

OPTIMUM DESIGN AND BALANCING OF THRESHER MACHINE

Ph.D. Thesis

PREM SINGH
ID No.: 2014RME9040



DEPARTMENT OF MECHANICAL ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR
JLN MARG, JAIPUR – 302017, INDIA
November, 2019

OPTIMUM DESIGN AND BALANCING OF THRESHER MACHINE

Submitted in

fulfilment of the requirements for the degree of

Doctor of Philosophy

by

PREM SINGH

ID: 2014RME9040

Under the Supervision of

Prof. Himanshu Chaudhary



Department of Mechanical Engineering
Malaviya National Institute of Technology Jaipur
November, 2019

© Malaviya National Institute of Technology Jaipur - 2017.

All rights reserved.

Declaration

I, **Prem Singh**, declare that this thesis titled, “**Optimum Design and Balancing of Thresher Machine**” and the work presented in it, are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this institution.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself, jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date:

Place: Jaipur

Prem Singh
(2014RME9040)

Certificate

This is to certify that the thesis entitled “**Optimum Design and Balancing of Thresher Machine**” being submitted by **Mr. Prem Singh (2014RME9040)** is a bonafide research work carried out under my supervision and guidance in fulfillment of the requirement for the award of the degree of Doctor of Philosophy in the Department of Mechanical Engineering, Malaviya National Institute of Technology, Jaipur, India. The matter embodied in this thesis is original and has not been submitted to any other University or Institute for the award of any other degree.

Place: Jaipur

Date:

Dr. Himanshu Chaudhary

Professor

Mechanical Engineering Department

Malaviya National Institute of

Technology, Jaipur

Jaipur-302017, India

Acknowledgement

I would like to express my deep and sincere gratitude to my thesis supervisor, Dr. Himanshu Chaudhary, for his invaluable guidance and support. He is an excellent teacher and his knowledge and logical way of thinking has been of great value for me. This research is impossible without his inspiring guidance, experience, and subject knowledge. I also take this opportunity to express my heartfelt thanks to the members of Departmental Research Evaluation Committee (DREC), Dr. T. C. Gupta, Dr. Dinesh Kumar and Dr. Amit Singh, who spared their valuable time and experiences to evaluate my research plan and the synopsis. I would also like to thank Prof. Dilip Sharma, Head of the Mechanical Engineering Department, and his office team for helping in all administrative works regarding the thesis.

I extend my thanks to all colleagues of JECRC University and friends, Vimal Pathak, NRNV Gowripathi Rao, and Ramanpreet Singh for their constant motivation and for believing in me. Finally, I cannot refrain myself from giving thanks to my parents, wife Rachana Singh and daughter Vidisha Singh who have surrendered their priority and time for me.

Prem Singh
Department of Mechanical Engineering
Malaviya National Institute of Technology Jaipur

Abstract

The dynamic performance improvement, i.e., smooth motor torque requirement, less vibration, etc. of the thresher machine, is presented in this thesis. Cleaning mechanism, threshing drum, and flywheel are considered to improve the dynamic performance of the thresher machine. The cleaning mechanism is balanced by optimizing the inertial properties of each moving link modelled as the dynamic equivalent system of point masses. The shaking force and shaking moments developed in the cleaning mechanism are derived in term of point mass parameters. Thus, the multi-objective optimization problem to minimize the shaking forces and shaking moments is formulated by considering the point mass parameters as the design variables. The formulated optimization problem is solved using a posteriori approach based algorithm as non-dominated sorting Jaya algorithm (NSJAYA) and a priori approach based algorithms like Jaya algorithm and Genetic algorithm (GA) under suitable design constraints. It is established that NSJAYA is computationally more efficient than the GA and Jaya. The optimal Pareto set for the balancing of the mechanism is determined and outlined. Hence, the user can select any solution based on the importance of the objective function. ADAMS Software is used for the validation of the balanced cleaning mechanism. Besides, the balancing of the threshing drum can improve the dynamic performance of the thresher machine and treated as rigid rotor due to its low speed of rotation. Therefore, the optimum two-plane discrete balancing procedure is proposed for the rigid rotor. The discrete two-plane balancing in which rotor is balanced to minimize the residual effects or the reactions on the bearing supports using discrete parameters such as masses and their angular positions on two balancing planes. A multi-objective optimization problem is formulated by considering reaction forces on the bearing supports as multi-objective functions and discrete parameters as the design variables. These multi-objective functions are converted into a single-objective function using appropriate weighting factors. The formulated problem is solved using the proposed modified Jaya algorithm. It is found that the modified Jaya algorithm is computationally more efficient than the GA algorithm. A number of masses per plane are used to balance the rotor. A comparison of reaction forces using the number of masses per plane is also investigated. The effectiveness of the proposed methodology is tested by the balancing problem of rotor available in the literature. It is also applied for the balancing of the threshing drum. ADAMS software and experimental tests are used for validation of a developed balancing approach.

The shape synthesis of the flywheel is another approach to improve the dynamic performance of the thresher machine. Thus, the optimal shape synthesis procedure of the flywheel using a cubic B-spline curve is proposed. The flywheel plays a vital role in storing kinetic energy in modern machines. Thus, the kinetic energy is an essential parameter to measure flywheel performance and can be improved by optimal thickness distribution of the flywheel, generally known as shape optimization. Therefore, the shape optimization model of the flywheel with maximization of the kinetic energy is formulated using a cubic spline curve under the design constraints like the mass of the flywheel and maximum values of Von Mises stresses. A flow chart is proposed to solve the two-point boundary value differential equation for calculation of Von Mises stress at each point between the inner and outer radius of the flywheel. The control points of the cubic B-spline curve are taken as design variables. Then the formulated problem is solved using particle swarm algorithm (PSO), genetic algorithm (GA), and Jaya algorithm. The effectiveness of the proposed approach is investigated through the design of flywheel taken from the literature and the flywheel design of the thresher machine.

Contents

Declaration.....	i
Certificate	ii
Acknowledgement	iii
Abstract.....	iv
List of Figures.....	ix
List of Tables	xii
List of Symbols and Abbreviations	xiii
CHAPTER 1 INTRODUCTION	1
1.1. Balancing of cleaning mechanism	2
1.2. Balancing of the threshing drum.....	4
1.3. Shape synthesis of flywheel.....	7
1.4. Contributions of the Research	8
1.5. Thesis Outline.....	9
1.6. Summary.....	10
CHAPTER 2 LITERATURE SURVEY	11
2.1. Cleaning mechanisms used to the thresher machine	11
2.2. Methods for balancing of mechanism.....	14
2.3. Methods for balancing of threshing drum	17
2.4. Mixed variables optimization algorithms	20
2.5. Shape synthesis methods of the flywheel	22
2.6. Summary.....	25
CHAPTER 3 OPTIMAL BALANCING OF THE CLEANING MECHANISM ..	27
3.1. Dynamic equations of motion for the rigid body.....	27
3.2. Equations of motion for dynamically equivalent point mass system	28
3.3. Dynamic analysis of the cleaning mechanism.....	30
3.4. Formulation of the optimization problem.....	31
3.5. Optimization algorithm.....	33
3.6. Results and discussions.....	37

3.7.	Summary.....	45
CHAPTER 4 MIXED VARIABLE OPTIMIZATION ALGORITHMS		47
4.1.	Formulation of mixed variable optimization problems	47
4.2.	A modified Jaya algorithm for mixed variable optimization problems.....	49
4.3.	Design Problems	51
4.3.1.	Example 1: Design of a welded beam.....	51
4.3.2.	Example 2: Pressure Vessel Design.....	54
4.3.3.	Example 3: 10-Bar Planar Truss design.....	56
4.3.4.	Example 4: A helical spring design	58
4.3.5.	Example 5: Compound gear train design.....	61
4.4.	Summary.....	63
CHAPTER 5 OPTIMAL TWO-PLANE DISCRETE BALANCING		64
5.1.	Dynamic model of rigid rotor	64
5.2.	Formulation of discrete optimization problem	67
5.3.	Results and Discussions.....	68
5.3.1.	Numerical problem of the rigid rotor	68
5.3.2.	Application - Balancing of the threshing drum.....	76
5.3.2.1.	Experimental study	79
5.4.	Summary.....	83
CHAPTER 6 OPTIMAL SHAPE SYNTHESIS OF THE FLYWHEEL		85
6.1.	A shape optimization model of the flywheel	85
6.1.1.	Parameters of the flywheel.....	87
6.1.2.	Stress analysis of flywheel.....	88
6.2.	Formulation of the optimization problem.....	92
6.2.1.	Design variables	92
6.2.2.	Objective function and constraints.....	92
6.3.	Optimization algorithm.....	93
6.4.	Flywheel design problems	95
6.4.1.	Numerical example of the flywheel design.....	96
6.4.2.	The flywheel design of the thresher machine	99
6.5.	Summary.....	103

CHAPTER 7 CONCLUSIONS.....	105
References.....	109
Appendix A MATLAB Code for Non-Dominated Sorting Jaya Algorithm (NSJAYA)	121
Appendix B MATLAB code for modified Jaya algorithm.....	123
Papers published based on this work	127
Brief Bio-data of the author	128

List of Figures

Fig. 1.1. A CAD Model of the thresher machine	1
Fig. 1.2. Cleaning mechanism in thresher machine.....	3
Fig. 1.3. Schematic diagram of four-bar crank-rocker linkage for cleaning mechanism...3	3
Fig. 1.4. Different types of threshing drum	6
Fig. 1.5. Thickness profile of the flywheel.....	8
Fig. 2.1. Slider-crank mechanism (Garvie and Welbank, 1967).....	11
Fig. 2.2. Mechanism for the cleaning unit (Joshi, 1981).....	12
Fig. 2.3. A crank pitman drive (Tan and Harrison, 1987).....	12
Fig. 2.4. A four-bar cleaning mechanism in the thresher machine.....	13
Fig. 2.5. Application of counterweights for complete force balancing (Berkof and Lowen, 1969).....	14
Fig. 2.6. Complete shaking force and shaking moment balancing using mass redistribution and inertia counterweights (Feng, 1990)	15
Fig. 2.7. Classification of balancing methods (Zhou and Shi, 2001).....	18
Fig. 2.8. General shape of the flywheel.....	23
Fig. 2.9. Topology optimization model of the flywheel (Jiang and Wu, 2016).....	25
Fig. 3.1. The i th moving link of the mechanism	27
Fig. 3.2. Dynamically equivalent point mass systems for i th moving link of the cleaning mechanism	29
Fig. 3.3. Reaction forces and moments with respect to fixed link in cleaning mechanism	30
Fig. 3.4. A flow chart of Jaya algorithm for balancing of the cleaning mechanism	34
Fig. 3.5. A flow chart for the non-dominated sorting Jaya algorithm	35
Fig. 3.6. Convergence of the best values of the objective function for GA and Jaya algorithm in all cases	39
Fig. 3.7. Variation in shaking force, shaking moment and driving torque with time for one cycle of the crank.....	40
Fig. 3.8. Pareto front for a cleaning mechanism.....	42
Fig. 3.9. Dynamic performance of the cleaning mechanism for one cycle of the crank. .	43
Fig. 3.10. Simulation of the cleaning mechanism using ADAMS software	44
Fig. 3.11. Input torque variation in the original and optimized cleaning mechanism for the complete cycle of crank	45

Fig. 4.1. A modified Jaya algorithm for mixed variable optimization problems	50
Fig. 4.2. Welded beam design (Nema et al., 2008)	52
Fig. 4.3. Convergence graph of best and mean values of objective function for welded beam design	53
Fig. 4.4. Pressure vessel design (Sandgren, 1990)	54
Fig. 4.5. Convergence of best and mean values of objective function for pressure vessel design.....	55
Fig. 4.6. A planar 10 bar truss structure(Rajeev and Krishnamoorthy, 1992)	56
Fig. 4.7. Convergence of best and mean values of objective function for planar 10-bar planar truss.....	57
Fig. 4.8. Design of helical spring (Nema et al., 2008)	59
Fig. 4.9. Convergence characteristic of objective function for spring design	60
Fig. 4.10. Gear train design (Guo et al., 2004).....	61
Fig. 4.11. Convergence graph of best and mean values of objective function for gear train design.....	62
Fig. 5.1. Rigid rotor model	65
Fig. 5.2. Convergence of the best objective function values in different cases of static balancing.....	70
Fig. 5.3. Convergence of the best objective function values in different cases of dynamic balancing.....	70
Fig. 5.4. Modeling of the rigid rotor in MSC ADAMS.....	72
Fig. 5.5. Validation of reaction forces in x and y -directions at supports P using ADAMS for static balancing.....	72
Fig. 5.6. Validation of reaction forces in x and y -directions at supports Q using ADAMS for static balancing.....	74
Fig. 5.7. Validation of reaction forces in x and y -directions at supports P using ADAMS for dynamic balancing	74
Fig. 5.8. Validation of reaction forces in x and y -directions at supports Q using ADAMS for dynamic balancing	75
Fig. 5.9. Convergence of the best objective function values in different cases for GA and modified Jaya algorithm	77
Fig. 5.10. Variation of reaction forces in x and y -directions at support P for the balancing of the threshing drum.....	78

Fig. 5.11. Variation of reaction forces in x and y -directions at support Q for the balancing of the threshing drum.....	79
Fig. 5.12. The experiment setup for the unbalanced threshing drum	80
Fig. 5.13. Convergence of the best objective function values in different cases.....	82
Fig. 6.1. 2D symmetric model of flywheel.....	86
Fig. 6.2. B-spline curve segments	86
Fig. 6.3. A flow chart for solving the piece wise equation with two-point boundary value condition	91
Fig. 6.4. A flow chart of Jaya algorithm for the shape of the flywheel.....	94
Fig. 6.5. Convergence of the best objective function value in PSO, GA, and Jaya.	97
Fig. 6.6. Optimal thickness distribution along the radial direction	98
Fig. 6.7. Optimized shape correspond to Jaya and conventional method	98
Fig. 6.8. Stress distribution along the radial direction for the optimized shape of the flywheel	99
Fig. 6.9. Convergence of the best objective function value in PSO, GA, and Jaya.	100
Fig. 6.10. Original shape and optimized shape of the flywheel	101
Fig. 6.11. A Cad model of the Optimized flywheel	102
Fig. 6.12. Stress distribution along the radial direction in (a) original flywheel (b) optimized flywheel	102
Fig. 6.13. Torque variation in the original and optimized flywheel.....	103

List of Tables

Table 2.1. Algorithms proposed to solve mixed variable optimization variables.....	20
Table 3.1. Link parameters of the cleaning mechanism	37
Table 3.2. Optimized values of link parameters for the balanced mechanism for all cases	38
Table 3.3. The RMS values of the balanced mechanism for all cases	41
Table 3.4. Non-dominated solutions acquired by NSJAYA.....	42
Table 3.5. The RMS values of the dynamic quantities for the cleaning mechanism.....	44
Table 3.6. Optimum parameters of the balanced cleaning mechanism using solution 1.45	
Table 4.1. Optimum designs of the welded beam.....	53
Table 4.2. Optimum solutions for design of pressure vessel	55
Table 4.3. Comparison of Optimum solutions for planar 10-bar truss structure	58
Table 4.4. Optimal solutions comparison for the spring design	60
Table 4.5. Optimal design for the compound gear train	62
Table 5.1. Dimensions of the rigid rotor in (m).....	68
Table 5.2. Parameters of a rigid rotor	68
Table 5.3. Parameters of GA and Discrete Jaya algorithm for numerical problems	71
Table 5.4. Comparison of modified Jaya algorithm to GA (Messenger and Pyrz, 2013) for all cases.....	73
Table 5.5. Dimensions of the threshing drum in (m).....	76
Table 5.6. Comparison of modified Jaya algorithm to GA for all cases.....	78
Table 5.7. Optimal solutions for all cases.....	82
Table 6.1. Material properties of the flywheel.....	96
Table 6.2. Design parameters of the flywheel	96
Table 6.3. The optimum value for the shape of the flywheel	97
Table 6.4. Material properties of flywheel.....	99
Table 6.5. Design parameters of the flywheel	99
Table 6.6. The optimum value for the shape of the flywheel	101

