

PAPER • OPEN ACCESS

Choice of fitness functions and parameter settings in Genetic Algorithms for analysis of induction motors

To cite this article: Stela Stoykova and Vasil Spasov 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **618** 012024

View the [article online](#) for updates and enhancements.

You may also like

- [Evolution of genetic redundancy: the relevance of complexity in genotype-phenotype mapping](#)
Nen Saito, Shuji Ishihara and Kunihiro Kaneko
- [Determining the parameters of induction motors by Genetic Algorithms](#)
Stela Stoykova and Vasil Spasov
- [GeneticKNN: a weighted KNN approach supported by genetic algorithm for photometric redshift estimation of quasars](#)
Bo Han, Li-Na Qiao, Jing-Lin Chen et al.



HONOLULU, HI
Oct 6–11, 2024

Abstract submission deadline:
April 12, 2024

Learn more and submit!



Joint Meeting of

The Electrochemical Society
•
The Electrochemical Society of Japan
•
Korea Electrochemical Society

Choice of fitness functions and parameter settings in Genetic Algorithms for analysis of induction motors

Stela Stoykova¹ and Vasil Spasov²

¹Industrial Engineering speciality, Technical University - Sofia, Branch Plovdiv

²Department of Electrical Engineering, Technical University - Sofia, Branch Plovdiv

E-mail: vasilspasov@yahoo.com

Abstract. The paper proposes methods for enhancing the accuracy and speed of Genetic Algorithms when determining the parameters of induction motors. A systematic study of the impact of genetic operators and algorithm parameters on optimization process performance is done. New fitness functions are developed containing additional quantities such as the input power of motors. Various genetic operator types and constraints are analysed providing an insight into a specific settings choice. Special attention is given to the factors generating stochastic noise and the ways to eliminate it. To evaluate the effectiveness of the proposed fitness functions and parameter settings, ten combinatorial optimization experiments are conducted on two types of induction motors. The results show that adequate fitness functions, combined with proper genetic operators setups, can significantly enhance the accuracy and speed of Genetic Algorithms while reducing the required input data.

1. Introduction

The notion of Genetic Algorithms (GA) optimization dates back to the principles of natural evolution introduced by Charles Darwin in his work *The Origin of Species*. The foundation of most modern GA optimization approaches is the basic principle of survival of the fittest in biological evolution.

The smallest operational unit of a GA is a chromosome or an individual that consists of a set of binary parameters called genes. The aim of GA is to optimize the individual solutions guided by a fitness function. Typically Genetic Algorithms incorporate the following genetic operators: initialization, selection, recombination, mutation and replacement [1]. A successive iteration through these processes allows for better fitness of individual solutions to be attained.

Initialization most commonly occurs with an initial population of randomly generated solutions. The optimization of solutions is performed in generations. A set of solutions is selected based on their fitness in each generation. The selected solutions are then modified to form a new population of fitter solutions. The new population is used in the next iteration. Termination of the algorithm occurs when either a maximum number of generations is reached, or a satisfactory fitness value is attained [2].

This paper presents a systematic study of the influence of the genetic operators and the algorithm parameters (fitness function and population size) on the optimization process accuracy and speed. For this purpose two new fitness functions are developed and various combinations of genetic operators and algorithm parameters are compared. The comparison is based on the following criteria: accuracy, computation time and convergence.

To verify the efficacy of the proposed new fitness functions and parameter settings, two types of induction motors are analysed in the paper. The results show that the new fitness functions, implemented with appropriate parameter settings, greatly improve the accuracy and speed of the analysis. In addition, less input data is needed as compared to the conventional GA optimization.



2. GA model and development of new fitness functions

To determine the parameters of induction motors, the T-shaped equivalent circuit is used [3, 4]. Two squirrel cage induction motors are analysed- the T80B4 type and the AO-90S-4 type. These motors are widely used in various industrial and household applications. Details of the motors and the nameplate data are given in [3].

The fitness function is an evaluation function that is used to summarize how close a given solution is to achieving the set aim. The fitness not only indicates how good the solution is, but also shows how close the chromosome is to the optimal one.

Since the fitness function is always problem dependent, its appropriate design for the specific aim is a major concern. Problems occurring due to inappropriately designed fitness function include difficulty in algorithm convergence or finding local and not global solutions.

The fitness function has to be concise and fastly computable. Since a typical Genetic Algorithm must be iterated many times in order to produce a viable solution, the execution speed is very important. In our case of multiobjective optimization the fitness function is even more difficult to determine as it should seek optimal solutions corresponding to all objectives.

