two particles are very close and with similar speeds, they will behave similarly, tending to the same solution. Even though the coefficients r_1 and r_2 seek to diversify the paths taken by these particles, the social and cognitive knowledge aims to lead them along the same path.

3 Computational Experiments

Five benchmark test problems were carried out to assess the performance of the proposed criterion. To the original Kennedy and Eberhart algorithm[6], an adaptive penalty method for handling constraints [8] and a parameter for inertia control [9] were added. In all problems, for each variation in the parameters, 25 independent runs were performed to provide more statistical reliability to the results. The values of c_1 and c_2 were set as $1,496$[10]. For each problem, a set of runs of the algorithm without the particle reduction criterion was performed. It served as a comparison parameter for the runs where the proposed approach was used. In all cases, executions were carried out with an initial population $P = \{30, 60, 90, 120, 150, 180, 210, 240, 270, 300\}$. The σ values were 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9} and 10^{-10}.

3.1 Tension/Compression Spring design

The purpose of this problem is to reduce the volume of a coil string under a constant tension/compression load as [11]. The total number of calls to the objective function was 36000 in each independent execution. Table 1 presents the best results obtained considering the 25 independent runs for all combinations of initial population and collision thresholds.

From Table 1, we can see that for the Tension/Compression Spring problem regardless of the initial population size, the dynamic reduction of the population size is always advantageous. The results obtained are very close to the best-known result [12], 0.012665. From 10^{-8} the particle collision always occurs at the same time as we can see in Table 2. However, from 10^{-6}, even if the collision threshold decreases, the result obtained is the same.

Table 1. The best solutions found for the Tension/Compression Spring problem considering the initial population size and the collision threshold.

Collision Threshold / Initial Population	No particle reduction	1	10^{-1}	10^{-2}	10^{-3}
30	0,0126982811259	0,0127030311611	**0,0126661508956**	0,0126837778465	0,0126982811259
60	0,0127137251662	**0,0126721396284**	0,012677913757	0,0127047227329	0,0126962690899
90	0,0127270746713	0,0126940928398	**0,0126714182919**	0,0127115319714	0,0127077495493
120	0,0127114113637	**0,012683554753**	0,0126912131431	0,0126927451476	0,0127005028697
150	0,0126960634072	**0,0126761185116**	0,0126871532126	0,0126856657554	0,0127247917023
180	0,0127665776853	**0,0126808217558**	0,0126842431208	0,0126814782792	0,0127037417042
210	0,0127237783055	0,0126893201624	**0,0126777460901**	0,0127075413577	0,0127498629112
240	0,0127091692276	0,0127091258552	**0,012680445816**	0,0127169427027	0,0126873761743
270	0,012773239133	**0,0126775412869**	0,0127073244463	0,0127088563818	0,0127587890458
300	0,0127785340375	**0,0126700251943**	0,012703855212	0,0127344534303	0,0127950225467

Collision Threshold / Initial Population	10^{-4}	10^{-5}	10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}
30	0,0126982811259	0,0126965162286	0,0126982811259
60	0,0127085320046	0,0126959504672	0,0127093633157
90	0,0126959313686	0,0127083040738	0,0127083040738
120	0,0127324587628	0,0127324587628	0,0127324587628
150	0,0127276538722	0,012748757847	0,012748757847
180	0,0127491918422	0,0127491918422	0,0127491918422
210	0,0127375812077	0,0127375812077	0,0127375812077
240	0,0127609060254	0,0127918196685	0,0127918196685
270	0,0127202914273	0,0127202914273	0,0127202914273
300	0,0127529396136	0,0127529396136	0,0127529396136

Table 2. Particle quantity final mean for the Tension/Compression Spring problem considering the initial particle quantity and the collision threshold.

