DOUBLE FOUR-BAR CRANK-SLIDER MECHANISM DYNAMIC BALANCING BY META-HEURISTIC ALGORITHMS

Habib Emdadi¹, Mahsa Yazdanian², Mir Mohammad Ettefagh³ and Mohammad-Reza Feizi-Derakhshi⁴

¹,⁴Department of Computer Engineering, University of Tabriz, Tabriz, Iran
²,³Department of Mechanical Engineering, University of Tabriz, Tabriz, Iran

ABSTRACT

In this paper, a new method for dynamic balancing of double four-bar crank slider mechanism by meta-heuristic-based optimization algorithms is proposed. For this purpose, a proper objective function which is necessary for balancing of this mechanism and corresponding constraints has been obtained by dynamic modeling of the mechanism. Then PSO, ABC, BGA and HGAPSO algorithms have been applied for minimizing the defined cost function in optimization step. The optimization results have been studied completely by extracting the cost function, fitness, convergence speed and runtime values of applied algorithms. It has been shown that PSO and ABC are more efficient than BGA and HGAPSO in terms of convergence speed and result quality. Also, a laboratory scale experimental double-four-bar crank-slider mechanism was provided for validating the proposed balancing method practically.

KEYWORDS

Mechanism; balancing; BGA; PSO algorithm; ABC algorithm; HGAPSO algorithm

1. INTRODUCTION

Dynamic balancing is one of the important concerns in high speed mechanisms such as crank slider mechanism, a special attention should paid to the inertia-induced force (shaking force) and moment (shaking moment) transmitted to the frame in order to reduce vibration amplitude. The methods of balancing linkages are well developed and documented in [1]. These techniques mostly are based on mass redistribution, addition of counterweights to the moving links, and attachment of rotating disks or duplication of the linkages [2]. In these methods, the shaking forces and shaking moments should be minimized. One of these methods is “Maximum recursive dynamic algorithm” that published by Chaudhary and Saha [3]. Another method which is documented by QI and Pennestrlis [4] is called “refined algorithm”. It presents a numerically efficient technique for the optimum balancing of linkages. In this approach, instead of solving directly the dynamic equations, a technique is introduced to solve the linked dynamic equations in a “shoe string” fashion. The comprehensive mass distribution method, for an optimum balancing of the shaking force and shaking moment is used by Yu [5] to optimal balancing of the spatial RSSR mechanism.

In recent years, optimization techniques have received widespread interest in the engineering sciences. These techniques were initiated, primarily, by the advent of high speed digital computers which give a practical, economical and accurate means of obtaining solutions to
meaningful problems in the optimization of static and dynamic systems. Dynamic balancing inherently constitutes an optimization problem. For instance, CWB (counterweight balancing) involves a trade-off between minimizing the different dynamic re-actions. Therefore, determining the counterweights’ mass parameters is an optimization problem. Also, Alici and Shirinzadeh [6] considered “Sensitivity analysis”. In this technique they formulated the dynamic balancing as an optimization problem such that while the shaking force balancing is accomplished through analytically obtained balancing constraints, an objective function based on the sensitivity analysis of shaking moment with respect to the position, velocity and acceleration of the links is used to minimize the shaking moment. Evolutionary algorithms such as GA and PSO are useful in optimization problems, because they are easy to implement for the complex real-world problems. In some cases that objective function could not expressed as an explicit function of design variables, using these evolutionary algorithms is more suitable for optimization than the traditional deterministic optimization methods. Wen-Yi Lin proposed a GA-DE hybrid algorithm for application to path synthesis of a four-bar linkage [7]. Acharyya and Mandal[8]applied BGA with multipoint crossover, the PSO with the CFA (constriction factor approach) and the DE to the path synthesis of a four-bar linkage. Jaamilahmadi and Farmani [9] proposed another related work. They described the application of GA to force and moment-balance of a four bar linkage.