List of Symbols and Abbreviations

\mathbf{r}_i	Position vector of the origin o_i of the i th link, from origin o_{i+1} of the $(i+1)$ th link
G_i	The mass center of i th link
\mathbf{r}_{ig}	Position vector of the mass center G_i of i th link, from origin o_i
$\overline{F_{sh}}, \overline{M_{sh}}$	Normalized shaking force and shaking moment in the mechanism
a_{ij}	Distance of j th point mass from origin o_i of i th link
w_1, w_2	Weighting factors for objective functions
δ_i	Angular position of i th link
θ_i	Angular position of mass center G_i of i th link in the local frame
θ_{ij}	Angular position of point mass m_{ij} in the local frame
ω_i	Angular velocity of i th link
$\dot{\omega}_i$	Angular acceleration of i th link
ρ	Material density of link's material
$\mathbf{x}^c, \mathbf{x}^d, \mathbf{x}^{int}$	The vector of continuous, discrete and integer variables, respectively
D_i, G_i	Set of discrete values and integer values for the i th variable, respectively
MDHGA	Mixed discrete hybrid genetic algorithm
HPB	Hybrid particle swarm branch and bound
MPSO	Modified particle swarm optimization.
TLBO	Teaching-Learning-Based Optimization
NSJAYA	Non-dominated sorting Jaya algorithm
GA	Genetic algorithm
PSO	Particle swarm optimization
SA	Simulated annealing algorithm
HSIA	Hybrid swarm intelligence approach
EP	Evolutionary programming approach
EA	Evolutionary Algorithm
DE	Differential evolution
I_G	The centroid inertia tensor of rotor
m_{ij}, α_{ij}	j th mass and corresponding angular position in the i th plane
N_i	The number of balancing masse for i th plane

F_{PO}, F_{QO}	The reaction forces at supports P and Q for unbalance rotor, respectively
D_1, D_2	Total numbers of discrete values of masses and discrete values of corresponding angular position, respectively
\tilde{m}_i	Continuous balancing solution of mass for ith plane
$\tilde{\alpha}_i$	Continuous balancing solution of angular position for ith plane
S	curve segments
p_i	ith control points
E_k	Kinetic energy stored in the flywheel
σ_a	Allowable stress for the material of the flywheel

Vectors are shown by bold face

Chapter 1

Introduction

Threshing is the essential post-harvest process and used to remove the edible part of grains from the harvested crops. It can be done either by the conventional methods or the thresher machine. Hand-beating and bullock-treading of harvested crops are convention methods of threshing (Joshi, 1981). But, these methods are time-consuming, uneconomical, and laborious.

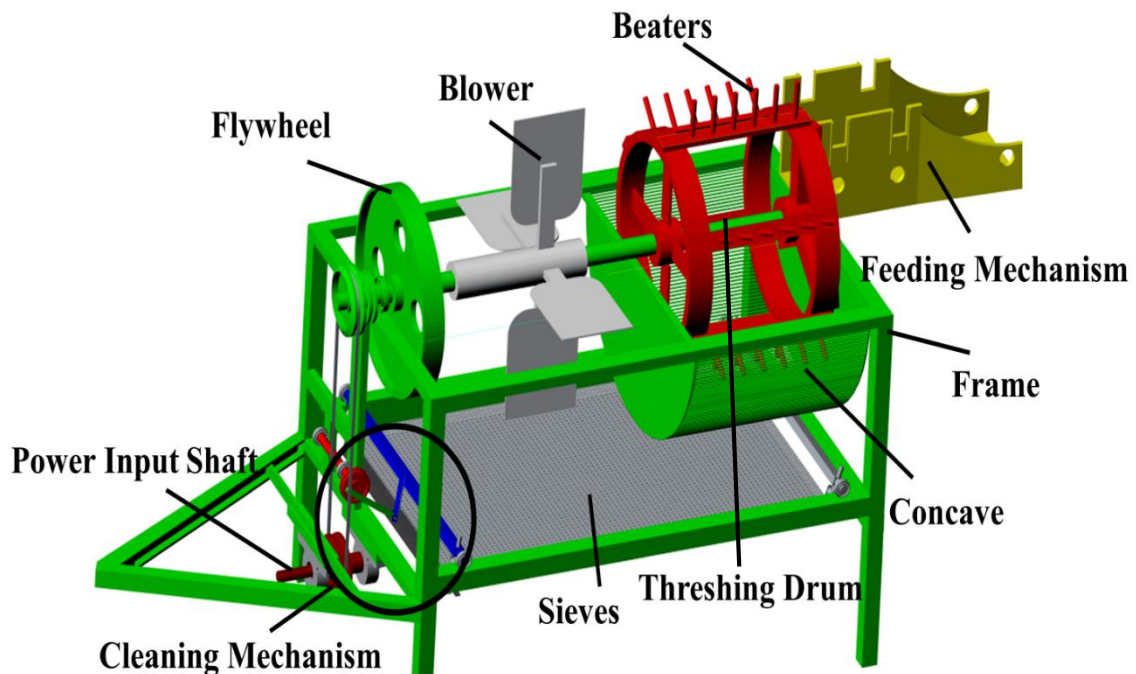


Fig. 1.1. A CAD Model of the thresher machine

Therefore, the thresher machine has been developed to overcome the difficulties of the conventional methods. It was first invented in 1786 by Scottish mechanical engineer Andrew Meikle. Generally, it is designed for multi-crops like mustard, wheat, millet, etc. and can be operated by the animal power or a diesel engine and tractor PTO (power take off) shaft (Ghaly, 1985). But, animal operated threshers have low cleaning efficiency and low output capacity. Thus, the diesel engine and tractor PTO shaft operated threshers have become more popular and used to detach the grains from the harvested crops with minimum time and minimum efforts. Normally, a thresher has three mechanisms, namely, feeding, threshing, and cleaning, as shown in Fig.1.1. In the feeding mechanism, dry crops feed manually using a suitable conveyor system or feeding hopper to threshing mechanism (Stout and Cheze, 1999). Threshing mechanism consists of a threshing drum with beaters. Rotation motion of the threshing drum

detaches the grains from the harvested crops by combined actions of impact and beating (Olaoye et al., 2010). After that, the grains pass through a concave into the cleaning mechanism. Generally, the cleaning mechanism is a four-bar crank-rocker mechanism in which sieves are attached to the rocker that separates grains from the husk and foreign materials (Test Code IS:6284, 1985).

The improvement in the dynamic performance of a thresher machine is a challenging task. In this study, the dynamic performance of the thresher machine is improved by balancing of cleaning mechanism, balancing of threshing drum, and the shape synthesis of the flywheel.

1.1. Balancing of cleaning mechanism

The dynamic performance of thresher machine can be improved by balancing of cleaning mechanism. Cleaning mechanism consists of three sieves which remove grains from straw and plant debris by the oscillation of sieves and operated by PTO shaft of the tractor as shown in Fig.1.1. This oscillation of sieves generates the vibrations. As a result, the unbalanced shaking force and shaking moment are developed on the support of the machine. Thus, balancing of cleaning mechanism is essential for its dynamic performances. Generally, cleaning mechanism is a four-bar crank-rocker mechanism. It consists of four links, namely, pulley driven by the belt as link #1, connecting rod as link #2, sieves carrier as link #3, and frame of thresher as link #0 as shown in Fig. 1.2. Figure 1.3 represents the four bar crank-rocker linkage for cleaning mechanism.

The balancing of shaking force and shaking moment in the cleaning mechanism depends on the redistribution of moving link masses. However, complete shaking force balancing can be done using either counterweight (Walker and Oldham, 1978) or the redistribution of the masses (Kochev, 1987). But, this balancing method increases other dynamic quantities in the mechanism while combined shaking force and shaking moment balancing methods like disk or inertia counterweights (Berkof, 1973; Esat and Bahai, 1999), moment balancing idler loops (Bagci, 1979), and a duplicate mechanism (Arakelian and Smith, 2005) increase the weight and complexity of the mechanism.

To overcome these difficulties; formulation of the multi-objective optimization problem has been proposed to reduce the shaking force and shaking moment. Conventional and evolutionary optimization algorithms are used to solve the formulated problem.

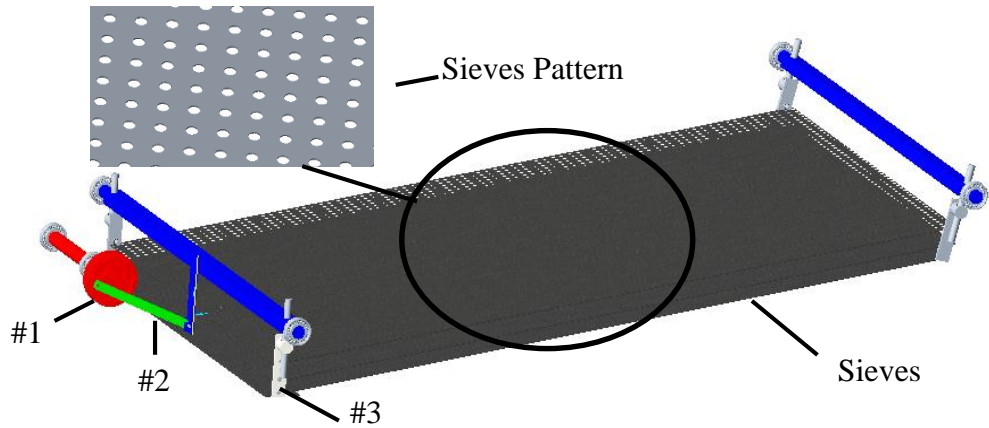


Fig. 1.2. Cleaning mechanism in thresher machine

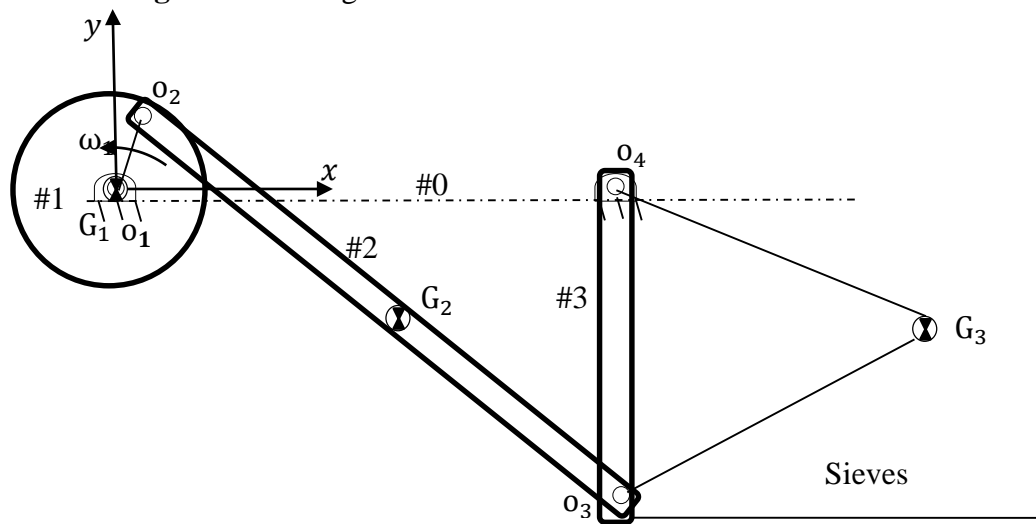


Fig. 1.3. Schematic diagram of four-bar crank-rocker linkage for cleaning mechanism

Note that conventional optimization algorithms require the initial solutions to find an optimal solution. Thus, these algorithms give the local solution to balance the mechanism. While evolutionary optimization algorithms use the priori approach, generally, the multi-objective optimization problem can be solved using two approaches as a priori approach and a posteriori approach (Rao et al., 2019). In a priori approach, the multi-objective optimization problem is converted into a single objective optimization problem using appropriate weights for each objective function (Rao and Saroj, 2016). This approach gives a unique optimal design in each simulation run. Thus, it generates multiple optimal solutions by running the algorithm multiple time with a different combination of weights (Deb et al., 2002).

Moreover, the optimum results obtained by this approach depend on weights assigned to objective functions. Assigning the weights to objective functions is difficult for an uncertain scenario. These limitations are eliminated by the posteriori approach. In this approach, weights are not assigned to the objectives before the start of the algorithm. It provides Pareto optimal solutions in a single run of the algorithm, and a designer can choose appropriate solutions based on the importance of objective functions. Moreover, the posteriori approach is computationally more efficient than the priori approach. Therefore, a posteriori approach based optimization algorithm is required to develop for the balancing problem of the mechanism.

1.2. Balancing of the threshing drum

Besides the balancing of the cleaning mechanism, the balancing of the threshing drum plays a vital role in the improvement of the dynamic performance of the thresher machine. The threshing drum is an essential element of the thresher machine. It removes the grains from the panicles by its combined action of impact and rubbing. The threshing drum consists of two wheels with blades attached between them. Generally, it can be classified like hammer mill or beater type, rasp bar type, wire loop type, syndicator type, spike tooth/peg type based on the design of the blades as shown in Fig. 1.4. Recently, spike tooth/peg type threshing drum is used for threshing of the harvested crop in India due to low power requirement, lower human injuries, and good quality of the chaff. But, the rotation action of this threshing drum develops the harmful vibration effects on the bearings. These vibration effects enhance human accidents and torque requirement, and also affect productivity. Therefore, the balancing of the threshing drum becomes an essential task for designers and engineers. Generally, individual wheel of the threshing drum is balanced statically using hit and trial method in which material is removed from a large spot (Prashad and Sharma, 1985). Thus, this method of balancing increases the computational time and also decrease the strength of the drum. However, the threshing drum is balanced in a similar manner of a rigid rotor due to its low speed of rotation. Generally, the rotor can be classified as rigid and flexible. Although, rigid rotors have negligible deformation owing to the high ratio of diameter to length and low speed of rotation, while flexible rotors have deformations because of high speeds of operation and long lengths (Darlow, 1989).

Further, the threshing drum can be balanced by attaching correction masses at a single plane and two planes. There are various balancing methods for balancing of

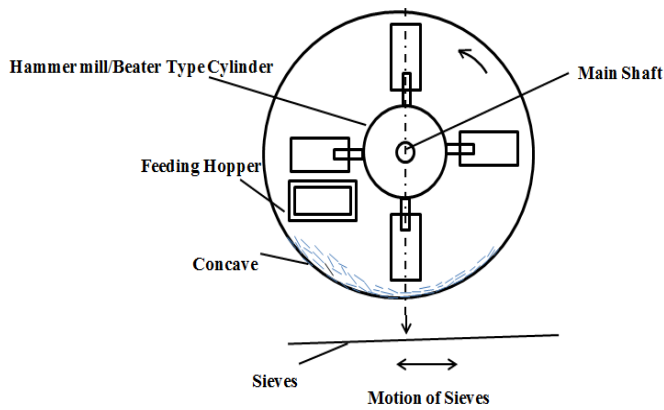
threshing drum as off-line, online or active, and optimization methods. However, off-line balancing methods need more time for calculation of correction masses and do not change the position of the unbalance during operation. Moreover, online or active balancing methods increase the cost of the system due to design the specific bearings and mass redistribution devices.