A widely used fitness function (called also objective function) for determining the parameters of induction motors is [3, 5]:

$$F_{obj} = \sum_{i=1}^n \left(\frac{\cos \varphi_{c,i}}{\cos \varphi_{m,i}} - 1 \right)^2 + \sum_{i=1}^n \left(\frac{I_{lc,i}}{I_{lm,i}} - 1 \right)^2. \quad (1)$$

In (1) $I_{lc,i}$ and $\cos \varphi_{c,i}$ are the computed values of current and power factor. $I_{lm,i}$ and $\cos \varphi_{m,i}$ are the measured values [6] and n designates the number of electrical input data sets of the motor.

The aim of GA is to minimize the error of F_{obj} . In the present work the value of the fitness function is set to zero and in this way its global minimum is found. To obtain relative error within 1%, the conventional fitness function (1) requires at least three data sets, i.e. three points from the motors load curves. With two data sets the maximum relative error is about 3% [3].

To improve the accuracy and reduce the number of required data sets, two new fitness functions are developed in this paper. The first developed fitness function FF1 is the reciprocal of the sum of modules of the type:

$$FF_{s1} = \frac{1}{F_1^2 + F_2^2}. \quad (2)$$

For each data set a new module of the type FF_{s1} is added.

F_1 and F_2 in (2) are functions of the computed and measured stator current and input power:

$$F_1 = \frac{I_{lc} - I_{lm}}{I_{lm}}; \quad F_2 = \frac{P_{lc} - P_{lm}}{P_{lm}}. \quad (3)$$

The second developed fitness function FF2 is the reciprocal of the sum of modules of the type:

$$FF_{s2} = \frac{1}{F_1^2 + F_2^2 + F_3^2}. \quad (4)$$

For each data set a new module of the type FF_{s2} is added.

F_3 in (4) is defined as a function of the power factor:

$$F_3 = (\cos \varphi_c - \cos \varphi_m) / \cos \varphi_m. \quad (5)$$

It should be noticed that the module F_3 in the second fitness function FF2 supplies a supporting data input that balances out the selection-produced stochastic noise (see Section 4). The new data points, obtained by the inclusion of F_3 , facilitate more accurate determination of globally optimal solutions.

3. Choice of parameter settings and convergence of GA

3.1 Initial population size

The quality of a solution is directly dependent on the initial population size [7]. A large population size guarantees higher diversity in the population and exploration of a more encompassing search space. Thus Genetic Algorithms have a greater opportunity to converge to a global optimum.

Large initial population sizes, however, require a considerable amount of computation time for evaluation of the fitness of the possible solutions. Thus in selecting appropriate population size, a trade-off must exist between efficient exploration of the search space and the computation resources available [7].

If the information for resuming a run of GA has been saved for a specific set of parameters, the initial population of a new run can be replaced by the final population of the previously saved run. This approach allows search to be localized in ranges of the search space that are known to produce optimal solutions. It, however, limits the range of possible solutions and can cause premature convergence to a locally optimal solution instead of a globally optimal one.

The presented GA setting combinations, detailed in Section 4, use randomly generated initial population to allow for diversity of possible solutions.

3.2 Selection and crossover

Four types of selection are used in the present study: tournament, roulette wheel, stochastic uniform sampling and stochastic remainder. Next their main characteristics, advantages and disadvantages are discussed.

Tournament selection has several benefits: it is efficient to code, applicable to parallel architectures and allows easy adjustment of the selection pressure through modification of tournament size. It, however, requires considerably more computational time than other selection techniques. GA achieve low error rates but are slow to converge on optimal solutions.

Roulette-wheel (or fitness proportionate) *selection* allows for weaker solutions to survive the selection process. This, however, can be considered an advantage, as though a solution may be weak, it may include some genes which could prove useful following the recombination process. The lack of guarantee that weaker solutions will be eliminated in this selection process generates stochastic noise and optimal solutions for some ranges of the search space may not be attained.

Stochastic uniform sampling is similar to roulette-wheel selection as individuals are assigned contiguous segments of a line, where the size of the segment corresponds to the individuals fitness. Pointers are placed over the line at equal distances from each other. The number of individuals to be selected equals the number of pointers.

The number m of individuals to be selected determines the distance between the pointers as $1/m$. The position of the first pointer is a randomly generated number in the range $[0, 1/m]$.