The aim of present work is applying new artificial intelligent based optimization methods for balancing double four-bar crank slider mechanism which is a special case of four-bar mechanism, but it has its own specification in dynamic balancing. Ettefagh and Abbasidoust [10] applied BGA for balancing of a crank slider mechanism, however in present work a comprehensive study of different algorithms proposed for balancing of a crank slider mechanism was carried out. The major objective of this paper is to compare the computational effectiveness and efficiency of BGA (Binary Genetic Algorithm), PSO (Particle Swarm Optimization), ABC (Artificial Bee Colony) and HGAPSO (Hybrid GA and PSO) algorithms. The performance of the optimization techniques in terms of computational time and convergence rate is compared in the results section. After kinematics formulating of the mechanism and extracting the object function with its’ constraints, results of the proposed method on a simulated mechanism will be reported. Finally, for experimentally validation of the method, a laboratory scale experimental crank slider mechanism will be considered and the proposed optimization method will be applied for its balancing.

2. MODELLING

The selected mechanism in this paper is shown in Figure 1, which is a benchmark double four-bar mechanism and this mechanism is applied in different machines [10]. Therefore the proposed balancing method in this paper may be easily applied to other mechanisms in different machines for balancing. As shown in Figure 1, the mechanism contains two cranks-sliders and four rotating disks. Each rode is attached to disk 1 and disk 4. All disks are connected to a shaft which is rotating with an electromotor. When the mechanism is working, shaking force and moments cause frame to vibrate. Our goal is to minimize shaking forces and moments according to the regarded constraints. The mechanism is modeled as shown in Figure 2 by computing shaking forces of the mechanism on all four planes.
In this paper, the method used for balancing is the addition of two counterweights on disks 2 and 3. In addition, system has an unbalance mass on disk 2. Shaking forces in crank slider as the dynamical forces of mechanism are formulated as:

\[ f_c = m_c R \omega^2 \]

\[ f_p = m_p R \omega^2 \left( \cos \theta + \frac{R}{L} \cos 2\theta \right) \]

Where \( m_c \) is the eccentric mass and \( m_p \) is the equal mass of the slider, \( R \) is corresponding radius and \( \omega \) is mechanism rotating speed. Also, \( \theta \) indicates the angular position of the disk and \( L \) is the connecting rod’s length. Equilibrium on \( x \) and \( y \) axes due to the acting forces are described by:
\[
\sum F_x = 0 \to m_2 \omega^2 R_x \cos(\theta + \alpha) + m_2 \lambda_2 \omega^2 \cos(\theta + \varphi_2) + m_2 r_2 \omega^2 \cos(\theta + \varphi_2) + m_1 R \omega^2 \cos(\theta + \alpha) + m_1 \lambda_1 \omega^2 \cos(\theta + \varphi_1) + m_1 r_1 \omega^2 \cos(\theta + \varphi_1) = 0
\]

Also Momentum equilibrium equations are written as:

\[
\sum M_x = 0 \to m_2 \omega^2 R_x \sin(\theta + \alpha) + m_2 \lambda_2 \omega^2 \sin(\theta + \varphi_2) + m_2 r_2 \omega^2 \sin(\theta + \varphi_2) = 0
\]

\[
\sum M_y = 0 \to m_2 \omega^2 R_x \sin(\theta + \alpha) + m_2 \lambda_2 \omega^2 \sin(\theta + \varphi_2) = 0
\]

In construction of optimization problems, as described before, some parameters of system is applies as the variables of cost function to minimize it. In our method, \( m_1, m_2, \varphi_1, \varphi_2, r_1 \) and \( r_2 \) are counterweights parameters that are unknown. We take \( r_1 \) and \( r_2 \) as known parameters. So we define the cost function as below. The cost function is consisting of forces equations and the constraints are considered by moment equations. For this purpose, we define some parameters as \( P_1, P_2, P_3 \) and \( P_4 \) as below:

\[
P_1(\theta) = \sum F_x, \quad P_2(\theta) = \sum F_y
\]

\[
P_3(\theta) = \sum M_x, \quad P_4(\theta) = \sum M_y
\]

These parameters are the sum of forces and moments in x, y direction. Consequently, the cost function is:

Cost function: \( f(m_1, m_2, \varphi_1, \varphi_2) = \int \left| P_1(\theta) \right| + \left| P_2(\theta) \right| dA \)