Besides experimental and theoretical balancing methods, researchers have been focused on the optimization methods to find the optimal correction masses. However, these optimization methods provide real values for correction masses and the corresponding angular positions to balance the threshing drum. Moreover, the discrete solution nearest to the continuous optimal solution is generally used due to a limited set of available standardized mass values. Thus these solutions induce a dynamic effect and residual unbalance. This dynamic effect and residual unbalance can be decreased by recalculating and applying additional correction masses to the balance planes. But, this iterative process increases the balancing cost and time.

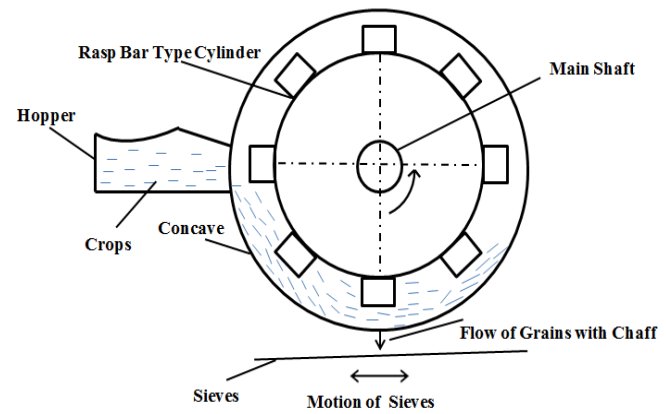
To overcome these difficulties; mixed variable optimization problems are explored. Whereas the multi-objective optimization problem is formulated to minimize the reaction forces on the bearings for the balancing of the threshing drum and the number of discrete masses per plane and corresponding predefined angular positions are treated as the design variables.

Generally, the optimization problems can be classified as a continuous variable, discrete/integer variable, and mixed variable optimization problems. In continuous variable optimization problems, variables are spaced within the bounds. Further, discrete/integer variables have the predefined set of standard values. Whereas, mixed variable optimization problems deal with the integer/discrete and continuous variables. Most of the researchers have been made efforts on continuous variable optimization algorithms. But, these algorithms are not sufficient to solve the practical design problems like structural design, a number of bolts for a connection, a standard diametric pitch of the gear, a number of the teeth of a gear, etc. in which variables are discrete and integer (Arora et al., 1994). Therefore, in recent years, mixed variable optimization methods are developed for practical design optimization problems. However, mixed-variable optimization problems can be solved by classical and evolutionary techniques. Sequential linear programming, Branch and bound methods, rounding-off techniques based on continuous variables, etc. are classical techniques (Guo et al., 2004). But, these

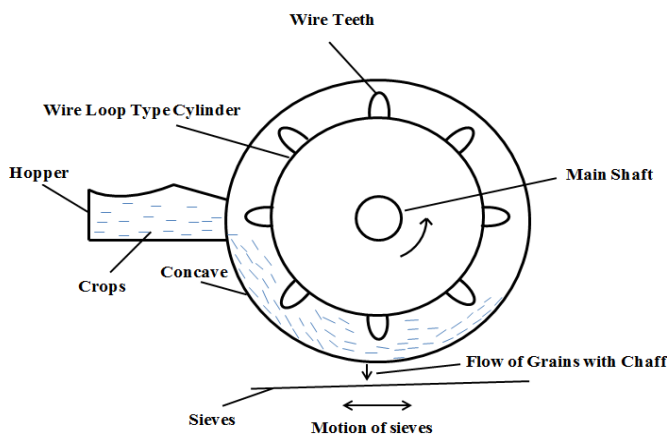
methods include more computational efforts due to the determination of derivatives and Hessian matrixes, and also provide the local solution.



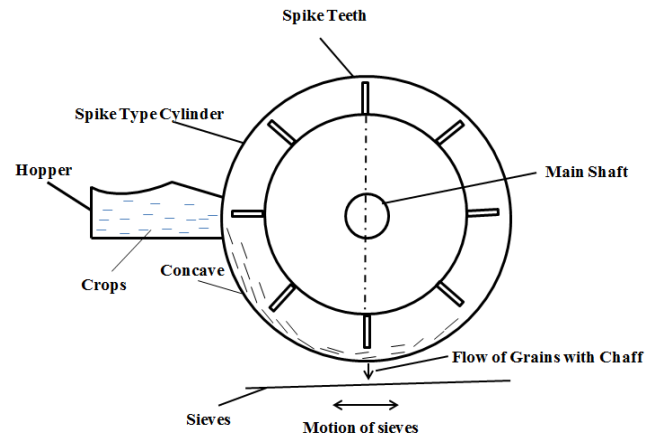
(a) Hammer or Beater type cylinder



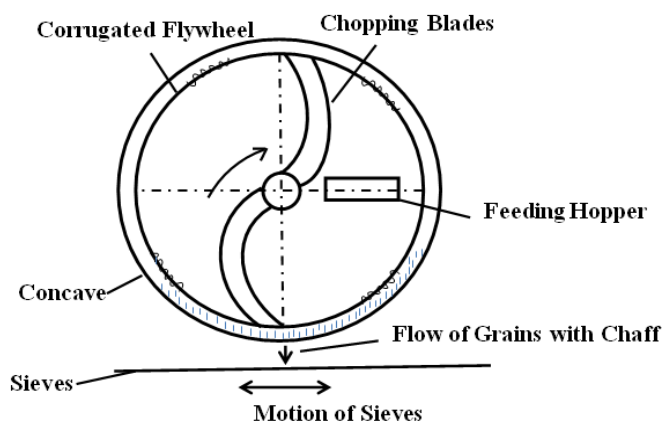
(b) Rasp bar type Cylinder



(c) Wire loop type cylinder



(d) Spike or Peg type cylinder



(e) Syndicator type cylinder

Fig. 1.4. Different types of threshing drum

Recently, evolutionary optimization algorithms are used to solve mixed-variable optimization problems. However, performances of these algorithms depend on the requirement of algorithmic parameters for their convergence. The selection of these parameters enhances the complexity of algorithms. Moreover, these algorithms cannot guarantee to give global solutions within a definite time. Further, the convergence of these algorithms is slow, thus increases the computational efforts.

Therefore, the original Jaya algorithm is modified for solving mixed variable optimization problems. Initially, it is proposed for continuous variable optimization problems by Rao (Rao, 2016). It does not need any algorithmic parameters for its convergence. Thus, the proposed algorithm can handle all types of variables and is suitable for the balancing of a threshing drum.

1.3. Shape synthesis of flywheel

The shape synthesis of the flywheel is the other approach to improve the dynamic performance of the thresher machine. The flywheel plays a vital role in the thresher machine and minimizes the variations in the speed of PTO shaft due to torque fluctuations in a threshing drum with the help of its kinetic energy. Therefore, the design of the flywheel should be efficient, so that fuel consumption, vibrations, and human accidents could be minimized. Performance of the flywheel can be improved by increasing its kinetic energy. Further, kinetic energy ($\frac{1}{2} I\omega^2$) depends on its rotating speed, material density, and geometry. But, high rotating speed will transfer undesirable centrifugal force, as a result of that, high stresses will develop in the machine.

Recently, the large mass of flywheel is used to increase the kinetic energy in the conventional thresher machine. However, the available space of machine limits the large mass of the flywheel, and it also transfers large force on the bearings.

Thus, researchers have been made efforts on the shape optimization of the flywheel rotor geometry. It is an efficient method to increase the kinetic energy by optimizing the rotor thickness distribution along a radial direction. In literature, Polynomial expansion, Fourier series, Fourier sine series, and Bezier curves have been used for the representation of the thickness distribution of the flywheel as shown in Fig.1.5. Unfortunately, the polynomial expansion, Fourier series, and Fourier sine series are limited by the number of coefficients and do not describe the degree of the resulting curve. However, Bezier curves describe the degree of the curve, but these curves are controlled globally and are limited by control points. Moreover, conventional

optimization techniques and finite element analysis have been used to find the optimal thickness distribution of the flywheel. These methods increase the computational efforts and are also less efficient.

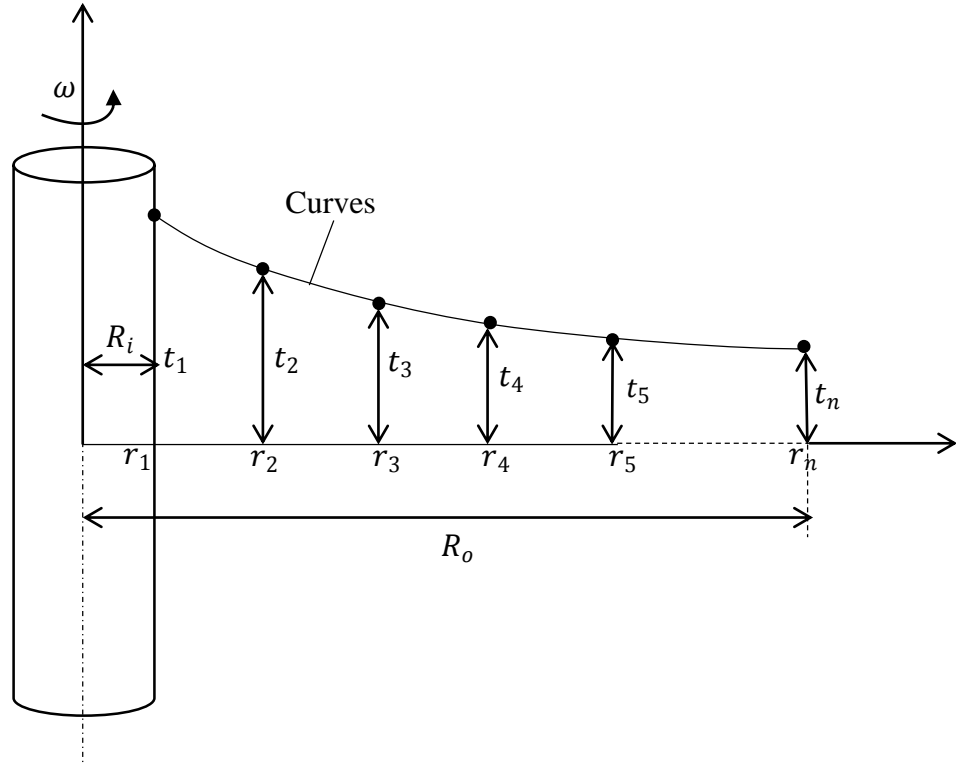


Fig. 1.5. Thickness profile of the flywheel

In this research, the shape optimization model of the flywheel is formulated using a cubic B-spline curve. The B-Spline curves have local control and are not limited by control points. Then, the optimization problem is solved by GA, PSO, and Jaya algorithm. The proposed approach is tested by flywheel design taken from literature and the flywheel design of an agricultural thresher machine. It is observed that the Jaya algorithm provides better results than the other algorithms and stores more energy compared to that of the original flywheel.

1.4. Contributions of the Research

The contributions of this research work are described below.

1. A multi-objective optimization problem formulation for the balancing of the cleaning mechanism is proposed.
2. A posteriori approach based algorithm as a non-dominated sorting Jaya algorithm (NSJAYA) is applied to find the optimal mass distribution of the links for the proposed multi-objective optimization problem.

3. A discrete optimization problem for two-plane balancing of the rigid rotor is proposed.
4. A modified Jaya algorithm is proposed to find the optimal discrete solutions. ADAMS software and the experimental test are used for validation.
5. A shape optimization model for the flywheel using the cubic B-spline curve is proposed.
6. Particle swarm algorithm (PSO), genetic algorithm (GA), and Jaya algorithm are used to solve the shape optimization problem. It is observed that Jaya algorithm is computationally more efficient than GA and PSO.

1.5. Thesis Outline

This thesis contains seven chapters structured as follows:

Chapter 1: Introduction

The aim and motivation of the research work to evolve the procedure for balancing of the cleaning mechanism and threshing drum, and shape synthesis of the flywheel is described in this chapter. A posteriori based optimization approach, Jaya algorithm for mixed variables, and nature-inspired algorithms are also introduced. The main contributions of the research work and the organization of the thesis are also highlighted in this chapter.

Chapter 2: Literature Survey

This chapter surveys the research work on various mechanisms of the cleaning unit, balancing methods of mechanisms, threshing drum balancing methods, mixed variables optimization algorithms, shape synthesis methods of the flywheel, and natural-inspired optimization algorithms. Finally, research gaps are outlined based on the literature surveys.

Chapter 3: Optimal Balancing of the Cleaning mechanism

The dynamic balanced mechanism for cleaning mechanism used in the thresher machine using a dynamically equivalent system of point masses is explained in this chapter. In order to balance the mechanism, the multi-objective optimization problem with minimization of shaking forces and shaking moments is formulated by considering the point mass parameters as the design variables. Also, the optimal Pareto set for the balancing of the mechanism is measured and outlined in this chapter.

Chapter 4: Mixed Variable Optimization Algorithms

This chapter explains the classical and evolutionary mixed variable optimization algorithms. It discriminates the influence of the algorithmic control parameters to find the optimal solution for mixed variable optimization algorithms. Moreover, the modified Jaya algorithm is proposed to solve the mixed variable optimization problems. The efficiency of the proposed algorithm is tested through the design problems taken from the literature.

Chapter 5: Optimal Two-Plane Discrete Balancing

Two-plane discrete balancing procedure for the rigid rotor is explained in this chapter. A multi-objective optimization problem is formulated by considering reaction forces on the bearing supports as multi-objective functions and discrete parameters on each balancing plane as design variables. The effectiveness of the proposed methodology is tested by the balancing problem of rotor available in the literature, and, it is also applied to the unbalanced threshing drum of the thresher machine. This chapter also investigates a comparison of reaction forces using the number of masses per plane.

Chapter 6: Optimal shape synthesis of the flywheel

This chapter presents the optimal shape synthesis of the flywheel using a cubic B-spline curve. The shape optimization model is formulated to maximize the kinetic energy of the flywheel. This chapter also describes the effectiveness of the proposed approach through the numerical problem of the flywheel design taken from literature and the existing flywheel design of the thresher machine.

Chapter 7: Conclusions

This chapter presents the important results obtained in this research work. The contributions and future scope of this research work are also highlighted.

1.6. Summary

The balancing method for the cleaning mechanism and threshing drum, the shape optimization of the flywheel, and corresponding optimization algorithms are explained in this chapter. It highlights the objective and motivation of the current thesis and, brief information about the thesis outline is also provided.

Chapter 2

Literature Survey

This chapter presents the detailed review for various cleaning mechanisms used in thresher machine, different balancing methods of these mechanisms, threshing drum balancing methods, and the shape synthesis methods of the flywheel. It also explores the review of the mixed variable optimization algorithms to find the optimal discrete solutions for balancing of threshing drum. Further, the methods for flywheel shape synthesis are also reviewed.