While roulette-wheel selection chooses solutions from the population by repeated random sampling, stochastic uniform sampling utilizes a single random value for sampling of all the solutions by choosing them at evenly spaced intervals. This approach ensures that stochastic uniform sampling produces solutions closer to the globally optimal and eliminates stochastic noise.

This selection technique is applicable to any number of selections but is preferably used in situations where more than one sample is required.

Stochastic remainder selection chooses chromosomes from the population based on their relative fitness [8]. The relative fitness λ of a chromosome is the ratio of the fitness associated with the individual fitness f_i and the average fitness f_{avg} of the population:

$$\lambda = f_i / f_{avg} . \quad (6)$$

In the first stage of the selection solutions are chosen deterministically based on the integer part of λ . In the second stage the free places in the mating pool are filled up by the roulette-wheel selection on the remaining fractional part. If the value of λ for an individual is 1,4 that individual is listed once in the mating pool because the integer part is 1. After all the individuals have been assessed and

assigned based on the integer part of their relative fitness, the roulette-wheel selection commences. The probability that an individual is chosen in this stage is proportional to the fractional part of λ .

Two types of crossover (recombination) are implemented in the present study: scattered and constraint dependent.

Scattered crossover uses a random binary vector. It selects the genes where the vector is 1 from the first parent and the genes where the vector is 0 from the other parent and introduces them into the offspring. A recombination of the characteristics of the parent individuals generates the child. This approach produces better results when applied to problems without linear constraints.

As shown in Section 4, scattered crossover produces low error rates with minimal computational time when combined with roulette-wheel selection. Due to the scattered crossover principle of generating offspring as a mixture of the genes of parent solutions, the stochastic noise effect of roulette selection is minimized.

Constraint dependent crossover is the default function when solving linear constraint problems. It generates offspring by taking a weighted average of the parents [9].

3.3 Mutation and constraints

The mutation operator is critical to the success of genetic algorithms since it determines the search directions and avoids convergence to local optima. A gene is selected randomly from the chromosome and undergoes a change to diversify the solution pool. Two types of constraint tolerant mutation operators are implemented in this study- adaptive feasible and constraint dependent.

Adaptive feasible mutation is the default with constraint bound problems. The mechanism of operation is based on randomly generated directions adaptive with respect to the last successful or unsuccessful generation [10]. Problem specific linear constraints set the direction and mutation probability.

Mutation probability controls the ratio of new individuals to the population size. Generally, it determines the rate of occurrence of mutation with consideration of the above-mentioned ratio.

This algorithm parameter allows for individuals with high fitness that are lost during optimization to be returned in the population. With too high mutation rate some good properties of the individuals are lost due to frequent random changes to the chromosomes. Too low mutation rates cause a decrease in the population diversity and possibility of converging on local optimum is high. The default mutation rate of 0,1 is used in the presented GA combinations.

Constraint dependent mutation is applied in our study in a combination with non-problem specific constraints: function and tolerance constraint.

Function constraint can be utilized in customizing the range of the stopping criteria of the algorithm with respect to the relative change in the best fitness function value over the stall generations.

Tolerance constraint is not implemented as a stopping criterion. It determines the general feasibility of the solutions generated by the algorithm in the last iterations with respect to linear constraints and exacts a change in direction if necessary.

3.4 Convergence of GA

The main disadvantage of most optimization algorithms is that after they find a local minimum, they are trapped and do not converge further to a better globally optimal solution. Genetic Algorithms possess the ability to locate and converge to globally optimal solutions if the appropriate parameter settings are applied.

To assess the ability of the algorithm to provide globally optimal solutions, the coefficient of variation C_v is introduced for every genetic algorithm parameter combination at the end of each run. The coefficient is obtained by the formula $C_v = \sigma / \bar{x}$, where σ is the standard deviation of the given parameter and \bar{x} is its mean value [11, 12]. The mean value and σ are computed as follows:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i; \quad \sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}}. \quad (7)$$

Here X_i is the set of end configurations for the run and n is the number of runs.

An average value of C_v for each combination is obtained over multiple runs. Higher C_v means more diversity in individual genes, hence poor convergence. C_v is not affected by gaps in parameter ranges at boundary conditions since the affected σ and \bar{X} cancel each other out in the ratio.