Collision Threshold / Initial Population	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}
30	1,04	1,08	3,96	16,76	21,2
60	1,04	1,32	6,68	29,16	37,56
90	1,04	1,28	6,92	48,32	61,88
120	1,04	1,08	14,28	61,96	78,16
150	1,04	1,16	18,36	93,84	108,52
180	1,04	1,12	16,96	98,68	120,12
210	1,04	1,2	21,72	119,32	139,8
240	1,04	1,12	20,8	143,4	165,68
270	1,04	1,12	24,72	173,4	179,24
300	1,04	1,2	25,16	195,72	208,72

Collision Threshold / Initial Population	10^{-5}	10^{-6}	10^{-7}	10^{-8}, 10^{-9}, 10^{-10}
30	22,88	23,04	24,12	24,12
60	39,88	41,32	41,4	41,76
90	64,84	64,8	64,84	64,84
120	81,92	82,4	82,4	80,44
150	108,4	107,16	107	107
180	120,32	120,24	120,24	120,24
210	139,08	140,72	140,72	140,72
240	167,52	167,52	167,52	167,52
270	180,52	180,52	180,52	180,52
300	209,64	209,88	209,88	209,88

3.2 Speed Reducer design

The objective of this problem is to minimize the weight of a speed reducer as presented in [13]. The number of fitness functions calls was 36000 in each independent execution. Table 3 presents the best results obtained considering the 25 independent runs for all combinations of initial population and collision thresholds.

Table 3. The best solutions found for the Speed Reducer problem considering the initial population size and the collision threshold.

Initial Population \ Collision Threshold	No particle reduction	1	10^{-1}	10^{-2}	10^{-3}	$10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	3000,22367543	3026,69819368	**2996,9476716**	2997,17612869	3000,22367543	3000,22367543
60	3000,11712848	3016,79944912	3005,39137576	**2997,89403302**	3000,11712848	3000,11712848
90	**2999,13099283**	3008,08026164	3007,56153083	3001,04996962	3002,61646461	**2999,13099283**
120	3001,9928708	2997,70245138	**2996,84329213**	2999,54860211	3002,25659372	3001,9928708
150	3005,18839925	**2996,52719254**	2997,53465328	3002,68181446	3001,36488904	3005,18839925
180	3001,54326857	**2996,64745821**	3005,80147339	3003,81193376	3008,59231679	3001,54326857
210	3005,26461488	3026,69029466	**3000,52008988**	3007,69754048	3005,26461488	3005,26461488
240	3007,85272681	3035,63033176	**2997,07851085**	3007,1647863	3013,47559077	3007,85272681
270	3010,59948912	**2996,45862562**	2997,46021276	3004,79387667	3011,01421377	3010,59948912
300	3008,57641118	3026,68857298	3009,59661205	**3007,68382384**	3008,57641118	3008,57641118

Analyzing Table 3, we observe that considering the initial quantity of particles, in 9 of 10 scenarios, the dynamic reduction of the population size was advantageous. Also, from 10^{-4}, the collision threshold's reduction does not influence the particle reduction and the final result. It is because, from this value, there is almost no collision between the particles, as shown in Table 4. The best-known solution [12] to this problem is 2996.3481.

Table 4. Particle quantity final mean for the Speed Reducer problem considering the initial particle quantity and the collision threshold.

Initial Population \ Collision Threshold	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}, 10^{-5}	$10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	1,08	1,16	4,44	29,36	30	30
60	1,04	1,16	14,24	59,52	60	60
90	1,04	1,04	24,04	89,2	89,96	90
120	1,04	1,12	39,64	119,12	119,96	119,96
150	1,04	1,12	56,56	149,16	150	150
180	1,04	1,12	78,12	179,48	180	180
210	1,04	1,12	99,52	209,6	209,96	209,96
240	1,08	1,16	124,88	239,2	240	240
270	1,04	1,12	150,68	269,48	270	270
300	1,04	1,16	182,8	299	299,96	299,96

3.3 Welded Beam design

The objective of this problem is to minimize the cost of a welded beam as [8]. The total number of calls to the objective function was 320000 in each independent execution. Table 5 presents the best results obtained considering the 25 independent runs for all combinations of initial population and collision thresholds.