The cost function is a function of the parameters \( m_1, m_2, \varphi_1 \) and \( \varphi_2 \), which should be found. In other words, the cost function is the area that \( \left| P_1(\theta) \right| + \left| P_2(\theta) \right| \) makes in polar system. Our goal is to minimize this area. Additionally, by defining C1 and C2 as follows:

\[
C_1(m_1, m_2, \varphi_1, \varphi_2) = \int \left| P_3(\theta) \right| dA
\]

\[
C_2(m_1, m_2, \varphi_1, \varphi_2) = \int \left| P_4(\theta) \right| dA
\]
Constraints are considered such that $C_1$ and $C_2$ to be less than predefined values which determined by searching among the maximum value of them during running the Meta-Heuristic Algorithms.

3. APPLYING META-HEURISTIC ALGORITHMS

PSO, ABC and HGAPSO algorithms, which is introduced in summary in following section, are applied for minimizing the mentioned cost function with defined constraints as it was described in previous section.

3.1. Particle Swarm Optimization algorithm

The PSO method introduced for the first time in 1995 by Kennedy and Eberhart [11]. This method based on social behavior of organisms such as fish schooling and bird flocking. Because of its simple concept and quick convergence, PSO can be applied to various applications in different fields. The approach uses the concept of population and a measure of performance like the fitness value used with other evolutionary algorithms. Similar to other evolutionary algorithms, the system is initialized with a population of random solutions, called particles.

Each particle maintains its own current position, its present velocity and its personal best position explored so far. The swarm is also aware of the global best position achieved by all its members. The iterative appliance of update rules leads to stochastic manipulation of velocities and flying courses. During the process of optimization the particles explore the $D$-dimensional space, whereas their trajectories can probably depend both on their personal experiences, on those of their neighbors and the whole swarm, respectively. In every iteration, each particle is updated by following the best solution of current particle achieved so far ($P_{best}$) and the best of the population ($gbest$). When a particle takes part of the population as its topological neighbours, the best value is a local best. The particles tend to move to good areas in the search space by the information spreading to the swarm. The particle is moved to a new position calculated by the velocity updated at each time step $t$. This new position is calculated as the sum of the previous position and the new velocity by equation (12).

$$v_{i}^{t+1} = wv_{i}^{t} + c_1 r_{1} \text{rand}(0,1)(P_{best_i} - x_{i}^{t}) + c_2 r_{2} \text{rand}(0,1)(gbest_t - x_{i}^{t})$$

(12)

The velocity updated as following equation:

$$x_{i}^{t+1} = x_{i}^{t} + v_{i}^{t+1}$$

(13)

The parameter $w$ is called the inertia weight and controls the magnitude of the old velocity $v_{i}^{t}$ in the calculation of the new velocity which is $w_{i}^{t+1}$, whereas the acceleration factors $C_1$ and $C_2$ determine the significance of $P_{best}$ and $gbest$ respectively. In the present work we selected $C_1=0.25$ and $C_2 =0.15$. Furthermore, $x_{i}^{t}$ is the current particle position and $x_{i}^{t+1}$ is the new position of the particle. $rand(0,1)$ is the random number uniformly distributed. In general, the inertia weight $w$ is set according to the following equation:

$$w = w_{max} - \frac{w_{max} - w_{min}}{I_{t_{max}}} \times Iter$$

(14)

Where $w_{max}$ is initial weight, $w_{min}$ is final weight, $I_{t_{max}}$ is the maximum number of iteration (generation) and $Iter$ is current iteration number.

Main steps of the procedure are:

1. Position and initial velocity of particles are generated randomly.
2. The objective function value is calculated for each particle.
3. \( P_{best} \) For each particle is considered equal to its initial position (in the first iteration). Also \( g_{best} \) determined.
4. Modify position and velocity of each particle using the equations (12) and (13).
5. Calculate the overall objective function for each particle.
6. If the value of new overall objective function is better than the value of each particle in \( P_{best} \), then it would be replaced in \( P_{best} \) and if in whole new population gets found which is better fitting in objective function would be replaced by \( g_{best} \).
7. Repeat steps 4 to 7 to reach the maximum number specified. The final answer is obtained from the \( g_{best} \) in the last iteration.