2.1. Cleaning mechanisms used to the thresher machine

Generally, there are different agricultural operations, but the cleaning of grains from husk and foreign materials is an essential agricultural operation. It can be done by natural air and mechanical fan (Simonyan and Yiljep, 2008). But, natural air is limited by its speed and random direction as a result of that the losses of grains. Moreover, the grains from the chaff and plant debris are separated using air generated by the mechanical fans (Gorial and O'callaghan, 1991a, 1991b). But, these fans increase the cost, complexity of the system, and labor requirement.

Therefore, the grains are separated from husk and foreign materials using the cleaning mechanism incorporate in the thresher machine. The cleaning mechanism consists of three sieves which separate grains from husk and foreign materials by the oscillating motion of sieves. Although, different cleaning mechanisms are used in the thresher machine and some of these are explored here.

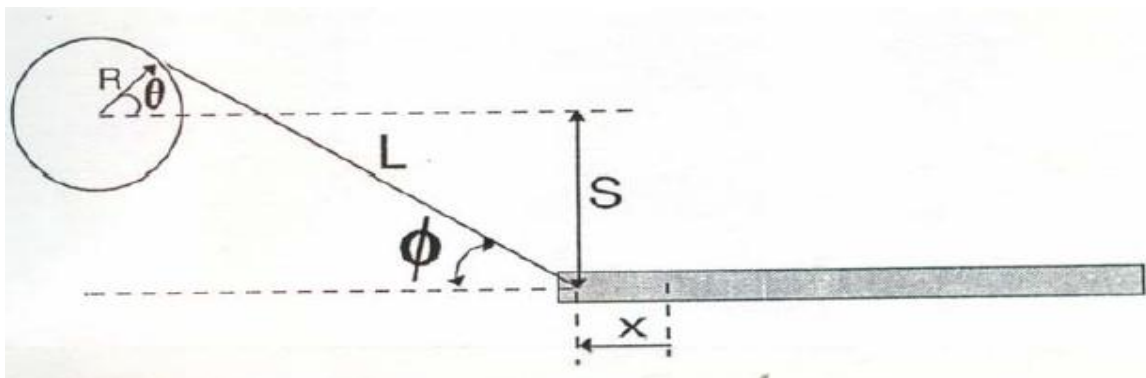


Fig. 2.1. Slider-crank mechanism (Garvie and Welbank, 1967)

A slider-crank mechanism is introduced for cleaning mechanism as shown in the Fig.2.1. It is the oldest mechanism and converts the rotational motion of the crank into the reciprocating motion of the sieves (Garvie and Welbank, 1967). Sieves are driven by the crank through the connecting rod. The reciprocating motion of sieves is used for cleaning of grains. But, this mechanism requires more power for its operation and develops excessive vibrations.

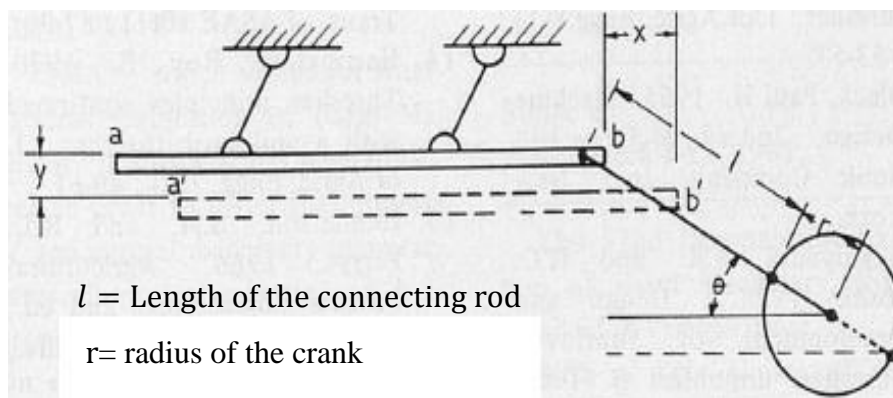


Fig. 2.2. Mechanism for the cleaning unit (Joshi, 1981)

Further, the cleaning mechanism is developed by the researcher (Joshi, 1981). In this mechanism, sieves are attached to the four hinged rods and, reciprocate by an eccentric crank through connecting rod as shown in Fig.2.2. But, the mechanism is limited by grain losses and unwanted vibrations.

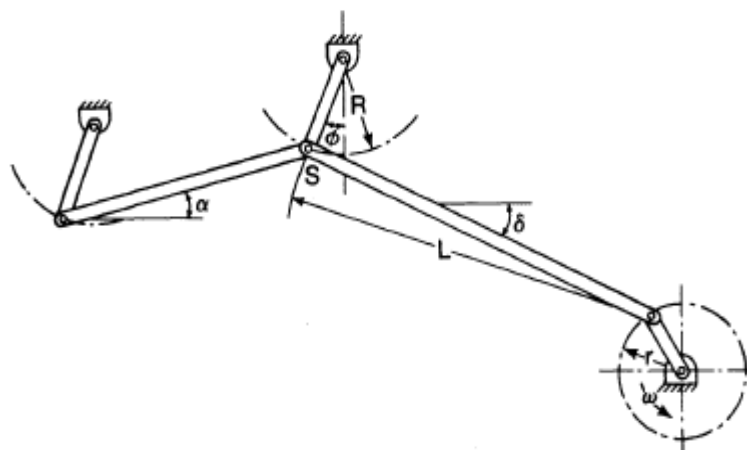


Fig. 2.3. A crank pitman drive (Tan and Harrison, 1987)

A crank pitman drive is applied to the oscillation of the sieves as shown in Fig.2.3. It is concluded that this drive is better than other mechanisms based on the kinematic analysis (Tan and Harrison, 1987). However, the mechanism generates excessive vibrations.

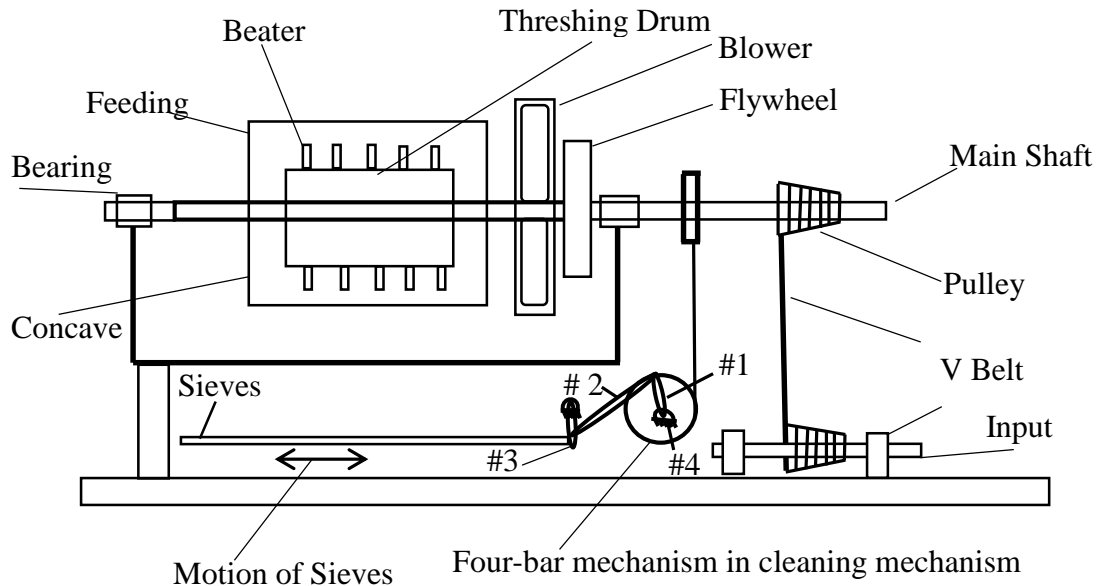


Fig. 2.4. A four-bar cleaning mechanism in the thresher machine

A four-bar mechanism has been designed for cleaning mechanism (Madan Lal, 2012; Test Code IS:6284, 1985). It consists of four links namely; offset pulley driven by the belt as link 1, connecting rod as link 2, sieves carrier as link 3, and frame of thresher as link 4 as shown in Fig.2.4. Offset pulley converts the rotary motion into oscillation motion through the connecting rod. Further, connecting rod is rigidly connected to the sieves carrier defined as the rocker that separates the grains from the husk by its oscillations.

But, these oscillations in the mechanism develops the forces and moments on the frame of the machine defined as shaking forces and moments. These forces and moments increase the vibration, driving torque, fatigues, etc. in the mechanism. Thus, the balancing of these mechanisms become a challenging task in the existing thresher machine. However, various methods have been developed for the balancing of the mechanisms explained in the next section.

2.2. Methods for balancing of mechanism

The cleaning mechanism is balanced to improve the dynamic performance of thresher machine by balancing of the shaking force and shaking moment. Recently, the complete balancing of shaking forces and shaking moments is the challenging task. Many techniques have been developed to balance the shaking forces and moments using various principles.

The shaking force can be wholly balanced by making the total mass center of the mechanism as stationary (Chaudhary H, Saha, 2009). Generally, the redistribution of moving link masses and counterweights (Fig.2.5) are applied to balance the shaking force. Different methods like the method of principle vector (V.A.Shchepetil'nikov, 1968), method of linearly independent vectors (Berkof and Lowen, 1969), the counterweights (Walker and Oldham, 1978), ordinary vector algebra based method (Kochev, 1987), etc. have been used to trace and make the total mass center of the mechanism stationary. But, complete force balancing procedure increases driving torque and shaking moment in the mechanism (Lowen et al., 1974). Hence, only complete force balancing is not effective. Therefore, the balancing of the shaking moment balance is also required.

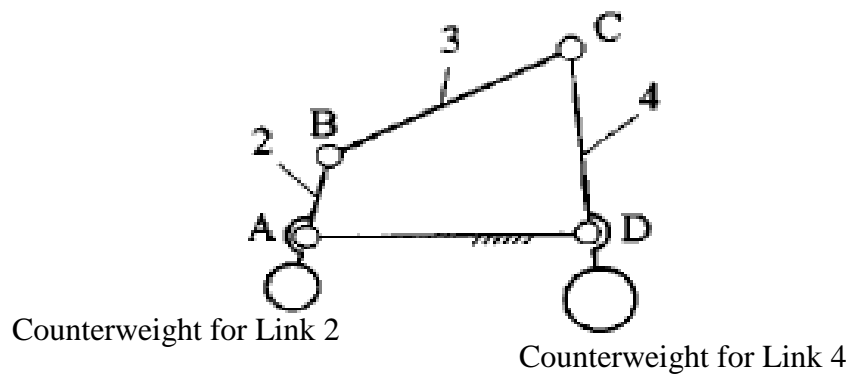


Fig. 2.5. Application of counterweights for complete force balancing (Berkof and Lowen, 1969)

In order to balance the mechanism, the shaking moment can be balanced completely by eliminating the angular momentum of the moving links along with the complete force balance. The total angular momentum is not eliminated by adding counterweights and link mass distribution (Kochev, 2000). Therefore, the moment can be wholly balanced using a cam-actuated oscillating counterweight (Kamenskii, 1968), physical pendulum (Berkof, 1973), moment balancing idler loops (Bagci, 1982), inertia counterweights (Tricamo and Lowen, 1983a, 1983b), geared counterweights (Esat and

Bahai, 1999; Ye and Smith, 1994), and a duplicate mechanism (Arakelian and Smith, 1999).

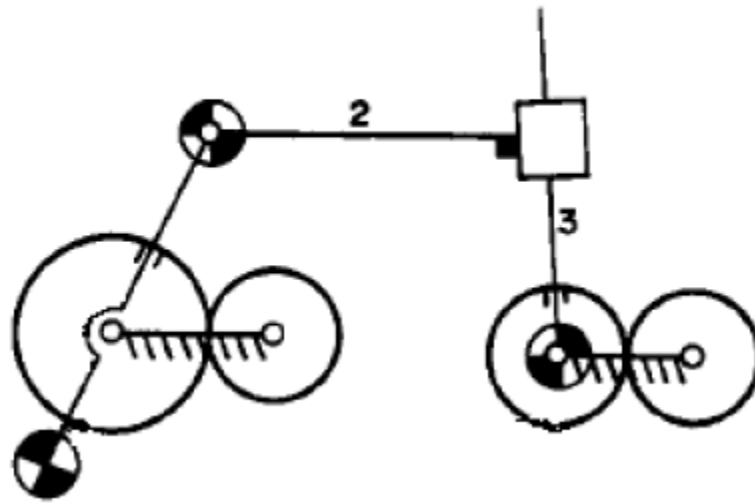


Fig. 2.6. Complete shaking force and shaking moment balancing using mass redistribution and inertia counterweights (Feng, 1990)

The complete shaking force and shaking moment balancing can be achieved by the extended method of linearly independent vectors developed by researchers (Elliott and Tesar, 1977). Moreover, the shaking force and moment have been balanced using mass redistribution and addition of the geared inertia counterweights (Feng, 1990) as shown in Fig.2.6. The shaking force and shaking moment have been completely balanced using an analytical method. In this method, the necessary and sufficient conditions for balancing are derived using computer algebra (Gosselin et al., 2009). It is observed that these methods enhance the weight, cost, and complexity in the mechanisms.

Besides the complete balancing of shaking force and shaking moment, the optimization of each moving link's inertial properties is an alternate way to reduce the shaking force and the shaking moment. Generally, the dynamic quintiles like driving torque, shaking forces, shaking moments, etc. depend on the inertial properties of moving link (Chaudhary and Saha, 2007a). An optimization method has been proposed to minimize the root mean square (RMS) value of shaking moment in a fully force-balanced four-bar mechanism (Lowen and Berkof, 1971). This method further is extended by restricting the limit of the link parameters (Carson and Stephens, 1978). The RMS values of the shaking moment have been minimized within the physical limits of the link parameters (Haines, 1981). The shaking moment has been partially balanced using the counterweight (Arakelian and Dahan, 2001). However, these optimization

methods have been applied to a fully force-balanced four-bar mechanism and consider only the shaking moment as a single objective.

Further, researchers have been made efforts on the multi-objective optimization problem to balance the mechanisms. The different approaches are available to convert the multi-objective problem into a single objective problem like weighted sum method, lexicographic method, weighted global criterion method, physical programming, weighted min-max method, exponentially weighted criterion, weighted product method, goal programming methods and bounded objective function method (Marler and Arora, 2004). Generally, the weighted sum method is more suitable in which proper weighting factors are assigned to all the objectives. A set of optimal solutions defined as Pareto-front has been obtained by different combinations of the weight factors to objectives (Marler and Arora, 2010). Then optimization problem may be solved using conventional and evolutionary optimization algorithms. There are some conventional optimization techniques outlined here, i.e., conventional optimization techniques based on Lagrangian approach have been applied to solve the optimization problem formulated for the combined balancing of shaking force, shaking moment, and torque using two-point mass model (Lee and Cheng, 1984; Pennestri and Qi, 1991).

Further, an optimization problem with minimization of the shaking force and shaking moment is formulated to find the optimal mass distribution of each moving link using three equimomental point mass system (Chaudhary and Saha, 2007b). Point-mass parameters and the weighted sum of the RMS values of the normalized shaking force and shaking moment are considered as design variables and objective function, respectively (Chaudhary and Saha, 2007b). Optimal balancing of combined shaking force and moment is achieved using the counterweight parameters determined by a convex optimization technique (Demeulenaere et al., 2009). A method for a planar mechanism has been proposed to examine the sensitivity of the shaking force and the shaking moment with respect to design variables (Chaudhary and Chaudhary, 2015a). The shaking force, shaking moment, the ratio of shaking moment to mass center distance of links, and inertial parameters of the links are considered as objective functions and design variables, respectively (Li, 1998).