Table 5. The best solutions found for the Welded Beam problem considering the initial population size and the collision threshold.

Initial Population \ Collision Threshold	No particle reduction	1	10^{-1}	10^{-2}	10^{-3}	10^{-4},10^{-5},10^{-6}, 10^{-7},10^{-8},10^{-9},10^{-10}
30	2,33623026054	**2,3248238694**	2,32486501317	2,32571056872	2,33623026054	2,33623026054
60	2,33116999317	2,32490423873	**2,32472256468**	2,33317094488	2,33679000247	2,33116999317
90	2,33672342083	2,32679827684	**2,32506036747**	2,33241503632	2,33670249933	2,33670249933
120	2,33835045196	2,32680386809	**2,32467725578**	2,33129694735	2,33676825611	2,33972334231
150	2,33315927522	2,32599374944	**2,32505781892**	2,33174563085	2,33315927522	2,33315927522
180	2,33135984688	2,32584835857	**2,32488075268**	2,3288078981	2,33838260156	2,33912564724
210	2,33009930785	2,32774634233	**2,32773697039**	2,33478296813	2,33528581999	2,33528581999
240	2,34460953053	2,33125733726	**2,32457026935**	2,33692346795	2,33269353811	2,32787151367
270	2,33866787114	**2,32505950236**	2,32785657702	2,33811608771	2,33388101311	2,33388101311
300	2,34231529814	2,32848411148	**2,32530207171**	2,33594284367	2,33987980916	2,33838330789

Analyzing Table 5, we can observe that regardless of the initial population size, the dynamic reduction of the population size was always advantageous. Again, from 10^{-4}, the reduction of the collision threshold does not influence the final result. From 10^{-6}, the collision threshold reduction does not influence population size reduction, as we can see in Table 6. The best-known result [14] for this problem is 1.724866.

Table 6. Particles quantity final mean for the Welded Beam design considering the initial population size and the collision threshold.

Initial Population \ Collision Threshold	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	$10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	1	1	6,8	29,64	29,96	29,96	29,96
60	1,08	1	17,56	58,92	59,72	59,72	59,72
90	1	1	25,52	88,72	89,6	89,64	89,64
120	1	1	32,96	117,84	119,36	119,36	119,4
150	1	1	44,68	145,96	148,88	148,92	148,92
180	1	1	58,24	176,6	178,56	178,64	178,64
210	1	1	64,8	205,6	207,8	207,92	207,92
240	1,08	1	78,04	235,08	238,2	238,36	238,36
270	1,04	1	86,56	263,04	267,2	267,24	267,24
300	1,04	1	93,48	293,48	296,68	296,68	296,68

3.4 Pressure Vessel design

The objective of this problem is to minimize the weight of a cylindrical pressure vessel with two spherical caps as [15]. There were 80000 calls to the objective function in each independent execution. Table 7 presents the best results obtained considering the 25 independent runs for all combinations of initial population and collision thresholds.

Table 7. The best solutions found for the Pressure Vessel design considering the initial particle quantity and the collision threshold.

Initial Population \ Collision Threshold	No particle reduction	1	10^{-1}	10^{-2}	10^{-3},10^{-4},10^{-5},10^{-6}, 10^{-7},10^{-8},10^{-9},10^{-10}
30	6078,96835992	**6059,71433517**	6111,53815168	6078,96835992	6078,96835992
60	6149,94262999	**6060,48217516**	6135,20583439	6149,94262999	6149,94262999
90	6121,26125975	**6059,74965892**	6229,48001112	6121,26125975	6121,26125975
120	6137,08748252	**6068,76732361**	6147,77479361	6137,08748252	6137,08748252
150	6114,31473115	**6063,66464809**	6082,89808163	6114,31473115	6114,31473115
180	6176,32880337	**6064,16114343**	6125,4298469	6194,31879251	6176,32880337
210	6114,57367747	**6059,75760896**	6147,43220529	6114,57367747	6114,57367747
240	6135,95859167	**6059,72080164**	6202,80608388	6135,95859167	6135,95859167
270	6136,51274403	**6059,80794369**	6129,60459092	6136,51274403	6136,51274403
300	6086,97908449	**6076,81719046**	6188,25534921	6103,69344382	6103,69344382

Analyzing Table 7, we observe that, again, independent of the initial population size, the dynamic reduction of the population size was always advantageous. In this problem, the collision threshold reduction stopped influencing particle reduction from 10^{-3}, as shown in Table 8. The best-known result [16] for this problem is 5849.7617.

