FITTING OF BEZIER CURVES USING THE FIREWORKS ALGORITHM

K. Srinivas Reddy[1], Amitava Mandal[2], Krishna Kumar Verma[3] and G. Rajamohan[4]
[1]PG Student, [2]Professor, [3]Associate Professor and [4]Assistant Professor
Department of Manufacturing Engineering
National Institute of Foundry and Forge Technology, Ranchi, India

ABSTRACT

A swarm intelligence based method is presented in this paper to solve the problem of parameterization of given set of planar and spatial data points, which may be uniformly or non-uniformly spaced, for least squares fitting with Bezier curves. The basic framework of the proposed method relies on fireworks algorithm. Performance of the proposed method is tested through several example curves of varying complexities.

KEYWORDS: *Bezier curves, Computer-aided design, Fireworks algorithm, Parameterization, Least squares method, Swarm intelligence.*

I. INTRODUCTION

Fitting of parametric curves and surfaces to measured or generated point clouds finds applications in many industrial and technological domains, such as computer-aided design and manufacturing (CAD/ CAM), medical imaging, reverse engineering, virtual reality, etc. While the point clouds are in three-dimensional *Cartesian space*, the curves and surfaces are created in *parametric space*. The process of assigning unique parameter values for each point in the point cloud is known as *parameterization*. The accuracy of fitted curves and surfaces depend on proper allocation of parameter values. Among the available parameterization methods, uniform, chord length and centripetal methods are commonly used. Fitting of parametric Bezier curves for the planar and spatial points is considered in the present work.

Bezier curve fitting is generally based on *least squares method (LSM)*. Curve fitting starts with the estimation of parameters and knot vectors followed by the determination of its control points by minimizing the sum of squares of distances between data points and the fitted curve. An attempt to fit B-spline curves for randomly measured points using the least squares method has been reported [1]. It may be noted that the number of unknown variables to be solved increases with large numbers of data points. In such cases, use of conventional optimization methods for solving the system of equations may not be feasible or may result in inferior curves. This has led researchers to explore the possibility of solving curve fitting problems using *neural networks* [2], *evolutionary algorithms*, etc. Some of the research efforts related to evolutionary algorithms are presented below.

Genetic algorithm (GA) is a commonly used evolutionary algorithm for solving complex optimization problems. Kumar, et al. proposed a parameterization method for fitting non-uniform B-spline curves using genetic algorithms and *radial basis function neural networks* [3]. Galvez, et al. developed two *artificial intelligence* techniques for the least-squares approximation of Bezier curves and surfaces, viz. a genetic algorithm based approach for parameterization and a functional networks scheme for handling additional functional constraints used [4]. Sun, et al. proposed an adaptive genetic algorithm to optimize the parameters of B-spline curves, as the randomly initialized regular genetic algorithm typically lead to local minima and require more control points to assure higher fitting accuracy [5].