But, these convention optimization techniques require an initial start point to find the optimum solution and give the local solution near to start point.

Some of the evolutionary optimization algorithms, i.e., Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) have been applied to solve the multi-

objective optimization problem based on the inertia counterweights and the physical pendulum approaches in order to balance the mechanism. Moreover, the multi-objective optimization problem is formulated to balance the planar four-bar mechanism and solved using the GA. The kinematic and dynamic parameters of the mechanism are considered as the design variables. Further, the optimization problem has been formulated by considering the shaking force and the shaking moment as multi-objective functions using the equimomental system of point-masses for each moving link. This multi-objective optimization problem is converted into a single objective optimization problem using the weight factors. Point-mass parameters are treated as the design variable. Then, the optimization problem is solved using the Teaching learning-based optimization (TLBO) algorithm (Chaudhary and Chaudhary, 2016) and GA (Chaudhary and Chaudhary, 2014) under the appropriate design constraints.

But, these evolutionary optimization algorithms use the priori approach and increase the computational efforts due to the different combination of the weighting factor for each objective function. Therefore, a posteriori approach based multi-objective optimization algorithm is required to develop for balancing problem of the mechanism. Besides the balancing of the cleaning mechanism, balancing methods for the threshing drum of the thresher machine are explored.

2.3. Methods for balancing of threshing drum

The balancing of the threshing drum plays an essential role in the dynamic performance improvement of the thresher machine. However, the threshing drum is balanced similar manner as the rigid rotor due to its low speed of rotation. Various methods (Fig.2.7) can be applied to balance the threshing drum. These methods with limitations are explored here.

Off-line balancing methods like vector method, four run method, and influence coefficient method have been used to calculate the correction masses for balancing. In the vector method, the correction balance masses are measured from the graph plotted between amplitude and phase. While the four-run method requires amplitude only and not phase angles. The correction masses are determined by measuring the amplitudes from the graph (Foiles et al., 1998). Generally, the influence coefficient method uses two approaches. Two-plane balance problem is considered in the first approach. Correction masses at two balance planes are calculated using influence coefficients which determined from the trial weights at two planes. The second approach considers

two single balance plane problems defined as a static-couple method. In this approach, static and couple influence coefficients are calculated to predict the correction masses at the two-plane.

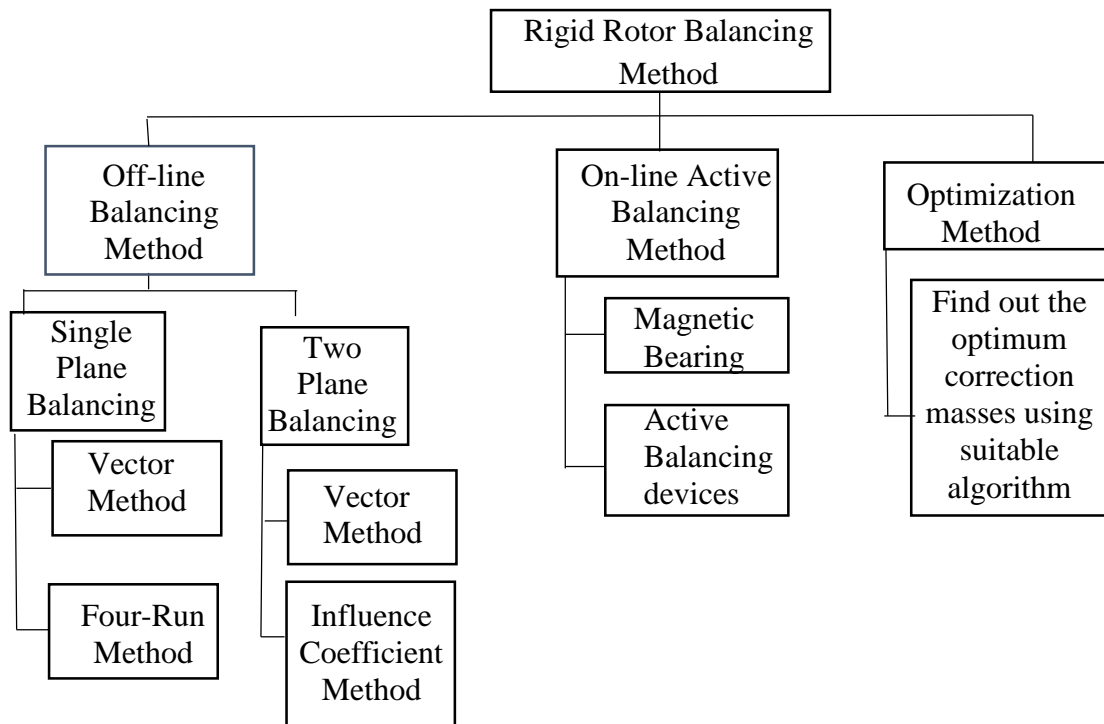


Fig. 2.7. Classification of balancing methods (Zhou and Shi, 2001)

Moreover, the two-plane balancing method without phase measurement is the extension of the four-run method and used to balance the rigid rotor (Everett, 1987). Further, multi-plane influence the coefficient experiment method (Kang et al., 2008) and modal-balancing method have been developed to balance the rotor (Morton, 1985). In both methods, the numbers of trial weights are used to find the vibration responses in different correction plane. Moreover, the unified balancing method is the combination of the modal method and the influence coefficient balancing method and used to find a better result with fewer runs (Darlow, 1989). But, off-line methods of balancing is required more time for calculation of correction masses and cannot change the situation of unbalance during operation. Further, the application of modal balancing is limited due to the known running mode shape.

To overcome these difficulties; researchers focused on active or online balancing methods. It can be classified as magnetic force balancing and mass redistribution balancing. In mass redistribution balancing, mass redistribution devices are used to reduce the unbalance response of the rotor during operation (Shiyu Zhou and Jianjun

Shi, 2001; Van de Vegte, 1981) while in magnetic force balancing, electromagnetic force has been applied to compensate rotor unbalance using electromagnetic bearings directly (Knospe et al., 1996). Further, a ball bearing, also known as a self-compensating balancing device, is used to balance the rigid rotors and provides the numerical solutions for the dynamic imbalance (Rodrigues et al., 2008). But, active or online balancing methods increase the cost and complexity of the system.

Besides experimental and theoretical works based on different balancing methods, the researchers focused on the optimization procedure of balancing. The weighted sums of the squares of the residual vibration have been minimized using the least squares optimization (Goodman, 1964). Moreover, the influence coefficient method is used to estimate the unbalance of rigid rotors by measuring the optimum vibration amplitude on two-balance planes (Everett, 1997). Further, an optimization problem has been formulated for the rotor using the modal balancing method without test runs. This problem is solved using a genetic algorithm to evaluate the balancing masses (Xu et al., 2000). The rigid rotor is balanced using the one-plane balancing procedure. In this method, the imbalance of the system is estimated during the acceleration period.

Further, an ordinary recursive least-squares estimation method has been applied to solve the problem (Zhou and Shi, 2002). However, an optimal procedure is developed for the flexible rotor. This rotor has been balanced using multi-balance planes, and optimum results are validated experimentally (Dyer et al., 2002). Holospectrum and genetic techniques are presented to balance the rotor systems (Liu and Qu, 2008). However, the optimization problem of the rigid rotor is solved by the GA algorithm. Optimal discrete masses at a corresponding angular position are applied for balancing of the rotor (Messenger and Pyrz, 2013). Unfortunately, the optimization techniques used in the balancing are based on only continuous variables and give continuous solutions of the balance masses and corresponding angular positions for the balance planes. These solutions are converted into the nearest discrete solution due to a limited set of available standardized mass values. Thus these solutions induce a dynamic effect. This dynamic effect can be decreased by recalculating of the correction masses. But, this iterative process increases the balancing cost and time. Therefore, optimization algorithms are explored which handle all types of the variables, i.e., continuous, discrete, and integer known as mixed variables.

2.4. Mixed variables optimization algorithms

Various optimization problems deal with the integer, discrete, and continuous variables, known as mixed-variable optimization problems, although many practical design problems consider only discrete and integer variables.

Table 2.1. Algorithms proposed to solve mixed variable optimization variables

S.No.	Algorithms	Type of algorithm	References
1.	Sequential linear programming	Classical	Loh and Papalambros, 1991
2.	Branch and bound methods	Classical	Borchers and Mitchell, 1994; Leyffer, 2001; Sandgren, 1990
3.	A penalty function based algorithm	Classical	FU et al., 1991; Shin et al., 1990
4.	Lagrangian relaxation	Classical	Geoffrion, 1974; Jeet and Kutanoglu, 2007
5.	Integer programming	Classical	Arora, 2000
6.	Differential Evolution	Natural Inspired or Evolutionary	Lampinen and Zelinka, 1999
7.	Evolutionary programming	Natural Inspired or Evolutionary	Cao, 2000
8.	Evolutionary algorithm	Natural Inspired or Evolutionary	Deb, 1997
9.	Simulated annealing (SA)	Natural Inspired or Evolutionary	Zhang and Wang, 1993
10.	Genetic algorithms based on mixed variables	Natural Inspired or Evolutionary	Wu and Chow, 1995
11.	Genetic algorithms based on discrete variables	Natural Inspired or Evolutionary	Rajeev and Krishnamoorthy, 1992
12.	Particle swarm optimization	Natural Inspired or Evolutionary	Guo et al., 2004; He et al., 2004
13.	Ant colony optimization	Natural Inspired or Evolutionary	Camp and Bichon, 2004
14.	Artificial bee colony	Natural Inspired or Evolutionary	Sonmez, 2011

Such as structural design, the number of bolts for a connection, balancing of the rotor using a set of the balance masses on each plane and standard diametric pitch of the gear are some examples of discrete variable optimization problems due to a predefined set of standard values. Integer variables are often used for identical elements in engineering design problem such as the number of the teeth of a gear (Beck et al., 1995).

Most of researchers have been focused on continuous variable optimization algorithms, where optimum values of design variables lie within their bounds. But, these algorithms are not sufficient for practical design problems due to their continuous behavior. Therefore, in recent years, the mixed variable optimization methods are developed for these practical design optimization problems. However, mixed-variable optimization problems can be solved by two classes of optimization techniques, like classical and natural-inspired optimization algorithms as presented in Table 2.1.

Classical algorithms, such as sequential linear programming (Loh and Papalambros, 1991), Branch and bound methods (BBM) (Borchers and Mitchell, 1994; Leyffer, 2001; Sandgren, 1990), a penalty function approach (FU et al., 1991; Shin et al., 1990), Lagrangian relaxation (Geoffrion, 1974; Jeet and Kutanoglu, 2007), rounding-off techniques based on continuous variables, cutting plane techniques and zero-one variable techniques (integer programming) (Arora, 2000) have been applied to mixed variable optimization problems in order to find out the optimum design variables.

But, these methods include more computational cost, low efficiency, and complexity due to the determination of derivatives and the Hessian matrix of the objective function (Arora et al., 1994). Moreover, most of these algorithms work on continuous and differentiable objective function. Further, these give an optimal local solution due to converge on optimal solution near to start point (Arora, 2004).

Recently, natural-inspired optimization algorithms are considered as a useful tool for mixed-variable optimization problems. The Differential Evolution (DE) (Lampinen and Zelinka, 1999), Evolutionary programming (EP) (Cao, 2000), Evolutionary algorithms (EA) (Deb, 1997), simulated annealing (SA) (Kripka, 2004; Zhang and Wang, 1993), Genetic algorithms (GAs) based on mixed variables (Coello et al., 2001; Rao and Xiong, 2005; Wu and Chow, 1995), and discrete variables (Rajeev and Krishnamoorthy, 1992), Particle swarm optimization (PSO) algorithms applied to mixed variables (Guo et al., 2004; He et al., 2004; Kitayama et al., 2006; Nema et al., 2008; Sun et al., 2011) and discrete variables (Li et al., 2009), ant colony optimization (Camp and Bichon, 2004), artificial bee colony (ABC) (Sonmez, 2011), and Teaching-

learning based optimization (TLBO) (Dede, 2014) are some natural-inspired optimization algorithms. Moreover, these algorithms do not require any calculation of derivatives and the Hessian matrix as in the case of classical optimization algorithms. The non-convex and non-differentiable objective function can also be handled using these algorithms.

But, the performance of these algorithms can be affected due to the requirement of algorithmic parameters for its convergence, i.e., GA needs a crossover, mutation rate; PSO needs inertia factor and social parameters; ABC needs limit value; while, crossover constant and scaling factor are required in DE. The selection of these optimization parameters increases the complexity of algorithms. However, these algorithms cannot guarantee to give global solutions within a definite time. Further, the convergence of these algorithms is slow, thus increases the computational efforts. Contrast to these techniques, TLBO algorithm is not required any algorithmic parameter for its convergence. It converges fast by changing the existing solution into the best solution of iteration. But, the optimization problems have been solved using two phases (teacher phase and learner phase) (Rao et al., 2011; Rao, 2015). It is also applied to only discrete optimization problems.

Thus, there is a requirement to develop a parameter-less nature-inspired optimization algorithm for mixed variable optimization problems and apply it to the balancing problem of the threshing drum. Further, the dynamic performance of the thresher machine can also be improved using the shape synthesis of the flywheel. Thus the shape synthesis methods of the flywheel are explored in the next section.

2.5. Shape synthesis methods of the flywheel

The function of the flywheel in the thresher machine is to smooth the torque fluctuations of the threshing drum by storing and releasing energy. The performance of the flywheel can be by its rotating speed, material density, and geometry. However, high rotating speed develops undesirable centrifugal force and high stress in the machine.

In order to increase the kinetic energy, three large flywheels have been used in the conventional thresher machine (Ahmad et al., 2013). Recently, the manufacturers have replaced three large flywheels into one heavy flywheel because three large flywheels increase the manufacturing cost of the flywheel (Madan Lal, 2012; Prashad and Sharma, 1985; Singh, 1978). Moreover, a material with high density can significantly

increase the flywheel mass and thus increase its energy storing capacity. But, a large mass of the flywheel is restricted by the space of the machine and also transfers the unreasonable large gravity force to the bearings (Jiang et al., 2016).

Fortunately, the geometry of a flywheel rotor can also influence the storage capacity of kinetic energy for the same flywheel mass. Composite materials (Danfelt et al., 1977; Metwalli et al., 1983; Takahashi et al., 2001; Tzeng et al., 2006) are widely used to achieve long life and high performance in the design of the geometry of a flywheel. But, the cost of composite materials is high compared to metallic materials, and the forming process is also relatively complex. Therefore, metallic materials are generally used to design for the various geometries of flywheels due to its low price and a simple manufacturing process.

Recently, researchers have been focused on shape optimization of the metallic flywheel rotor geometry cross-section. It is an efficient approach to increase the kinetic energy by optimizing the rotor thickness along a radial direction (Jiang and Wu, 2017a). To achieve optimal thickness distribution along the radial direction, the shape optimization problem with objectives of minimizing the volume, maximizing the kinetic energy, and minimizing the stresses has been formulated under suitable design constraints using polynomial expansion. The coefficients of polynomial expansion of thickness are taken as design variables. Further, the formulated problem is solved using conventional optimization technique (Sandgren and Ragsdell, 2016).