Table 8. Particles quantity final mean for the Pressure Vessel design considering the initial particle quantity and the collision threshold.

Initial Population \ Collision Threshold	1	10^{-1}	10^{-2}	$10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	1,04	23,96	29,88	30
60	1,04	52,36	59,88	60
90	1,04	79,2	89,92	89,96
120	1	105,92	119,88	120
150	1,04	133,28	149,8	149,96
180	1,08	162,88	179,68	179,88
210	1	191,28	209,72	209,8
240	1,08	219,48	239,36	239,76
270	1,08	247,28	269,2	269,76
300	1,12	275,36	299,08	299,4

3.5 Cantilever Beam design

The objective of this problem is to minimize the volume of a Cantilever Beam as shown in [8]. There were 35000 calls to the objective function in each independent execution. Table 9 shows the best results obtained considering the 25 independent runs for all combinations of initial population and collision thresholds.

Table 9. The best solutions found for the Cantilever Beam problem considering the initial population size and the collision threshold.

Initial Population \ Collision Threshold	No particle reduction	1	10^{-1}	10^{-2}	$10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	71466,7701795	188477,838597	**71301,7019544**	71466,7701795	71466,7701795
60	72048,7558079	188477,838597	**71547,7815753**	72048,7558079	72048,7558079
90	**72699,7118653**	84606,5502359	72803,6020529	**72699,7118653**	**72699,7118653**
120	81931,7457655	81835,0775044	**72904,9286324**	79440,2146461	81931,7457655
150	75896,588014	188477,838597	**72606,5085339**	75896,588014	75896,588014
180	**73013,3054734**	114861,241775	75405,4941606	73188,375981	**73013,3054734**
210	86550,2816143	88019,7024759	**73071,1013437**	86550,2816143	86550,2816143
240	**75630,4802142**	161616,333813	80595,6731905	80974,9671788	**75630,4802142**
270	85167,8530706	112257,340492	**79927,2314887**	85167,8530706	85167,8530706
300	74785,5966223	133998,643981	**73862,0904634**	77937,979454	74785,5966223

Analyzing Table 9, we observe that in 7 of the ten scenarios, the dynamic reduction of the population size was advantageous. In the other three scenarios (initial population of 90, 180, and 240), even though the result without particle reduction was the best, this was equated to some with particle reduction. It is also interesting to note in this problem that the collision threshold with value 1, generated an excessive and premature reduction of the particles causing an unwanted convergence to local minimums. For this problem, the best-known result [17] is 63893.52.

Table 10. Particles quantity final mean for the Cantilever Beam design considering the initial population size and the collision threshold.

Initial Population \ Collision Threshold	1	10^{-1}	10^{-2}	10^{-3}	~$10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}$
30	1,08	20,92	29,96	30	30
60	1,08	43,96	59,8	60	60
90	1,16	65,64	89,64	90	90
120	1,12	87,56	119,44	119,96	120
150	1,12	108,68	149,56	150	150
180	1,12	129,2	179,48	179,96	180
210	1,08	150,76	209,64	210	210
240	1,08	172,12	239,32	239,96	240
270	1,08	196,52	269,16	270	270
300	1,08	218,4	298,88	300	300

4 Concluding remarks and future works

In this work, we propose and study an approximation metamodel in the form of a particle reduction criterion for particle swarm algorithms. The expected and confirmed behavior was that if two particles are very close and similar speeds, they will also have similar behavior, tending to the same solution, generating a redundancy in the analyzed search space. With the elimination of one of these particles, we no longer waste calls to the objective function and achieved slightly better results than when we let these particles survive until the end of the algorithm's execution.