Zhao, et al. applied genetic algorithms to optimize the parameters of Bernstein basis functions for least squares fitting of Bezier curves to the planar data points [6]. Hasegawa, et al. proposed a multi-objective evolutionary algorithm, known as *generalized differential evolution algorithm 3* (GDE3), for parameter optimization in fitting of Bezier curves. The authors have used two objective functions, viz. error of fit and length of the curve, to obtain aesthetically pleasing curves while achieving a good trade-off between the error and length [7]. Galvez and Iglesias presented an *iterative mutually coupled hybrid evolutionary approach* for fitting of B-spline curves. Their approach used genetic algorithm for data parameterization and the particle swarm optimization (PSO) for knot placement [8].
Loucera, et al. attempted to solve the problem of approximating given set of noisy and noise-free data points with Bezier curves through a hybrid strategy that combines *linear least-squares minimization*, *simulated annealing* and *information science metrics*, namely *data-fidelity* and *model-complexity* [9]. Galvez, et al. used the *clonal selection theory*, an *artificial immune system (AIS) algorithm*, to solve the problem of obtaining the rational Bezier curve that fits a given set of data points better in the least-squares sense [10]. Iglesias, et al. presented a *hybrid mesh adaptive search algorithm* that uses clonal selection theory as first step, followed by an efficient local search procedure [11]. This approach has been shown to yield better results than just using the clonal selection theory.
Recently, the *swarm intelligence algorithms* are being used to obtain suitable parameters for accurate fitting of curves and surfaces. Swarm intelligence is the study of computational systems inspired by *collective intelligence* [12]. Galvez and Iglesias applied a powerful *metaheuristic algorithm*, called the *firefly algorithm* inspired from the social flashing behaviour of *fireflies* in nature, to obtain an optimal approximating Bezier curve to given set of data points with proper selection of control parameters [13]. The authors also used the firefly algorithm to approximate a given set of noisy data points using explicit B-spline curves [14]. Iglesias and Galvez presented an approach to fit smooth rational Bezier curves for noisy data points using *memetic firefly algorithms* by formulating the fitting problem to be a continuous multivariate nonlinear optimization problem [15]. Iglesias, et al. presented an approach based on the *bat algorithm* for optimal parameterization for least-squares fitting of polynomial Bezier curves [16]. The application of bat algorithm has also been extended to rational Bezier curves [17]. Iglesias and Galvez used the *memetic electromagnetism algorithm* for approximating the data points with rational Bernstein polynomial curves. This algorithm combines a global optimization algorithm, viz. electromagnetism algorithm, with a local search method [18].
Every evolutionary algorithm-based approach involves the tuning of several parameters that form the basic framework in order to achieve good performance of the particular algorithm. The values of these parameters are usually set by user-experience and for a completely new problem they are set by large collection of empirical results. Hence, an algorithm which involves parameters that are easy to set is needed. One such algorithm is the *fireworks algorithm* [19] that involves only four parameters. These parameters are almost same for every parameterization problem. Some of the essential details of this algorithm are given in Section III. In the present work, a fireworks algorithm based parameterization approach is proposed for fitting of polynomial Bezier curves and its performance has been tested with several example cases.
Remainder of the paper is organized as follows. Section II describes about the least squares fitting of Bezier curves and Section III describes about the proposed method. Section IV presents and discusses the results obtained using the proposed method and finally the conclusions are presented in Section V.

II. LEAST SQUARES FITTING OF BEZIER CURVES

The *Bezier curve* can be mathematically defined as [20]:

$$p(u) = \sum_{i=0}^{n} p_i B_{i,n}(u); \quad 0 \le u \le 1 \tag{1}$$

where, the vectors p_i represent the $(n + 1)$ *vertices* of a *characteristic or control polygon* and $B_{i,n}(u)$ is the i^{th} Bernstein basis function of degree n. The vertices are known as *control points* as they control the shape of the Bezier curve. Eq. (1) can be used to compute the points on a Bezier curve. The values of Bernstein basis function can be computed using Eq. (2).

$$B_{i,n}(u) = \binom{n}{i} u^i \ (1-u)^{n-i} \tag{2}$$

Let us consider the reverse engineering scenario, where the *data points* are known and the control points are to be estimated. Assume that there are N data points (D_j) in a ξ-dimensional space (ξ = 2 for *planar curves* and 3 for *space curves*). The control points of the Bezier curve can be estimated using the *least squares* (*LS*) fitting method, i.e. by *minimizing the sum of squared errors* (E) over the set of data points. The LS fitting can be represented as follows:

$$E = \sum_{j=1}^{N} [D_j - p(u)]^2 \tag{3}$$

The system of equations to be solved by LS fitting can be represented using Eq. (4).

$$D_j = \sum_{i=0}^{n} Q_i B_{i,n}(u); \quad 0 \le u \le 1; \quad j = 0,\ldots,(N-1) \tag{4}$$

The above equation can be written in matrix form as shown below.