4. Results and discussion

The equivalent circuit parameters of the T80B-4 and the AO-90S-4 type squirrel cage induction motors are estimated using the new fitness functions and various combinations of genetic operators discussed in Section 3. The obtained GA results for the two motor types are compared to the analytical data sets shown in Table 1. The accuracy is measured by the relative error:

$$\varepsilon = (X_{GA} - X_{an}) / X_{an}. \quad (8)$$

In (8) X_{GA} is the value of the parameter computed by the GA optimization and X_{an} is the analytical value of the same parameter from Table 1.

Table 1. Data sets used in GA for the motors.

Motor type	Stator current [A]	Slip	Power factor
AO-90S-4	2,58	0,06	0,63
	3,36	0,10	0,72
T80B-4	2,06	0,06	0,59
	2,56	0,10	0,70

The two new fitness functions FF1 and FF2 are applied to the induction motors with different parameter settings. Ten combinatorial optimization experiments, designated as GA1 to GA10, are presented in Table 2.

Table 2 shows the induction motors equivalent circuit parameters estimated by GA as well as the relative errors with regard to the analytical values. The relative errors are for tests with two data sets, shown in Table 1.

The induction motors parameters are: stator resistance R_1 and stator leakage reactance X_1 , rotor resistance R_2 and leakage reactance X_2 (both referred to stator), and magnetizing reactance X_m .

First the new fitness functions are discussed. Proof of the positive effect of appropriately chosen auxiliary data modules in the fitness functions on stochastic noise occurrences is the performance of GA7. Even though this GA parameter combination utilizes roulette wheel selection that produces stochastic noise, it yields one of the lowest coefficients of variation (figure 1). Thus GA7 has high convergence, high speed performance, i.e. low computation (CPU) time and low errors (Table 2). This is due to the use of the fitness function FF2 with the balancing auxiliary data module F_3 .

The computations in this paper are performed on a quad core Intel Pentium Processor N3710, 1.6 GHz and 4GB RAM.

To evaluate the effect of the population size, simulations with various population sizes are carried out (figure 2). Population size selection is problem specific and as a result fitness function dependent. The combinations used (GA1 and GA6) correspond to the two fitness functions FF1 and FF2.

A scale of the best fitness of solutions is used. The actual fitness values for GA are not given and the results are scaled by a factor of 1E6. Best fitness near zero indicates poor performance whereas best fitness of one indicates excellent convergence.

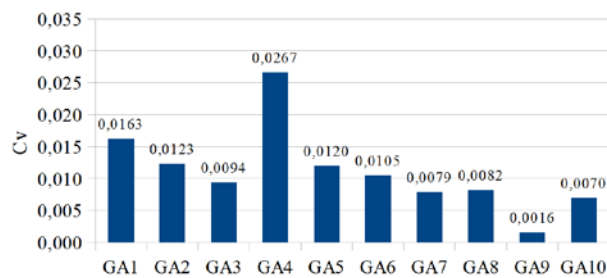


Figure 1. Values of the coefficient of variation.

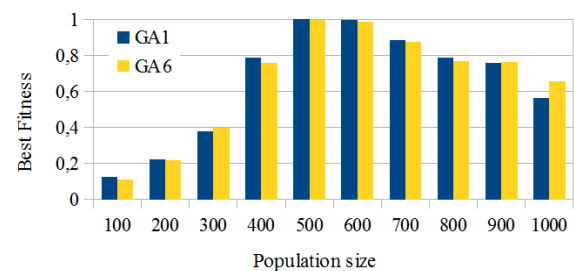


Figure 2. Effect of the population size.

Figure 2 shows that standard population sizes of 250 and below result in premature termination of the GA. Thus optimal solutions are not reached for the given fitness functions FF1 and FF2. Initial population sizes in the range of 350 to 750 result in termination due to the satisfactory value of fitness function attained.

An indication of good exploration of the search space is the high success rate of reaching the global optimum with population sizes above 500. Population sizes larger than 600 do not yield noticeable improvement of the solution. Hence, a mean value of population size 550 that allows converging to an optimal solution without excessive computation time is used further in this paper.

Next the parameter settings and convergence of GA are analysed. GA2 and GA5 that implement tournament selection have the highest CPU times (Table 2). Combinations GA2 and GA5 achieve low error rates but are slow to converge.

The roulette-wheel selection in GA3 does not eliminate weaker solutions. As a result this selection generates stochastic noise (figure 3). Hence the search space may not be fully explored.