A more detailed analysis of the collision threshold is still needed. In the experiments carried out, a single value for the collision threshold was used for the speed and position of all decision variables. It is likely that the choice of different values according to the amplitude of each decision variable's domain generates more significant results as well as the use of different thresholds for the particles position and speed.

As the objective of this work was to confirm the hypothesis that the reduction of particles during the execution of a PSO would be attractive, the algorithm used as a comparison was simple and sometimes presented worse results than the best already knowns. However, experiments with more refined PSO modifications can also generate more encouraging results.

An analysis of the solution's stagnation would also be interesting since the main objective of the proposed strategy is to avoid waste in calls to the objective function. Analyzing the solution's stagnation during the execution of the algorithm can allow the execution to end even before all calls to the objective function are made or even the reinsertion of particles to feed the search for new optimums.

Finally, it is expected to show in the future that the proposed metamodel is also efficient in other problems classes, such as problems that require a simulator, where the formulation is not explicit and in other knowledge areas problems, besides structural optimization.

Acknowledgements. The authors wish to thank the reviewers that helped the quality of the paper, CNPq (306186/2017-9), and CAPES (finance code 001), Centro Federal de Educação Tecnológica de Minas Gerais, Universidade Federal de Juiz de Fora – Programa de Pós-Graduação em Modelagem Computacional, for their support.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] Kennedy, J. & Eberhart, R. C., 2001. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[2] Ser, J. D., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello], C. A. C., & Herrera, F., 2019. Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, vol. 48, pp. 220 – 250.

[3] Isiet, M. & Gadala, M., 2020. Sensitivity analysis of control parameters in particle swarm optimization. *Journal of Computational Science*, vol. 41, pp. 101086.

[4] Piotrowski, A. P., Napiorkowski, J. J., & Piotrowska, A. E., 2020. Population size in particle swarm optimization. *Swarm and Evolutionary Computation*, vol. 58, pp. 100718.

[5] Chen, D. & Zhao, C., 2009. Particle swarm optimization with adaptive population size and its application. *Applied Soft Computing*, vol. 9, n. 1, pp. 39 – 48.

[6] Eberhart, R. & Kennedy, J., 1995. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43.

[7] Eiben, A. E. & Smith, J. E., 2015. *Introduction to Evolutionary Computing*. Springer Publishing Company, Incorporated, 2nd edition.

[8] Lemonge, A. & Barbosa, H., 2004. An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical Methods in Engineering*, vol. 59, pp. 703 – 736.

[9] Shi, Y. & Eberhart, R., 1998. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pp. 69–73.

[10] Clerc, M. & Kennedy, J., 2002. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, vol. 6, n. 1, pp. 58–73.

[11] Bernardino, H., Barbosa, H., & Lemonge, A., 2006. Constraint handling in genetic algorithms via artificial immune systems.

[12] Aragón, V., Esquivel, S., & Coello, C., 2010. A modified version of a t-cell algorithm for constrained optimization problems. *International Journal for Numerical Methods in Engineering*, vol. 84, pp. 351 – 378.

[13] Mezura-Montes, E., Coello Coello, C. A., & Landa-Becerra, R., 2003. Engineering optimization using simple evolutionary algorithm. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pp. 149–156.

[14] Kaveh, A. & Talatahari, S., 2010. A novel heuristic optimization method: Charged system search. *Acta Mechanica*, vol. 213, pp. 267–289.

[15] Coello, C. A. C., 2000. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, vol. 41, n. 2, pp. 113 – 127.

[16] Mahdavi, M., Fesanghary, M., & Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, vol. 188, n. 2, pp. 1567 – 1579.

[17] Gandomi, A. H., Yang, X.-S., & Alavi, A. H., 2011. Mixed variable structural optimization using firefly algorithm. *Computers Structures*, vol. 89, n. 23, pp. 2325 – 2336.