$$[D] = [B][Q] \tag{5}$$

In Eq. (5), vector [D] contains the data points, vector [Q] contains the control points and matrix [B] contains the Bernstein basis function matrix. The control points [Q] may be obtained using Eq. (6).

$$[Q] = [B]^{-1}[D] \tag{6}$$

III. PROPOSED METHOD

In solving the system of equation in Eq. (4) and Eq. (10), suitable values of u (for curves) and u and v (for surfaces) are to be determined first. Given the nonlinear nature of the problem, the parameters are to be determined through optimization. The *fireworks algorithm* is used for solving this system of equations. The flowchart of the proposed method is shown in Figure 1. Some of the *preparatory steps* needed for the fireworks algorithm are as follows:

1) The parameter values are considered as fireworks and are encoded as real coded vectors of length equal to number of given data points (N), i.e. $u = [u_1, u_2, \ldots, u_N]$.
2) The objective function is the function E given in Eq. (3).
3) The degree of the underlying curve and number of control points are not known apriori and are dependent on the complexity of the curve. The method therefore starts with less number of control points, and increases it until the error reduces below a certain threshold value.
4) Setup the control parameters, namely the *population size* (n_{pop}), the *number of iterations* (n_{ite}), *total number of sparks* (n_{spr}) and the *maximum explosion amplitude* (A).

3.1 Number of sparks

The number of sparks to be generated for each member in the population can be found using Eq. (7).

$$s_i = n_{spr} \frac{E_{\max} - E_i + \zeta}{\sum_{i=1}^{n_{pop}} (E_{\max} - E_i) + \zeta} \tag{7}$$

where, ζ is a small constant introduced to avoid the zero division error.

3.2 Amplitude of explosion

The amplitude of explosion of each member in the population can be estimated using Eq. (8).

$$A_i = A \frac{E_i - E_{\min} + \zeta}{\sum_{i=1}^{n_{pop}} (E_i - E_{\min}) + \zeta} \tag{8}$$

where, ζ is a small constant introduced to avoid the zero division error.

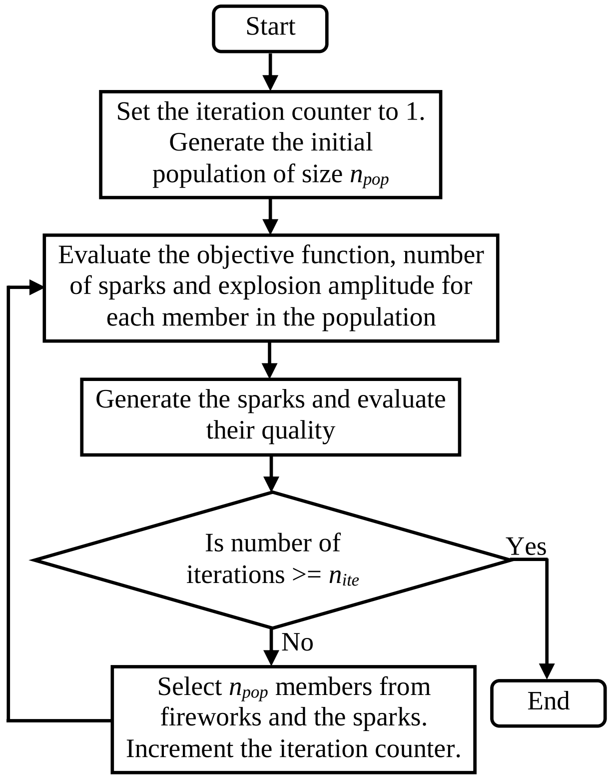


Figure 1. Flowchart of the proposed method

3.3 Generation of sparks

Sparks are generated by mimicking the explosion process. Two kinds of spark generation methods are generally employed considering the convergence and diversity of explosion. One method generates the sparks towards the best among the population, while the other generates sparks in a random way. Both ways of generating sparks are given below in Eq. (9).

$$\begin{aligned} u_i^s &= u_i^f + A_i \; rand(0,1) \; sign(u_{ibest}^f - u_i^f) \\ u_i^s &= u_i^f + (-1)^{randi(2,1,1)} \; A_i \; rand(0,1) \\ u_i &\in u; \; i = 1,2,\ldots,N; \end{aligned} \tag{9}$$

where, the superscripts *s* and *f* stand for sparks and fireworks respectively.