Figure 3 shows the flowchart of the developed PSO algorithm [11].

![Flowchart of the PSO algorithm](image)

Figure 3. Flowchart of the PSO algorithm

### 3.2. Artificial Bee Colony algorithm

The artificial bee colony (ABC) algorithm is a new population-based stochastic heuristic approach which was introduced for the first time by Karaboga in 2005 [12]. This algorithm is very simple and flexible, which does not require external parameters like crossover rates, especially suitable for engineering application. The colony of artificial bees consists of three groups of bees to search foods generally, which includes employed bees, onlookers and scouts. The first half of colony consists of the employed artificial bees and the second half includes the onlookers. The employed bees search the food around the food source in their memory. They perform waggle dance upon returning to the hive to pass their food information to the other of the colony (the onlookers). The onlooker bees are waiting around the dance floor to choose any of the employed bees to follow based on the nectar amount information shared by the employed bees. The employed bee whose food source has been exhausted by the bees becomes a scout who must find new food source. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The onlooker bee chooses probabilistically a food source depending on the amount of nectar shown by each employed bee, see equation (15).

\[
P_i = \frac{f_{it_i}}{\sum_{i=1}^{N} f_{it_i}} \tag{15}
\]
Where $fit_i$ is the fitness value of the solution $i$ and $SN$ is the number of food sources which are equal to the number of employed bees.

Each bee searches for new neighbor food source near of their hive. After that, employed bee compares the food source against the old one using equation 16. Then, it saves in their memory the best food source [13-14].

$$v^{new}_{ij} = x^{old}_{ij} + \varphi_{ij} (x^{old}_{ij} - x^{old}_{kj})$$  \hspace{1cm} (16)

Where $k \in \{1,2,3,\ldots, NP\}$ and $j \in \{1,2,3,\ldots, n\}$ are chosen randomly. $\varphi_{ij}$ is a random number between $[-1, 1]$. In ABC, Providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predefined number of cycles is an important control parameter of the ABC algorithm, which is called “limit” for abandonment. Assume that the abandoned source is $x_i$ and $j=\{1, 2\ldots N\}$ then the scout discovers a new food source to be replaced with $x_i$. This operation can be defined using equation 17.

$$x^1_i = x^\text{min}_i + \text{rand}(0,1)(x^\text{max}_i - x^\text{min}_i)$$ \hspace{1cm} (17)

ABC implementation stages of the algorithm:

1. Initialization
2. Place employed bees on the food sources
3. Place onlookers on the food sources
4. Send scout to carry out the global search

Flowchart of the ABC algorithm is shown in figure 4 [12].

![Flowchart of the ABC algorithm](image-url)
3.3. Hybrid of Genetic Algorithm and Particle Swarm Optimization (HGAPSO)

Concept of GA:

Genetic algorithm was introduced by John Holland in 1967. Later, in 1989, by Goldberg’s attempts got accepted. And today, due to its capability, it has prosperous place among other methods. Routine optimization in genetic algorithm bases on random-directed procedure. In this way depending on Darwin’s theory of evolution and fundamental ideas the method has been firmed. First of all, for some constants which are called population, a swarm of main parameters, eventually get produced. After performing scalar simulator program, a number that is attributed as standard deviation or set fitness data, impute as a member of that population. Fitness process is done for all created members then by summoning genetic algorithm operators such as succession (crossover), mutation and next generation selection (selection) are formed. This routine continues till satisfying convergence [15].

Concepts of PSO:

Initially, velocity and position of each particle randomly select and direct all particles gradually to the best answer. PSO has high speed in local and total search. Applying PSO takes to dominate rapid and untimely convergence of GA that is achieved with elitism operations. Also, PSO can fit particles of each generation whereas GA is seeking for an answer generation to generation. Depending on these points, GA and PSO are good supplements of each other [16].