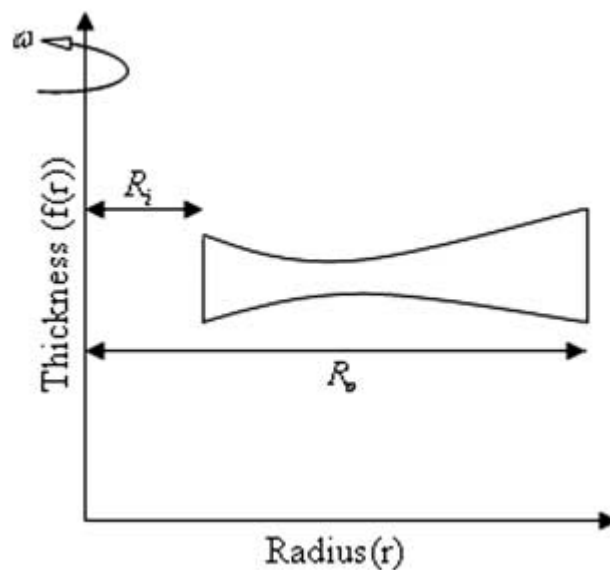


Fig. 2.8. General shape of the flywheel

In another study, coefficients of a Fourier series and Fourier sine series are considered as design variables. The ratio of inertia to volume and kinetic energy are regarded as objective functions with constraints of the maximum allowable stress, maximum thickness, and maximum mass for the optimal shape of the flywheel. These series represent the thickness as a function of the radius. However, the radial and tangential stresses of the flywheel are computed using a two-point boundary value differential equation (Ebrahimi, 1988; Ghotbi and Dhingra, 2012) as shown in Fig. 2.8. However, these curves, i.e., polynomial expansion, Fourier series, and Fourier sine series do not describe the degree of the resulting curve.

The exact optimal shape is determined under the constraints of the geometry, which is derived from arbitrary design parameters, rotational speeds and the strength of disk, using discrete optimization (Berger and Porat, 2017). An injection island genetic algorithm is used to search for the optimal shape of the flywheel with the maximum kinetic energy (Eby et al., 1999). However, the shape of the flywheel is optimized for different kinds of heterogeneous materials using conventional optimization techniques (Huang and Fadel, 2000). The optimum radius of the multi-ring flywheel is obtained by considering the maximization of the stored energy as an objective function (Kyu Ha et al., 2001). The bi-objective optimization problem is formulated using a Bezier curve. Thickness points and volume fraction are considered as the design variable. bi-objectives are converted into a single objective using weighted Tchebycheff method (Huang et al., 2002). Further, a nonlinear optimization problem is formulated to maximize the energy density (stored energy per unit mass) using the parametric geometry modeling method. Then, the downhill simplex method has been used to solve this problem (Jiang et al., 2016).

But, Bezier curves are controlled globally and also increase the computation cost due to the control points. Further, the conventional optimization techniques have been applied for the shape optimization of the flywheel.

In addition to the optimization, finite element analysis has been applied for the optimum shape of the flywheel; a two-dimensional finite-element model is used to find out the optimal thickness distribution for the shape of flywheel along radially centrally bored flywheel under the purpose of reaching an even stress distribution (Kress, 2000). Six different geometries of the flywheel are studied and compared by kinetic energy using finite element analysis (Arslan, 2008). The solid disk profiles of the flywheel are modeled by using cubic splines. The control points of cubic splines are determined to

find out the maximum kinetic energy using finite element analysis (Mahdi, 2010). Optimization approach which acts on the two-dimensional axisymmetric finite element model is used to minimize the mass and maximize the kinetic energy to find out the optimum shape (Pedrolli et al., 2016). However, finite element analysis for the optimal shape of the flywheel increases the computational cost and is also less efficient.

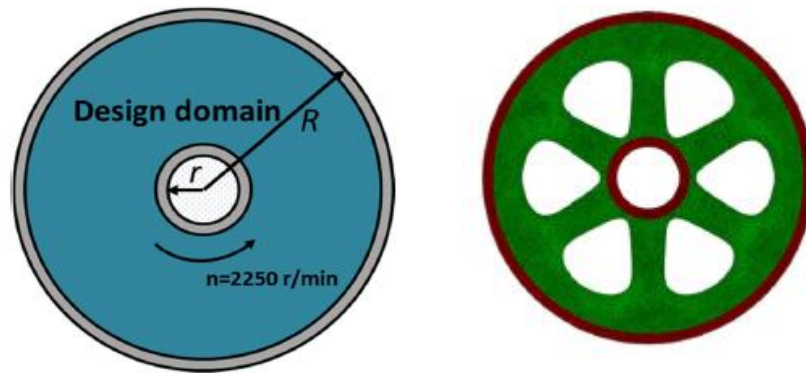


Fig. 2.9. Topology optimization model of the flywheel (Jiang and Wu, 2017b)

Besides the shape optimization of the flywheel, the topology optimization method has been developed to optimize the flywheel rotor geometry layout in order to increase the energy storage capacity as shown in Fig.2.9. In this method, the flywheel rotor is divided into design domain, inner ring, and outer ring. Then, a series of optimized flywheel layouts based on the variable density method is obtained using finite element analysis under different constraints (Jiang and Wu, 2017b). Topology optimization of the rotor for a flywheel energy storage system has been applied to find the best rotor structure using a genetic algorithm (GA). But, the topology optimization method is limited by the initial shape of the flywheel.

2.6. Summary

Various cleaning mechanisms have been designed to the thresher machine as reviewed in Section 2.1. It is observed that the balancing of these mechanisms is a challenging task. Thus, Section 2.2 explores the balancing methods for the mechanisms. Based on the literature review, the complete shaking force balancing is not effective due to the increment of other dynamic quantities in the mechanism. Another approach, i.e., complete shaking force and shaking moment balancing minimizes the other dynamic quantities. But this balancing method increases the complexity and weight of the mechanism. Thus, the optimal balancing procedure has been used to reduce the

shaking force and shaking moment. However, the conventional and evolutionary optimization algorithms can be applied for the balancing of the mechanisms. But, these conventional optimization techniques require an initial start point to find the optimum solution and give the local solution near to start point. In contrast, evolutionary optimization algorithms are based on the priori approach and also increase the computation time due to the combination of the weight factors for each objective function. Therefore, post-priori based optimization algorithm is applied for the balancing of the mechanism and minimizes the computational efforts.

In Section 2.3, the methods used for balancing of the threshing drum are reviewed, and it is observed that the off-line balancing methods increase the computational efforts due to the calculation of correction masses whereas active or online balancing methods increase system's complexity and cost. However, optimization techniques have been applied to overcome these difficulties. But, these optimization techniques are based on continuous variables and give the optimal continuous solutions of the balance masses and corresponding angular positions. These solutions are recalculated for the nearest discrete solution due to a limited set of available standardized mass values. Thus this iterative process increases the balancing cost and time.

Therefore, the algorithms based on mixed variables are explored and reviewed in Section 2.4. However, classical and natural-inspired optimization algorithms can be applied to mixed variable optimization problems. But, classical optimization algorithms increase the computational cost due to Hessians and derivatives calculations and also have less probability of global optimum solution. In contrast, natural-inspired optimization algorithms are affected by the algorithmic control parameters for their convergence. Therefore, a parameter-less nature inspired mixed variable optimization algorithm is required for the balancing of the threshing drum and also reduces the balancing cost and time.

The dynamic performance of the thresher machine can also be improved using shape synthesis of the flywheel and methods used for shape synthesis of the flywheel are reviewed in Section 2.5. It is concluded that the optimization methods for flywheel shape synthesis are based on the conventional approach, and curves used for optimal thickness distribution of the flywheel rotor are restricted by control points, coefficient parameters, the degree of the resulting curve, etc. while finite element analysis and topology optimization approaches require an initial design and also increase the computation efforts.

Optimal balancing of the cleaning mechanism

This chapter proposes the dynamically balanced cleaning mechanism for the thresher machine using the concept of a dynamically equivalent point-mass system. The cleaning mechanism is balanced by optimizing the inertial properties of each link which defined by the dynamic equivalent system of point masses. The shaking force and the shaking moments developed in the cleaning mechanism are derived in term of point mass parameters. Thus, the multi-objective optimization problem with minimization of shaking forces and shaking moments is formulated by considering the point mass parameters as the design variables. The formulated optimization problem is solved using a posteriori approach based algorithm as non-dominated sorting Jaya algorithm (NSJAYA) and a priori approach based algorithms like Jaya algorithm and Genetic algorithm (GA) under suitable design constraints. The optimal Pareto set for the balancing of the mechanism is calculated and outlined.

3.1. Dynamic equations of motion for the rigid body

The i th moving rigid link of the mechanism is modeled as a rigid body in xy plane for kinematic and dynamic analysis as shown in Fig.3.1. oxy and $o_i x_i y_i$ are the global and local coordinate systems, respectively. The link length is the distance between joints o_i and o_{i+1} denoted by r_i .

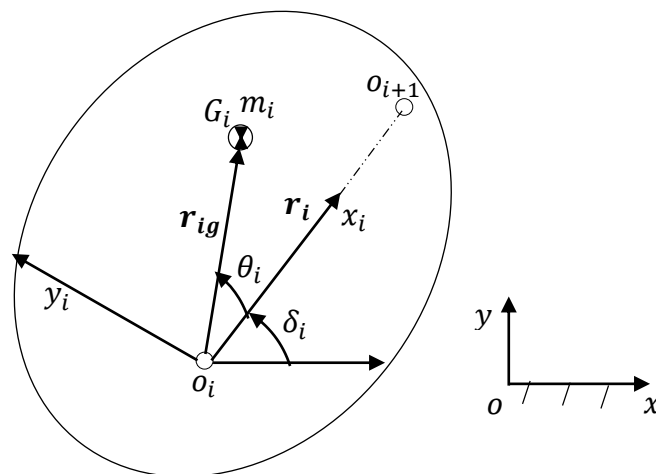


Fig. 3.1. The i th moving link of the mechanism

The mass and moment of inertia of link about o_i are represented as m_i and I_i , respectively. The position of mass center G_i is defined by r_{ig} at an angle θ_i made from

axis $o_i x_i$ fixed to the link. The axis $o_i x_i$ is at an angle δ_i from the axis ox and aligned between joints o_i and o_{i+1} . The bold faces represent the vectors.

The resultant forces at o_i and moment of i th moving rigid link about o_i are determined using Newton-Euler (NE) equations ‘(Chaudhary H, Saha, 2009).’

$$\begin{aligned}
 & \begin{bmatrix} 0 & 0 & 0 \\ -m_i \omega_i r_{ig} \cos(\theta_i + \delta_i) & 0 & 0 \\ -m_i \omega_i r_{ig} \sin(\theta_i + \delta_i) & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{ix} \\ v_{iy} \end{bmatrix} \\
 & + \begin{bmatrix} I_i & -m_i r_{ig} \sin(\theta_i + \delta_i) & m_i r_{ig} \cos(\theta_i + \delta_i) \\ -m_i r_{ig} \sin(\theta_i + \delta_i) & m_i & 0 \\ m_i r_{ig} \cos(\theta_i + \delta_i) & 0 & m_i \end{bmatrix} \begin{bmatrix} \dot{\omega}_i \\ \dot{v}_{ix} \\ \dot{v}_{iy} \end{bmatrix} \\
 & = \begin{bmatrix} M_i \\ f_{ix} \\ f_{iy} \end{bmatrix} \quad (3.1)
 \end{aligned}$$

Where ω_i is the angular velocity, v_{ix} and v_{iy} are the components of the linear velocities. $\dot{\omega}_i$, \dot{v}_{ix} , and \dot{v}_{iy} represent the angular acceleration and the components of the linear accelerations, respectively. The resultant moment about o_i and the components of the resultant force at o_i are represented by M_i , f_{ix} , and f_{iy} , respectively.

3.2. Equations of motion for dynamically equivalent point mass system

The shaking force and the shaking moment in the mechanism are determined using the dynamically equivalent system of point masses for each moving link as shown in Fig.3.2. The dynamically equivalent system can improve the dynamic performance of the mechanism using the mass distribution of the moving links that represent the inertial properties, i.e., mass center position, mass, and inertia of each moving link (Routh, 1905). However, the planar and spatial mechanism can be balanced using the two-point-masses (Wenglarz et al., 1969) and four point-masses (Huang, 1983). Thus, mass distribution can be obtained using the methods of optimization (Sherwood and Hockey, 1969).

The criteria for dynamically equivalent systems is that the center of mass, the moment of inertia, and mass of the original link must be equal to those of point mass system (Chaudhary H, Saha, 2009). The p point masses m_{ij} , make the angles θ_{ij} from the axes $o_i x_i$ located at distances a_{ij} .

The criteria for the equivalent system of point masses for the rigid link is expressed as

$$\sum_i^n \sum_j^p m_{ij} = m_i \quad (3.2)$$

$$\sum_i^n \sum_j^p m_{ij} a_{ij} \cos(\theta_{ij} + \delta_i) = m_i r_{ig} \cos(\theta_i + \delta_i) \quad (3.3)$$

$$\sum_i^n \sum_j^p m_{ij} a_{ij} \sin(\theta_{ij} + \delta_i) = m_i r_{ig} \sin(\theta_i + \delta_i) \quad (3.4)$$

$$\sum_i^n \sum_j^p m_{ij} a_{ij}^2 = I_i \quad (3.5)$$

Where i denote the number of links, and j defines the point masses for the i th link.

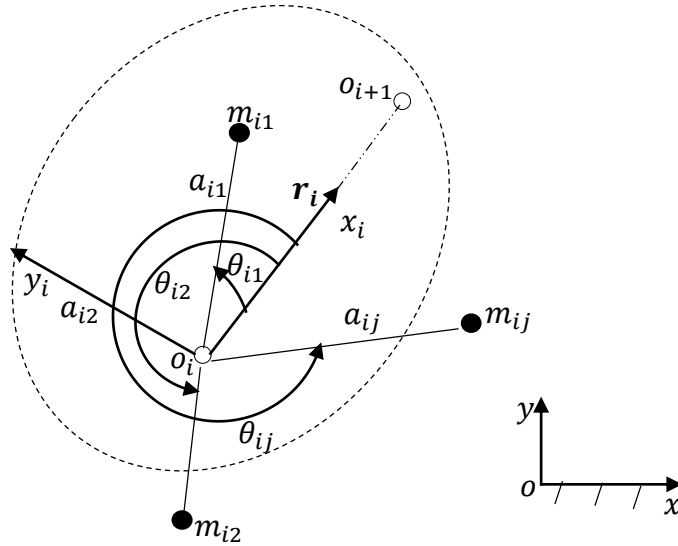


Fig. 3.2. Dynamically equivalent point mass systems for i th moving link of the cleaning mechanism

The Eq. (3.1) can be redefined for the equivalent system of point masses of the i th link using Eqs. (3.2-3.5) as:

$$\begin{aligned}
& \begin{bmatrix} 0 & 0 & 0 \\ -\omega_i \sum_i^n \sum_j^p m_{ij} a_{ij} \sin(\theta_{ij} + \delta_i) & 0 & 0 \\ -\omega_i \sum_i^n \sum_j^p m_{ij} a_{ij} \sin(\theta_{ij} + \delta_i) & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{ix} \\ v_{iy} \end{bmatrix} \\
& + \begin{bmatrix} \sum_i^n \sum_j^p m_{ij} a_{ij}^2 & -\sum_i^n \sum_j^p m_{ij} a_{ij} \sin(\theta_{ij} + \delta_i) & \sum_i^n \sum_j^p m_{ij} a_{ij} \cos(\theta_{ij} + \delta_i) \\ -\sum_i^n \sum_j^p m_{ij} a_{ij} \sin(\theta_{ij} + \delta_i) & \sum_i^n \sum_j^p m_{ij} & 0 \\ \sum_i^n \sum_j^p m_{ij} a_{ij} \cos(\theta_{ij} + \delta_i) & 0 & \sum_i^n \sum_j^p m_{ij} \end{bmatrix} \begin{bmatrix} \dot{\omega}_i \\ \dot{v}_{ix} \\ \dot{v}_{iy} \end{bmatrix} \\
& = \begin{bmatrix} M_i \\ f_{ix} \\ f_{iy} \end{bmatrix}
\end{aligned} \tag{3.6}$$

The parameters, m_{ij} , a_{ij} and θ_{ij} are considered as design variables. There are total $3pn$ parameter where p and n are point masses and the number of moving links, respectively.