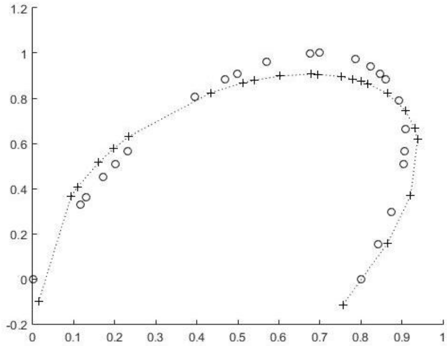
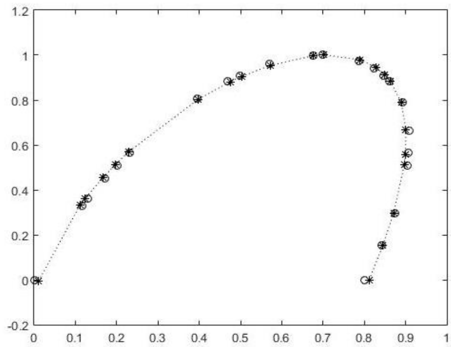
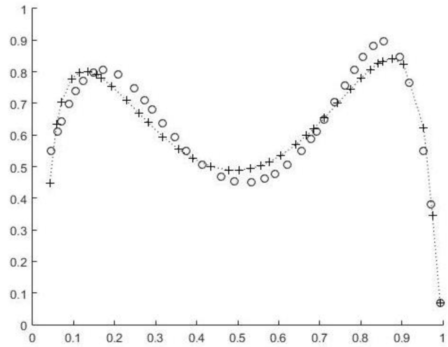
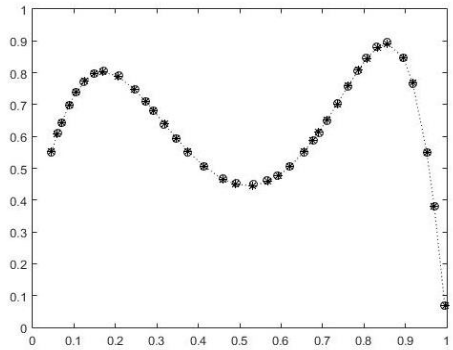
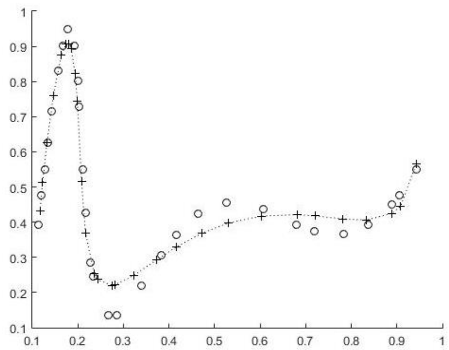
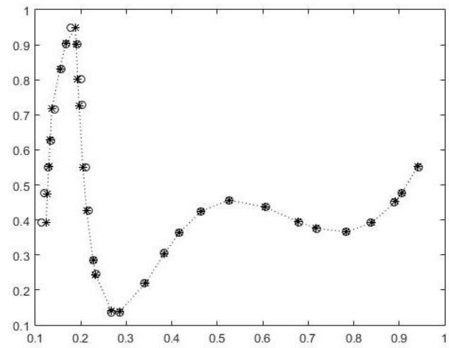
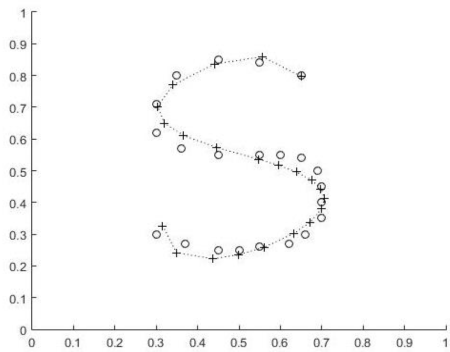
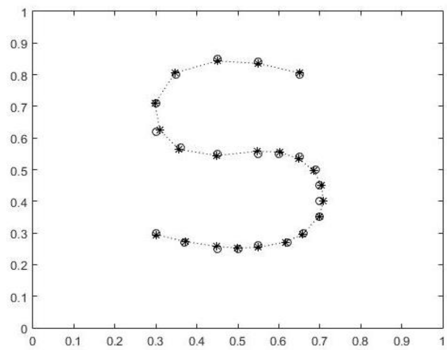
3.4 Selection of sparks or locations