The HGAPSO algorithm is following as bellow:
1. A population randomly should be selected, initially, which is parameters of GA and particles of PSO.
2. The fitness of concluding is calculating.
3. Half of population with high fitness would be selected and then PSO function would be performed. But a factor breeding ratio was presented with $\phi$ symbol [17]. The factor determines the percentage of population on which the PSO operation is done.
4. Enhanced elites would be transmitted to the next generation directly and mutation and remixing functions would be done on the rest population.

Figure 5 shows all the above steps and flowchart of the HGAPSO algorithm is shown in Figure 6 [16].

![Figure 5. Steps of HGAPSO algorithm [16]](image)
4. RESULTS AND DISCUSSION

In this section the results of mentioned algorithms implementation in our optimization problem for minimizing cost function will be presented and discussed. Table 1 expresses the results of dynamic balancing optimization with PSO, ABC, BGA and HGAPSO algorithms for 200 and 300 iterations. Also programming is done with MATLAB12, with 2.8 GHZ computer system processor and 1024MB memory. By these tables, best and worst average results are shown in bold. Results comparison indicates that PSO algorithm is more efficient than other algorithms, as the average cost of objective function for 20 iterations is 29853.35062, and for 300 iterations is 29853.30955 that is less than three other algorithms.
Also, above mentioned results in the table are illustrated in Figures 7 and 8 in bar chart which show the comparison of objective function cost. By observing these figures, it may be concluded that the cost order is as following: PSO<ABC<BGA<HGAPSO.
In addition, the runtime comparison shows that PSO algorithm finds the optimal solution in shorter time than the others. Also the average runtime for this algorithm for 200 iterations is 13.2666839 seconds and for 300 iterations is 21.4484653 seconds. According to numerical results PSO algorithm is 3.84 times faster than BGA and 2.56 times faster than HGAPSO and 2.21 times faster than ABC. Figures 9 and 10 indicate the runtime of these algorithms for one performance with 200 and 300 iterations, respectively. By observing these figures, it maybe found that the runtime order is as following: PSO<ABC<HGAPSO<BGA. It should be mentioned that the runtime for Hybrid algorithm of BGA and PSO algorithm is 1.5 times faster than BGA algorithm in reaching optimal solution.
Figure 9. Comparing runtime for 200 iterations

| Algorithm | Average CPU Time | Best CPU Time | Worst CPU Time |
|-----------|------------------|---------------|----------------|
| PSO       | 13.2666839       | 12.264391     | 14.755708      |
| ABC       | 26.6315405       | 24.497529     | 28.761479      |
| HGAPSO    | 34.2630126       | 32.479893     | 35.567686      |
| BGA       | 29.82878         | 27.81643      | 32.98539       |

Figure 10. Comparing runtime for 300 iterations

| Algorithm | Average CPU Time | Best CPU Time | Worst CPU Time |
|-----------|------------------|---------------|----------------|
| PSO       | 21.4484653       | 20.567431     | 22.241028      |
| ABC       | 45.1617406       | 40.422035     | 49.274214      |
| HGAPSO    | 55.5041648       | 51.955371     | 56.950295      |
| BGA       | 71.41247         | 61.1857       | 85.51516       |

Figure 11 indicates the time order of these algorithms for one performance with 100 and 200 repetitions. In this chart, the implementation of time progress of these algorithms is in term of seconds.
Figure 11. Computation time variation versus generation for 100 iterations (Left) and 200 iterations (right)

Figure 12 shows fitness functions of four algorithms in 100 runs. In this figure the range of fluctuation of fitness value for BGA and HGAPSO algorithms is between 29853 and 29855, while this range for BGA and HGAPSO is approximately between 29853 and 30200. Also high difference between the best and worst result of the BGA and HGAPSO algorithms indicates the weakness and dependence of these two algorithms and their sensitivity to the initial population, while PSO and ABC solution range is around the mean values, so it indicates the stability of them. Also, the PSO and ABC algorithms are close to average result that this matter indicates the stability of these two algorithms.