3.3. Dynamic analysis of the cleaning mechanism

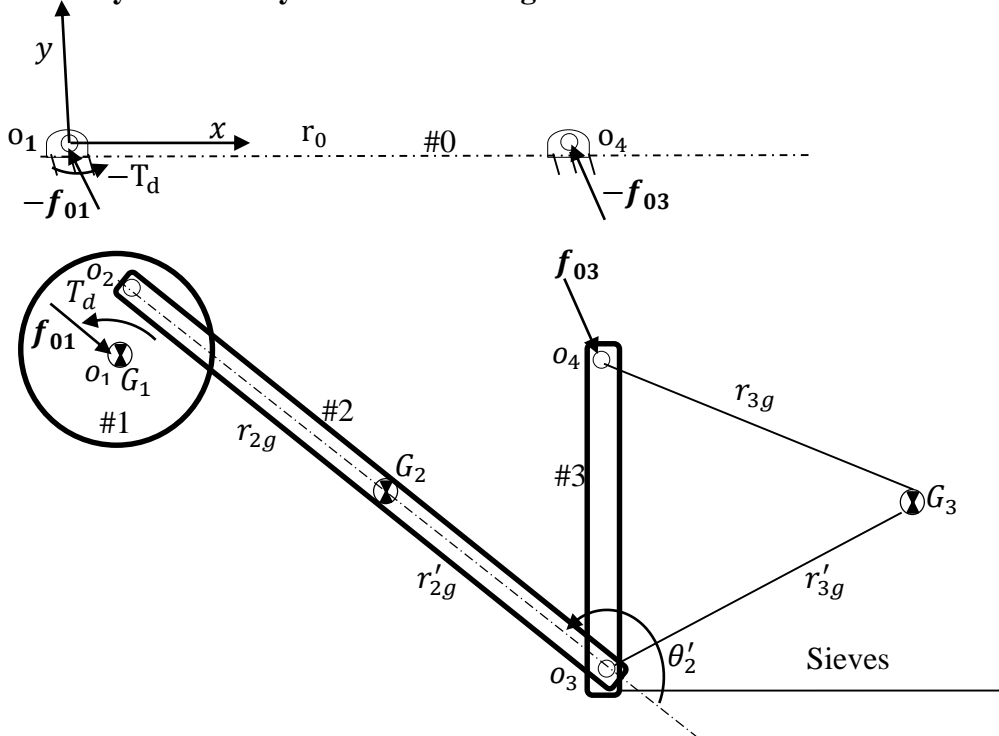


Fig. 3.3. Reaction forces and moments with respect to fixed link in cleaning mechanism

This section describes developed shaking forces and shaking moments in the cleaning mechanism as shown in Fig.3.3. The vector sum of reactions of all inertia

forces is defined as the shaking force. Whereas, the shaking moment is defined as the resultant of inertia couple and moment of inertia forces (Chaudhary H, Saha, 2009). It is considered that only driving torque is acting about the joint o_1 . Thus, the shaking force and the shaking moment transmit to the fixed link, #0 with respect to o_1 , are expressed, respectively, as:

$$\mathbf{f}_{sh} = -(\mathbf{f}_{01} + \mathbf{f}_{03}) \quad (3.7)$$

$$M_{sh} = -T_d - \mathbf{r}_0 \times \mathbf{f}_{03} \quad (3.8)$$

Where vectors \mathbf{f}_{01} and \mathbf{f}_{03} are the reaction forces of fixed link 0 acting on links 1 and 3, respectively, while T_d represents the driving torque about joint 1. \mathbf{r}_0 is the vector from o_1 to o_4 .

As reaction forces, shaking force and shaking moment have different magnitudes and units. To harmonize them, the forces and moments are normalized correspond to the parameters of the reference link as (Chaudhary H, Saha, 2009):

$$\overline{f_{sh}} = \frac{\mathbf{f}_{sh}}{m_1 r_1 \omega_1^2} \quad (3.9)$$

$$\overline{M_{sh}} = \frac{M_{sh}}{m_1 r_1^2 \omega_1^2} \quad (3.10)$$

3.4. Formulation of the optimization problem

A priori and posteriori based optimization problems with minimization of the shaking force and the shaking moment are formulated using a concept of dynamically equivalent point mass system. Each link is represented using the p point mass system. Total $3p$ parameters for each link needs to specify the point masses. The design vector \mathbf{x}_i , for the i th link, is described as

$$\mathbf{x}_i = [m_{i1} \ a_{i1} \ \theta_{i1} \ m_{i2} \ a_{i2} \ \theta_{i2} \ \dots \ m_{ip} \ a_{ip} \ \theta_{ip}]^T \quad (3.11)$$

Although, the design variable vector $9p$, \mathbf{x} for mechanism having 3 moving link is given as

$$\mathbf{x} = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \mathbf{x}_3^T]^T \quad (3.12)$$

Where m_{ij} is the j^{th} point mass for the i^{th} link, a_{ij} and θ_{ij} are the corresponding lengths and angular position of m_{ij}

The priori approach based optimization problem is considered the weighted sum of the RMS values of normalized shaking force, $\overline{f_{sh,rms}}$ and the shaking moment, $\overline{M_{sh,rms}}$ as objective function, while the posteriori approach based optimization

problem is not required any weighting factors and minimizes the RMS values of shaking force, $F_{sh,rms}$ and the shaking moment, $M_{sh,rms}$, simultaneously. The priori and posteriori based optimization problems are stated under the appropriate constraints, respectively, as:

$$\text{Minimize } Z(\mathbf{x}) = w_1 \overline{f_{sh,rms}} + w_2 \overline{M_{sh,rms}} \quad (3.13)$$

$$\left. \begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = F_{sh,rms} \\ \text{Minimize } f_2(\mathbf{x}) = M_{sh,rms} \end{array} \right\} \quad (3.14)$$

Subjected to

$$\left. \begin{array}{l} g_{1i}(\mathbf{x}) = 0.25m_i^o - \sum_j m_{ij} < 0 \\ g_{2i}(\mathbf{x}) = \sum_j m_{ij} - 1.5m_i^o < 0 \\ g_3(\mathbf{x}) = r_{1g} - r_1 < 0 \\ g_4(\mathbf{x}) = r_{2g} - r_2 < 0 \\ g_5(\mathbf{x}) = r_{3g} - 10r_3 < 0 \\ g_6(\mathbf{x}) = 2.5r_3 - r_{3g} < 0 \\ g_{7i}(\mathbf{x}) = I_i - I_i^o < 0 \\ g_8(\mathbf{x}) = \theta_3 - \theta_3^o \leq 0 \\ LB_r \leq x_r \leq UB_r \quad r = 1, \dots, N \end{array} \right\} \text{for } i = 1, 2, 3 \text{ and } j = 1, 2, \dots, p \quad (3.15)$$

Where w_1 and w_2 represent the weighting factors that assigned to objective functions based on relative importance, respectively, generally, these are used in a priori based optimization problem and convert the multiple objective functions into the single objective function (Marler and Arora, 2010). The designer can take any values of these factors like 0, 1, between 0 and 1 for different objectives based on an application. Moreover, the design constraints depend on the permissible values of the parameters of the link. The maximum and minimum masses of the i th link are decided by the strength of links. Similarly, the limiting values of the moment of inertias determine the limits on parameters, a_{ij} . UB_r and LB_r are the upper and lower limits on the r th design variable, the number of design variables is denoted by N , and the superscript “o” indicates the parameters of the original mechanism.

To obtain an optimum solution, a priori and posteriori based constrained optimization problems are converted into an unconstrained problem using penalty function (Singh et al., 2017). A significant penalty value is added to the objective function for each constraint violation. As a result, the objective function proceeds toward an infeasible solution. Hence, the global optimum solution is obtained by satisfying all the constraints using a suitable optimization algorithm. The original

constrained optimization problem is then stated as an unconstrained optimization problem in which the first and second terms describe the objective function and the penalty function, respectively. Finally, a priori based optimization problem is formulated as:

$$\text{Minimize } Z(\mathbf{x}) = w_1 \overline{f_{sh,rms}} + w_2 \overline{M_{sh,rms}} + \sum_{c=1}^{n_c} F_c (P_f)^c \quad (3.16)$$

Where n_c is the number of constraints, while P_f is the penalty value of the order of 10^6 which assigned to objective function if the constraints are not satisfied. Boolean Function F_c is defined as

$$F_c = \begin{cases} 1 & \text{if } g_c(\mathbf{x}) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

While a posteriori based optimization problem is formulated as

$$\text{Minimize } f_1(\mathbf{x}) = f_{sh,rms} + P_f \times \sum_{c=1}^{n_c} \max(0, g_c(\mathbf{x})) \quad (3.17)$$

$$\text{Minimize } f_2(\mathbf{x}) = M_{sh,rms} + P_f \times \sum_{c=1}^{n_c} \max(0, g_c(\mathbf{x})) \quad (3.18)$$

$$\text{Minimize } Z(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})]^T \quad (3.19)$$

$$LB_r \leq x_r \leq UB_r \quad r = 1, \dots, N \quad (3.20)$$

3.5. Optimization algorithm

The formulated problem described in the previous section is solved by a priori and posteriori based algorithms. The optimization problem defined in Eq. (3.16) is solved using a priori based algorithms like GA and Jaya algorithm. However, the GA algorithm requires the algorithmic parameter for its convergence.

Further, these algorithms converge to a global solution, but there is no guarantee that the solution is optimal (Singh et al., 2017). While Jaya algorithm does not require any algorithmic parameters and, is also easier to implement (Rao, 2019). It converges fast to the optimal global solution by updating the best and worse solutions. It is more computationally efficient than GA. The detailed procedure of this algorithm is explained in Fig.3.4.

Whereas formulated problem defined in Eq.(3.19) can be solved by some posteriori based algorithms like GA(Ani et al., 2013; Coello, 2000), MPSO (Feng et al., 2012), PSO (Parsopoulos and Vrahatis, 2002), NSGA-II (Deb et al., 2002), etc. Although these algorithms give a global solution, it is no assurance that the solution is optimal (Singh

et al., 2017). Moreover, these algorithms need algorithmic control parameters which affect their performance (Rao, 2019). However, NSJAYA algorithm is also the posteriori based algorithm. It does not need any algorithmic control parameters. The solutions obtained using this algorithm are improved in a similar way as in Jaya algorithm which proposed by Rao in 2016.

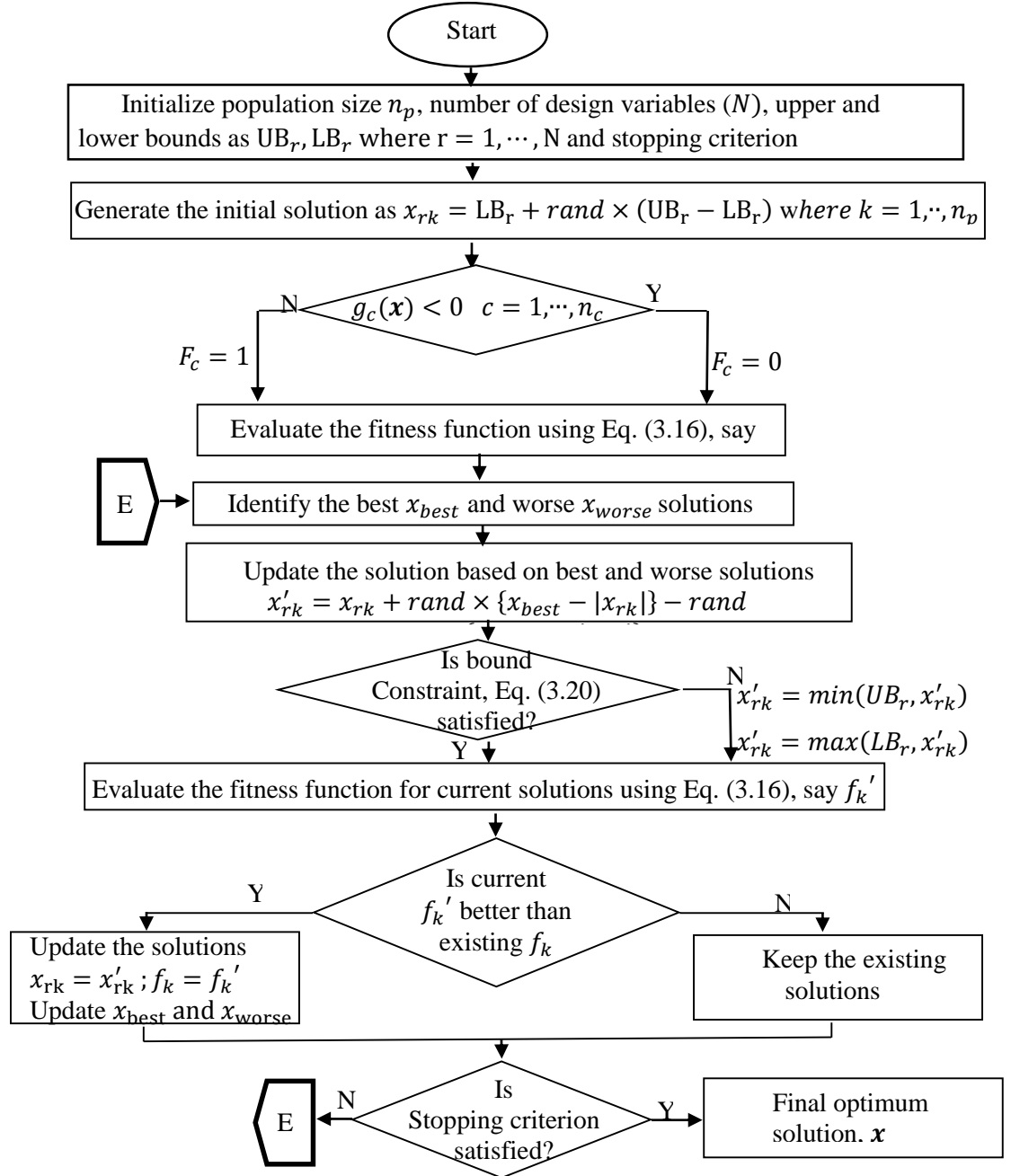


Fig. 3.4. A flow chart of Jaya algorithm for balancing of the cleaning mechanism

Moreover, NSJAYA algorithm handles the multi-objectives using crowding distance computation mechanism and non-dominated sorting approach given in ref.