At the end of every iteration, the current best member, for which the objective function is optimal among population is kept for the next explosion generation. After that, (n_{pop} - 1) members are selected based on their distance to other members so as to keep diversity.

IV. RESULTS AND DISCUSSION

The chord length parameterization and proposed fireworks algorithm based parameterization methods have been implemented using MATLAB. The former has been implemented for comparison purposes. Several example Bezier curves of varying complexities have been used to compare these methods. Four examples of planar curves and two examples of space curves approximated by both the methods are shown respectively in Tables 1 and 2. Axes of the plots are marked in normalized coordinates. A visual examination of the fitted curves clearly reveal that the proposed parameterization method yields better looking curves than those obtained using chord length method.

Table 1. Planar curves approximated by chord length and proposed parameterization methods

Example No.	Approximation Method	
	Chord Length Parameterization	**Proposed Parameterization**
P1		
P2		
P3		
P4		

In tables 1 and 2, the points marked with *circle* are the data points and those marked with *plus sign* are points on the approximated Bezier curves. Table 3 shows the errors associated with the fitted curves using both the methods. The dashes in E_z column for the first four examples mean that they are planar curves. Table 3 clearly reveals that the proposed parameterization method yields far smaller error-of-fit values than those obtained using the chord length parameterization method. Thus, we can conclude that the proposed parameterization method performs well in terms of both fitting of visually appealing Bezier curves and yielding much smaller error-of-fit values.

Table 2. Space curves approximated by chord length and proposed parameterization methods

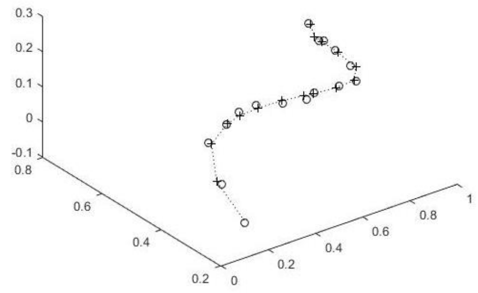
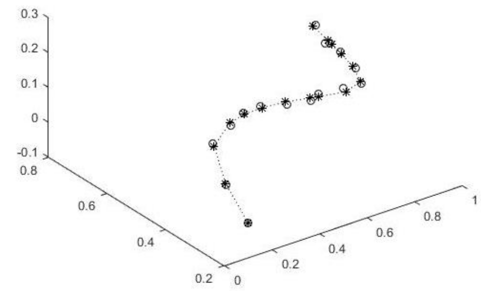
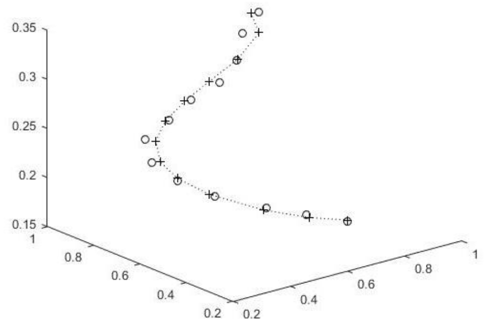
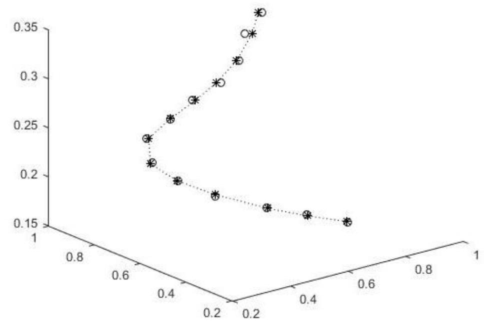
Example No.	Approximation Method	
	Chord Length Parameterization	**Proposed Parameterization**
S1		
S2		