There is an obvious difference in the effectiveness of GA3 and GA7, both of which use roulette-wheel selection. GA3 uses FF1 and GA7 uses FF2 with the auxiliary module F3. It is evident that GA7 outperforms GA3. GA7 converges faster (figure 1), has significantly lower CPU time and is more accurate (Table 2). Figure 4 shows the convergence history of GA7 which lacks stochastic noise due to the auxiliary input data in F3.

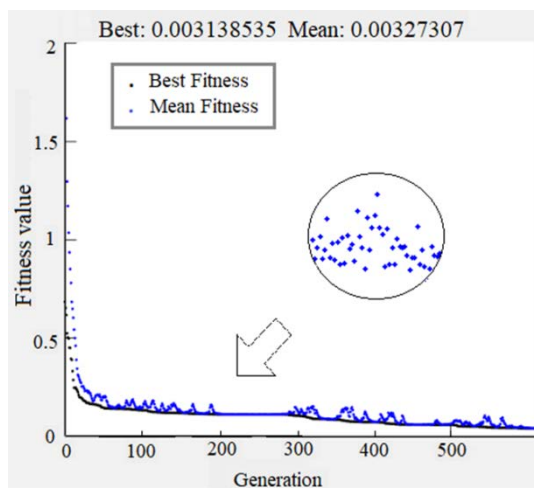


Figure 3. Stochastic noise with roulette-wheel.

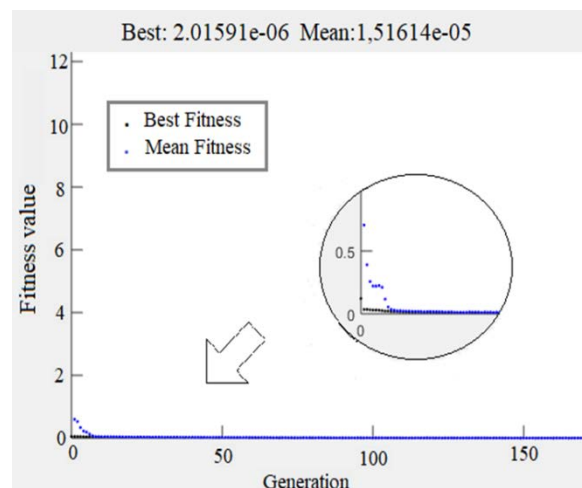


Figure 4. Elimination of stochastic noise by FF2.

The comparison of GA1 to GA3, utilizing FF1, and GA6 to GA7, based on FF2, shows that GA with stochastic uniform sampling have slower rate of convergence than those with roulette-wheel. The final error assessment, however, indicates that the stochastic uniform sampling based GA produce considerably lower errors than the roulette-wheel based GA.

Table 2. Combinatorial optimization experiments for the two induction motors.

		GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10
Motor type	AO-90S-4	+		+		+		+			+
	T80B-4		+		+		+		+	+	
Fitness function	FF1	+	+	+	+						+
	FF2					+	+	+	+	+	
Selection	Tournament		+			+					
	Roulette			+				+			
	Stochastic uniform	+					+			+	+
	Remainder				+				+		
Crossover	Scattered			+	+			+	+		
	Constraint dependent	+	+			+	+			+	+
Mutation	Constraint dependent			+	+	+			+		
	Adaptive feasible	+	+				+	+		+	+
Function constraint		1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-8	1E-8
Tolerance constraint		1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-6	1E-7	1E-7
Relative error [%] of:	R_1	0,06	0,06	0,24	-0,18	0,81	0,14	0,22	-0,06	0,05	0,12
	R_2	0,34	-0,34	-0,16	-0,67	-0,01	-0,14	-0,04	-0,01	0,05	0,03
	X_1	-0,01	0,09	0,04	0,06	0,09	-0,02	0,05	-0,05	0,01	-0,004
	X_2	-0,02	0,09	0,04	0,06	0,08	-0,02	0,04	-0,05	0,01	0,04
	X_m	-0,02	0,01	0,07	0,03	-0,10	-0,001	-0,03	0,01	-0,003	-0,001
CPU time [s]		201	369	234	236	337	215	202	250	281	299

The stochastic remainder selection in GA8 results in moderate speed, good accuracy and fast convergence.

As evident from GA9 and GA10, constraint dependent crossover achieves outstanding results in linear constraint problems. Both algorithms exhibit fast convergence and very good accuracy.

The combination of scattered crossover and stochastic remainder selection in GA4 slows down the convergence rate (Figure 1), though the CPU time is hardly affected and the accuracy remains good (Table 2). This is due to the deterministic nature of the two approaches.