Figure 12. Fitness functions fluctuation for 100 runs for four algorithms

Figure 13 and 14 display the four algorithms convergence in a sample run of them (for 200 and 300 iterations). According to the minimization target, an important point which should be considered is that the charts are descending. In addition the scale of $10^4$ is using for measuring of objective function. As comparing of these two figures, it may be observed that the PSO and ABC algorithms get convergence approximately after 30 generations also they achieve the optimized solution in high speed rather than HGAPSO and GA.
Finally, Figure 15 indicates cost function diagram in balanced and unbalanced states by mentioned algorithms in polar plot. This type of plot is a standard plot which applied for illustrating the ability of the proposed method in balancing of the mechanism in one revolution of the mechanism driver (0-360°). As it was described in previous sections, the main purpose of optimization is minimizing of the cost function in equation 9 which maybe interpreted as minimizing the limited area in the mentioned plot. Therefore better balancing method is a method which has minimum and uniform area in the polar plot. By observing this figure, it maybe concluded that PSO has better performance than other algorithm, because it contains minimum area in comparing with other algorithms.
5. EXPERIMENTALLY VALIDATION OF THE PROPOSED METHOD

For showing the practicability of the proposed balancing method based on mentioned optimization algorithms, a laboratory scale experimental double four-bar crank-slider mechanism was provided as shown in Figure 16 [10]. By using two accelerometers (type 4507 B&K Co.) and data acquisition system (PULSE B&K Co.), the acceleration of the right and left bearing of the mechanism can be extracted. Also a tachometer (Type 0024 MM B&K Co.) is applied for identifying the one revolution of the crank for estimating the angular acceleration. As a sample, the acceleration signals of the unbalanced and balanced mechanism, applying BGA, are illustrated in Figures 17 and 18. By comparing the mentioned linear and angular acceleration signal of the unbalanced and balanced mechanism, the possibility and accuracy of the proposed method in balancing of real mechanisms can be observed. In addition the corresponding angular acceleration signals are illustrated in Figure 19. As it is obvious from these figures, the proposed method is able to balance the mechanism because the overall amplitude of the linear and angular acceleration was decreased after balancing. Other similar results may be obtained by applying other algorithms, which cannot be illustrated in this paper because of the page limitation. By considering the experimental result, it can be observed that the same conclusion maybe derived as in previous section while comparing these algorithms.
Figure 16. A laboratory scale experimental double four-bar crank-slider mechanism setup

Figure 17. Acceleration signal of the right part of the mechanism for before and after balancing

Figure 18. Acceleration signal of the left part of the mechanism for before and after balancing
6. CONCLUSION

In this paper, four Meta-Heuristic algorithms, that are PSO, ABC, BGA and HGAPSO, were applied for balancing of double four-bar crank slider mechanism. This mechanism is a basic and benchmark mechanism which is applied in different machine and engines. Therefore proposed balancing method may be applied for different type of machines for reducing the vibration and noise because of unbalancing. Kinematic modelling of the mechanism and suitable objective function which should be minimized and the constraints are derived. In order to comparing the optimization algorithm, different results based on cost function, fitness, convergence speed and runtime values were extracted and studied. The results show that the PSO algorithm converges quickly to the optimum solution, while the optimality of PSO solutions has a higher quality than other algorithms. Also, reviewing the responses of the four algorithms in several performances with various primary populations determined that, PSO and ABC are more stable than BGA and HGAPSO. Therefore PSO and ABC have a good potential in finding optimal solution and its convergence characteristics are more favourable. In addition, the practicability of the method was proved by applying the proposed method on a laboratory scale experimental complex crank slider mechanism.

REFERENCES

[1] A.A. Sherwood & B.A. Hockey, (1968)”The optimization of mass distribution in mechanisms using dynamically similar systems”, Journal of Mechanisms, Vol. 4, pp. 243–260.
[2] G.G. Lowen & R.S. Berkof, (1971) ”Determination of forced-balance four-bar linkages with optimum shaking moment characteristics”, ASME Journal of Engineering for Industry, Vol. 93, No. 1, pp. 39–46.
[3] H. Chaudhary & S. K. Saha, (2007) “Balancing of four-bar linkages using maximum Recursive dynamic algorithm”, Mechanism and Machine Theory, Vol. 42, pp. 216–232.
[4] N. M. Qi & E. Pennestrl, (1991) “Optimum balancing of four-bar linkage, a Refined algorithm,” Mech. Mach. Theory, Vol. 26, No. 3, pp. 337–348.
[5] Y. YU, (1987) “Optimum shaking force and shaking moment balancing of the RSS'R spatial linkage,” Mech. Mac& Theory, Vol. 22, No. 1, pp. 39–45.