Table 3. Error-of-fit values yielded by both parameterization methods for the sample curves considered

Example No.	Chord length parameterization method			Proposed parameterization method		
	E_x	E_y	E_z	E_x	E_y	E_z
P1	0.0200	0.1089	-	7.12E-04	2.36E-04	-
P2	0.0050	0.0648	-	6.93E-05	1.54E-04	-
P3	0.0010	0.0400	-	8.00E-04	5.00E-05	-
P4	0.0020	0.0100	-	2.80E-04	4.90E-04	-
S1	0.0012	0.0033	8.84E-04	2.77E-04	3.64E-04	4.78E-04
S2	0.0037	0.0040	5.40E-06	3.35E-04	8.40E-04	8.91E-06

V. CONCLUSIONS AND FUTURE SCOPE

Parameterization is a crucial step in fitting of parametric curves and surfaces. This paper introduces a new method to address the parameterization problem in fitting of Bezier curves by approximating the given point cloud data in the least squares sense. Based on the experimentation using the point clouds of curves of varying complexities, it has been found that the proposed parameterization method yields visually appealing curves along with very small fitting errors.

An extension of the proposed method to fitting of the Bezier surfaces is being reported separately. Efforts are underway to extend the proposed parameterization method for fitting of B-Spline curves and surfaces also.

REFERENCES

[1]. W. Ma & J. P. Kruth (1995) "*Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces*", Computer Aided Design, Vol. 27, pp. 663-675.