When combined with roulette-wheel selection, scattered crossover in GA3 and GA7 is very fast and produces low errors (Table 2). Since scattered crossover generates offspring solutions as a mixture of the genes of parent solutions, the stochastic noise effect of roulette-wheel selection is eliminated.

GA9 and GA10 use constraint dependent mutation combined with function and tolerance constraint. Since double precision of data is default in MATLAB, the allocation of constraints is in the range of [1E-7, 1E-8]. The constraints added to GA9 and GA10 result in high rates of convergence (figure 1) and minimal errors with an insignificant increase in the CPU time (Table 2).

5. Conclusion

The paper analyzes the choice of fitness functions and parameter settings for enhancing the accuracy and speed of Genetic Algorithms. To validate the analysis, the equivalent circuit parameters of two induction motors are evaluated.

Two new fitness functions (FF1 and FF2) are developed and compared. Ten combinatorial optimization experiments are conducted including various genetic operator types and constraints. To assess the ability of Genetic Algorithms to provide globally optimal solutions, the coefficient of variation is introduced for every combination.

It is found out that the inclusion of auxiliary data modules in FF2 balances out the selection-produced stochastic noise. The impact of the population size on the convergence is analyzed. It is shown that a mean value of population size 550 allows converging to an optimal solution without excessive computation time.

Four types of selection are compared: tournament, roulette-wheel, stochastic uniform sampling and stochastic remainder. The accuracy assessment indicates that the stochastic uniform sampling produces considerably lower errors than the other selection types.

Two types of crossover are implemented: scattered and constraint dependent. The constraint dependent crossover achieves outstanding results. When combined with roulette-wheel selection, scattered crossover is fast, produces low errors and eliminates the stochastic noise.

Two mutation operators are implemented: adaptive feasible and constraint dependent mutation. The constraint dependent mutation results in high rates of convergence and minimal errors with only a slight increase in the computation time.

To obtain relative error within 1%, the conventional fitness function and parameter settings require at least three data sets, i.e. three points from the motors load curves. With two data sets the maximum relative error is about 3%. In contrast, the proposed new fitness functions and GA parameter settings enable to obtain relative error less than 1% with only two data sets.

Based on the results it can be concluded, that adequate fitness functions, combined with proper genetic operators setups, can significantly enhance the accuracy and speed of Genetic Algorithms. In addition, the number of required input data sets can be reduced.

Acknowledgements

The authors would like to thank the Research and Development Sector at the Technical University of Sofia for the financial support.

References

- [1] Wagner S et al. 2018 *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications* (Chapman and Hall / CRC)
- [2] Haupt R and Haupt S 2004 *Practical genetic algorithms* (Wiley & Sons)
- [3] Stoykova S and Spasov V 2019 Determining the parameters of induction motors by Genetic Algorithms 8th International Scientific Conference TECHSYS 2019 (Technical University – Sofia, Plovdiv branch) (to be published)
- [4] Ozyurt C and Ertan B 2005 Prediction of induction motor parameters from manufacturer's data XIV-th International Symposium on Electrical Apparatus and Technologies SIELA 2005 (Plovdiv, Bulgaria) vol II pp 96-105
- [5] Spasov V, Rangelova V, Kostov I and Drambalov V 2014 An efficient approach for determining induction motors parameters *Acta Technica Corviniensis - Bulletin of Engineering* Tome VII pp 117-122
- [6] Kostov I and Rangelova V 2010 Power factor determination of induction motor frequency controlled drives *International Journal of Engineering* (Faculty of Engineering Hunedoara) Tome VIII issue 2 pp 67-72
- [7] Lobo F et al. 2007 *Parameter setting in evolutionary algorithms* (Springer-Verlag)
- [8] Kramer O 2017 *Studies in computational intelligence* vol 679: *Genetic algorithm essentials* (Springer International Publishing)
- [9] Petrowski A and Ben-Hamida S 2017 *Computer Engineering Series: Evolutionary algorithms* vol 9 (Wiley-ISTE)
- [10] Merjalili S 2018 *Evolutionary algorithms and Neural Networks: Theory and Applications*
- [11] Iba H 2018 *Evolutionary approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks* (Springer Singapore)
- [12] Mikhailov K and Sclocco A 2018 The Apertif monitor for bursts encountered in real-time auto-tuning optimization with genetic algorithms *Astronomy and Computing* vol 25 pp 139-148