[6] G. Alici & B. Shirinzadeh, (2006) “Optimum dynamic balancing of planar parallel manipulators based on sensitivity analysis,” Mechanism and Machine Theory, Vol. 41, pp. 1520–1532.

[7] Wen-Yi Lin, (2010) “A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage,” Mechanism and Machine Theory, Vol.45, No.8, pp. 1096-1107

[8] S. K. Acharyya & M. Mandal, (2009) “Performance of EAs for four-bar linkage synthesis ,” Mechanism and Machine Theory, Vol.44, pp.1784-1794.

[9] A. JaamioIahmadi & M. R. Farmani, (2006) “Optimal force and moment balance of a four bar linkage via genetic algorithm,” 14th Annual (International) Mechanical Engineering Conference-Esfahan - Iran.

[10] M. M. Ettefagh, F.abbasidust, H.milanchian & M.yazdanian (2011) “Complex Crank-Slider Mechanism Dynamic Balancing by Binary Genetic Algorithm (BGA)”, symposium in Innovations in Intelligent Systems and Applications (INISTA2011) - Istanbul, pp.277 – 281.

[11] J. Kennedy & R. Eberhart, (1995). “Particle Swarm Optimization,” Proceeding of IEEE International Conference on Neural Networks in Perth-WA, Vol.4, pp. 1942-1948.

[12] D. Karaboga & B. Basturk, (2008) “On the performance of artificial bee colony (ABC) algorithm,” Applied Soft Computing, Vol. 8, pp. 687–697.

[13] S.H. Sishaj & P. Simon, (2010) “Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions,” Electric Power Components and Systems, Vol. 38, No. 7, pp. 786-803.

[14] F. Kang, J. Li , H. Li, Z. Ma & Q. Xu, (2010) “An improved artificial bee colony algorithm”, Intelligent Systems and Applications (ISA), pp.1-4.

[15] Z. Michalewicz, (1999) “Genetic Algorithms+ Data Structures=Evolution Programs”, Springer-Verlag Berlin Heidelberg New York.

[16] Chia-Feng Juang, (2004) “A Hybrid of Genetic Algorithm and Particle swarm Optimization for Recurrent Network Design”, IEEE Transactions on system, Man and Cybernetics, Vol. 34,No. 2, pp.997-1006.

[17] M.Settles, P. athan & T. Soule, (2005)” Breeding Swarms: A New Approach to Recurrent Neural Network Training”, Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO 2005), pp. 185-192.

Authors

Mir Mohammad Ettefagh received the PhD degree in Mechanical Engineering from the University of Tabriz in 2007. He is currently an Assistant professor in Mechanical Engineering department of the University of Tabriz. His research interests include vibration and dynamic systems, optimization of mechanism, fault diagnosis of machines, structural health monitoring.

Mohammad-Reza Feizi-Derakhshi, Ph.D. is currently a faculty member at the University of Tabriz. His research interests include: natural language processing, optimization algorithms, intelligent methods for fault detection and intelligent databases.

Habib Emdadi received his BS degree in Software Engineering Technology from University of Tabriz, Iran in 2009. He is currently working toward the MS degree of Computer Science with Intelligent Systems in Tabriz University, focusing on Optimization Techniques. His research interests include Evolutionary Computation, Optimization Techniques, soft computing and Computer Programming.

Mahsa Yazdanian received the BS degree in Mechanical Engineering from University of Tabriz, Iran in 2010. She is currently working toward the MS Degree in the Department of Mechanical Engineering. Her research interests include artificial-based optimization of mechanism and condition monitoring of rotating machinery.