[2]. J. Barhak & A. Fisher (2001) "*Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques*", IEEE Transactions on Visualization and Computer Graphics, Vol. 7, Issue 1, pp. 1-16.
[3]. G. S. Kumar, P. K. Kalra & S. G. Dhande (2003) "*Parameter optimization for B-Spline curve fitting using genetic algorithms*", Evolutionary Computation, CEC '03, Vol. 3, pp. 1871-1878.
[4]. A. Galvez, A. Iglesias, A. Cobo, J. P. Pey & J. Espinola (2007) "*Bezier curve and surface fitting of 3D point clouds through genetic algorithms, functional networks and least squares approximation*", Proc. of International Conference on Computational Science and its Applications, Vol. 4706, pp. 680-693.
[5]. Y. H. Sun, W. J. Xiang & D. S. Xia (2010) "*Parameter optimization for B-Spline curve fitting based on adaptive genetic algorithm*", Journal of Computer Applications, Vol. 30, pp. 1878-1882.
[6]. L. Zhao, J. Jiang, C. Song, L. Bao & J. Gao (2013) "*Parameter optimization for Bezier curve fitting based on genetic algorithm*", Advances in Swarm Intelligence, Vol. 7928, pp. 451-458.
[7]. A. Y. Hasegawa, C. Tormena & R. S. Parpinelli (2014) "*Bezier curve parameterization using a multi-objective evolutionary algorithm*", International Journal of Computer Science and Applications, Vol. 11, No. 2, pp. 1-18.
[8]. A. Galvez & A. Iglesias (2013) "*A new iterative mutually coupled hybrid GA-PSO approach for curve fitting in manufacturing*", Applied Soft Computing, Vol. 13, pp. 1491-1504.
[9]. C. Loucera, A. Galvez and A. Iglesias (2014) "*Simulated annealing algorithm for Bezier curve approximation*", Proc. of the International Conference on Cyberworlds (CW), Santander, pp. 182-189.
[10]. A. Galvez, A. Iglesias & A. Avila (2014) "*Applying clonal selection theory to data fitting with rational Bezier curves*", Proc. of the International Conference on Cyberworlds (CW), Santander, pp. 221-228.
[11]. A. Iglesias, A. Galvez & A. Avila (2016) "*Hybridizing mesh adaptive search algorithm and artificial immune systems for discrete rational Bezier curve approximation*", The Visual Computer, Vol. 32, No. 3, pp. 393-402.
[12]. J. Brownlee (2012) Clever Algorithms, www.cleveralgorithms.com, accessed on 26th May 2016.
[13]. A. Galvez & A. Iglesias (2013) "*Firefly algorithm for Bezier curve approximation*", Proc. of 12th International Conference on Computational Science and its Applications, pp. 81-88.
[14]. A. Galvez & A. Iglesias (2013) *Firefly algorithm for explicit B-Spline curve fitting to data points,* Mathematical Problems in Engineering, Vol. 2013, http://dx.doi.org/10.1155/2013/528215.
[15]. A. Iglesias & A. Galvez (2015) "*Memetic firefly algorithm for data fitting with rational curves*", Proc. of the IEEE Congress on Evolutionary Computation, pp. 507-514.
[16]. A. Iglesias, A. Galvez & M. Collantes (2015) "*Bat algorithm for curve parameterization in data fitting with polynomial Bezier curves*", Proc. of the International Conference on Cyber Worlds, pp. 107-114.
[17]. A. Iglesias, A. Galvez & M. Collantes (2015) "*Global-support rational curve method for data approximation with Bat algorithm*", Artificial Intelligence Applications and Innovations, Vol. 458, IFIP Advances in Information and Communication Technology, pp. 191-205.
[18]. A. Iglesias & A. Galvez (2015) "*Memetic electromagnetism algorithm for finite approximation with rational Bezier curves*", Advances in Swarm and Computational Intelligence, Vol. 9140, pp. 30-40.
[19]. Y. Tan & Y. Zhu (2010) "*Fireworks algorithm for optimization*", Advances in Swarm Intelligence, Vol. 6145, pp. 355-364.
[20]. M. E. Mortenson (2006) *Geometric Modelling*, 3rd Edition, Industrial Press Inc.

Authors

K. Srinivas Reddy received his B. Tech. (Mechanical Engineering) degree from JNTU, Kakinada. He is currently pursuing his M. Tech. (Manufacturing Engineering) degree at National Institute of Foundry & Forge Technology, Ranchi. His research interests include computer aided design and other aspects of manufacturing engineering.

Amitava Mandal received his B. Tech. (Mechanical Engineering), M. Tech. (Mechanical Engineering) and Ph. D. (Mechanical Engineering) degrees from Indian Institute of Technology Delhi. His teaching and research interests include the finite element analysis, non-conventional machining, welding and other aspects of manufacturing engineering.

Krishna Kumar Verma received his M. Sc. (Mathematics) degree from Bhagalpur University, Bhagalpur and M. Phil. (Applied Mathematics) degree from Indian School of Mines, Dhanbad. His teaching and research interests include computer aided design, numerical methods and operations research.

G. Rajamohan received his B. E. (Mechanical Engineering) degree from College of Engineering, Anna University, Chennai, M. E. (Manufacturing Technology) degree from National Institute of Technology, Tiruchirappalli and Ph. D. (Mechanical Engineering) degree from Indian Institute of Technology Madras. His teaching and research interests include machining, metrology and computer aided inspection, image processing and applications of computers in design and manufacturing.