(Rao et al., 2018). Although the determination of the best solution in multi-objectives is difficult due to the multiple opposing objectives.

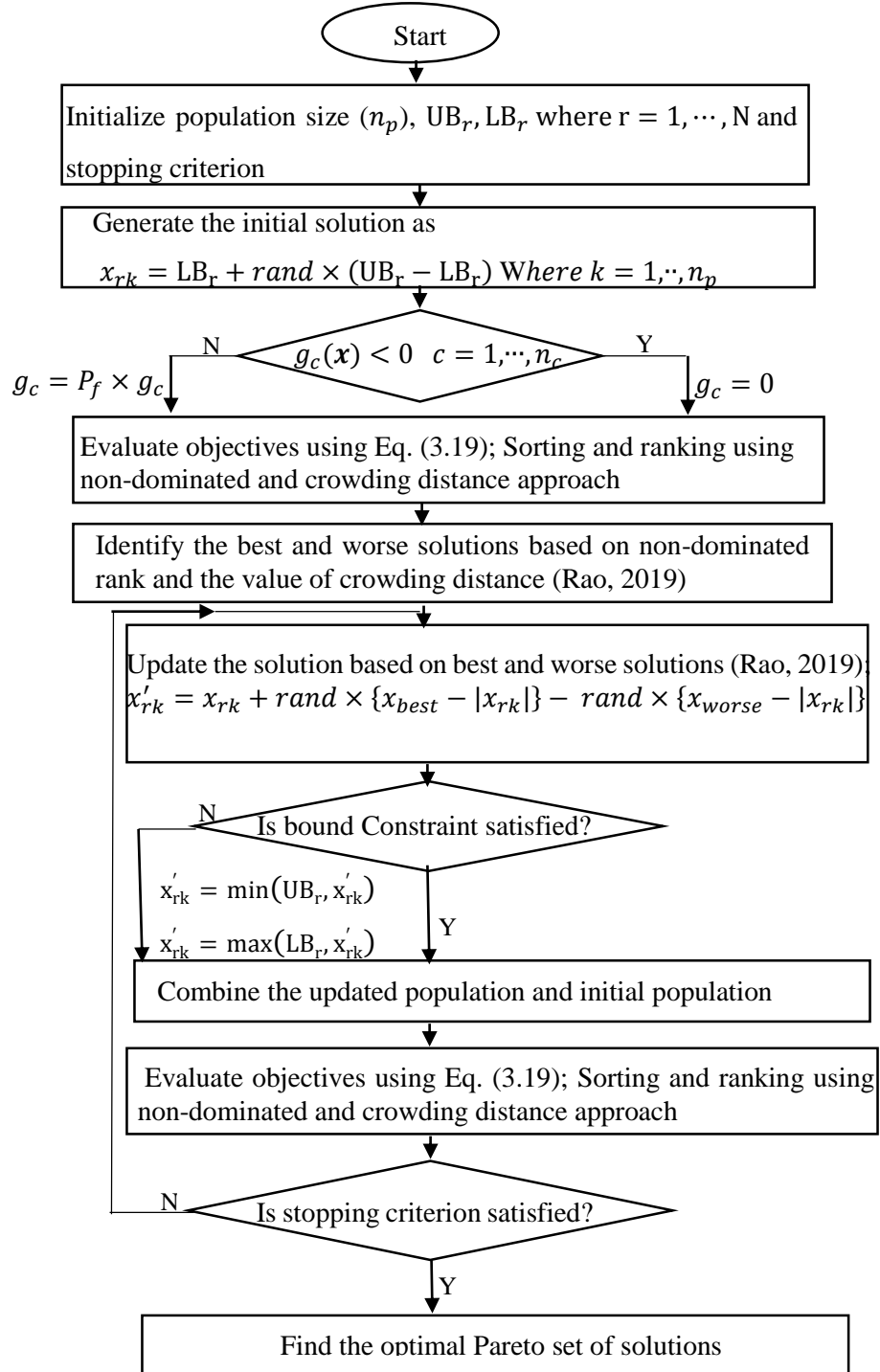


Fig. 3.5. A flow chart for the non-dominated sorting Jaya algorithm

Therefore, NSJAYA algorithm is developed to find the best solutions for the balancing of the cleaning mechanism. In this algorithm, these solutions are selected by comparing the rank of solutions using the crowding distance value and the non-dominance concept. In starting, initial solutions “p,” are randomly created. Then, initial

solutions are arranged in ascending order with rank using the non-dominance approach (Rao, 2019). In this approach, non-dominance solutions are identified based on the criteria such as a solution x^* is considered a non-dominant than other solution x if and only if solution x^* is better than solution x with respect to all the objectives and the solution x^* is strictly better than solution x in at least one objective. Thus these non-dominance solutions are arranged in ascending order with ranks. The higher rank (rank=1) and lower rank solution are chosen the as best and worse solution, respectively. If solutions have the same rank, then crowding distance value is calculated for these solutions. Generally, crowding distance (CD) is a measure of "density of solutions surrounding a particular solution in the population" and calculated as:

$$CD_i = CD_i + \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}} \quad (3.21)$$

The crowding distance ($CD_1 = CD_l = \infty$) is assigned to the first and last solution of the non-dominant solution. Where $i = 2, \dots, l - 1$ and $j = 1 \dots, m$. m represents the number of objective functions.

The solution with the low value of crowding distance is considered as the worse solution and vice versa. The solutions are modified according to the general equation of Jaya algorithm (Rao et al., 2018). Then, the population based on updated solutions is merged with the initial solutions, and $2p$ populations are generated. Combined solutions are sorted and ranked again, and the crowding distance value of each solution is calculated (Rao, 2019). The best solutions are chosen according to the new rank and value of crowding distance. The solution of the higher rank and high value of crowding distance is considered as best compared to other solution (Rao et al., 2017). The dominance concept for finding the rank of solutions, the computation of crowding distance, and operator of crowding comparison are detailed in refs. (Deb et al., 2002; Rao et al., 2018, 2017).

The termination criteria of this algorithm are described by function evaluations and the number of iterations. The product of the number of iterations and initial populations is known as the number of function evaluations (Rao, 2019). So, function evaluations are not affected by design variables, but the computational time of the algorithm can be enhanced. The flowchart of this algorithm is presented in Fig.3.5. Further, this algorithm decreases the computational time than the other multi-objective optimization techniques and a priori base optimization algorithm like GA and Jaya algorithm.

Moreover, the first time it is applied to the dynamic balancing of the planar mechanism in this study.

3.6. Results and discussions

In this section, the proposed optimization approach is applied for the balancing of the cleaning mechanism used thresher machine (Singh, 1978; Varshney, 2004). In this study, each link of the mechanism is modelled using a three-point mass system for better mass distribution. However, more than three-point mass system can be applied, but it will enhance the evaluation time of the objective function and constraints. Moreover, The five parameters out of nine parameters of each link for the three-point mass model are assigned to reduce the dimension of the problem (Chaudhary H, Saha, 2009), as:

$$\theta_{i1} = 0; \theta_{i2} = 120^\circ; \theta_{i3} = 240^\circ \text{ and } a_{i1} = a_{i2} = a_{i3}$$

Hence, m_{i1}, m_{i2}, m_{i3} , and a_{i1} , other parameters of each link are taken as the design variables. Hence there are 12 design variables and, limits on these design variables are considered for both priori and posteriori based optimization problems, as:

$$m_{i1} \in [0.05, 5]; m_{i2} \in [0.05, 1.0]; m_{i3} \in [1.0, 15]; a_{i1} \in [0.25a_{i1}^0, 2a_{i1}^0], i = 1, 2, 3$$

In the cleaning mechanism, the input motion is given to pulley as a link 1 by PTO shaft of the tractor using V-belt that rotates at the constant speed of 250 rpm (26.18 rad/s). The link parameters of the original cleaning mechanism are presented in Table 3.1 (Singh, 1978; Varshney, 2004).

Table 3.1. Link parameters of the cleaning mechanism

Link i	Link length	Total link mass	Moment of inertia	Mass center location	
	$r_i (m)$	$m_i (kg)$	$I_i^G (kg \cdot m^2)$	$r_{ig} (m)$	$\theta_i (deg)$
#1	0.020	0.6288	5.2975e-04	0.0238	0
#2	0.2397	0.3513	1.76e-03	0.11984	0
#3	0.1393	24.075	1.973e+01	0.7782	84.143
#4	0.2169	-	-	-	-

Three different cases are applied to balance the mechanism in the priori based optimization problem described in Eq. (3.16). In case 1 and 2, the total force balancing is considered, and the values of weighting factors are taken as $w_1 = 1$ and $w_2 = 0.0$.

In case 1, links 1 and 3 are chosen for the point mass system, while the parameters of link 2 are considered as its original values. The point masses of links, #1 and #3 are considered as the design variables. Then, optimization problem are solved under the constraints $g_{1i}(x)$ and $g_{2i}(x)$ defined in (Eq. 3.12). The optimum results are compared

with full analytical force balancing conditions proposed by Berkof and Lowen (R. S. Berkof and G. G. Lowen, 1969) given as:

$$\left. \begin{aligned} m_1 r_{1g} &= m_2 r'_{2g} \frac{r_1}{r_2} \\ \theta_1 &= \theta'_2 \end{aligned} \right\} \quad (3.22)$$

$$\left. \begin{aligned} m_3 r_{3g} &= m_2 r_{2g} \frac{r_3}{r_2} \\ \theta_3 &= \theta_2 + \pi \end{aligned} \right\} \quad (3.23)$$

All parameters are shown in Figs. 3.1 and 3.3.

While all three moving links are modelled using the point-mass system in case 2. In case 3, the combined shaking force and shaking moment balancing is considered. Thus $w_1 = 0.5$ and $w_2 = 0.5$ are considered in this case. All three moving links are considered for the point-mass system in this case.

Table 3.2. Optimized values of link parameters for the balanced mechanism for all cases

Cases	Link i	Total link mass	Mass center location		Mass distance product
		$m_i(kg)$	$r_{ig} (m)$	$\theta_i (deg)$	$I_i^G (kg-m^2)$
Case 1; ($w_1 = 1, w_2 = 0$) Link 1 and 3 are considered	#1	0.2851	0.0104	180	0.000458
	#3	28.2743	0.0010	180	57.0541
Case 2; ($w_1 = 1, w_2 = 0$) All links are considered	#1	0.6903	0.0307	256.65	7.3118e-04
	#2	0.1505	0.1672	358.59	0.0024
	#3	6.0243	0.5984	56.0169	7.1975
Case 3; ($w_1 = 0.5, w_2 = 0.5$) All links are considered	#1	0.4202	0.0068	120.00	1.3005e-05
	#2	0.2501	0.1459	0.0114	9.0812e-04
	#3	6.0455	0.3763	84.1344	5.1685

The formulated problem for all cases is solved using Jaya algorithm and GA. These algorithms are coded in MATLAB 2014. A laptop having 3GB RAM and processor with 2.20 GHz is used to run these algorithms. The number of iterations and population size are chosen as 300 and 100 for GA, respectively, while Jaya algorithm takes the population size of 100 and the number of iteration as 280. Both algorithms take the 10 independent runs to find out the best objective function value with respect to the design variables corresponding to the best run. The computational efficiency of the algorithms for all three cases is shown in Fig.3.6. GA gives the best objective function value at 30000 function evaluations, while Jaya algorithm reaches to best objective function value at 28000 function evaluations. The algorithm is considered more efficient if it takes fewer function evaluations to find the optimum solution (Singh and Chaudhary,

2019). Hence, Jaya algorithm is comparatively more efficient and takes 93.6% function evaluations compared to GA.

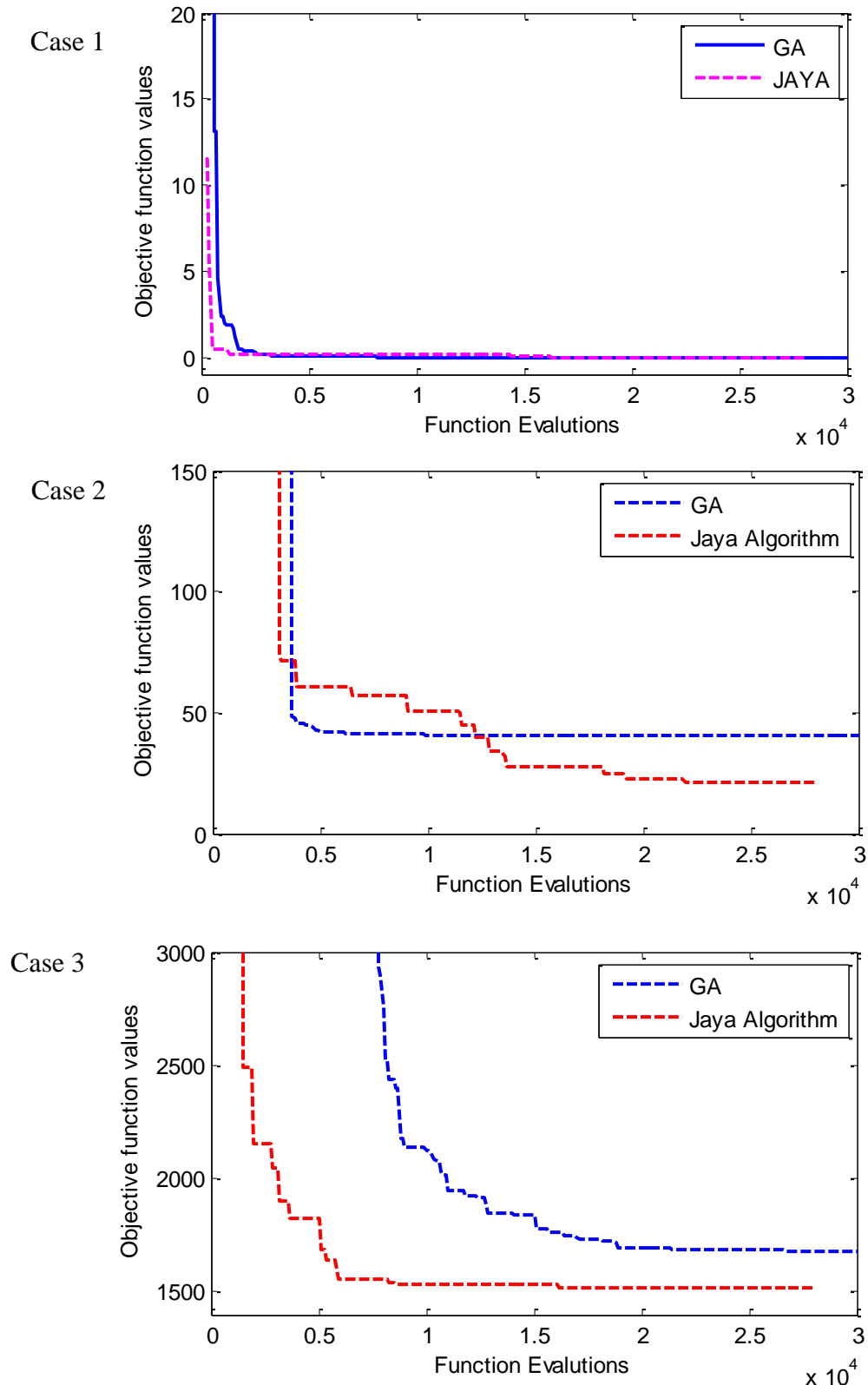


Fig. 3.6. Convergence of the best values of the objective function for GA and Jaya algorithm in all cases

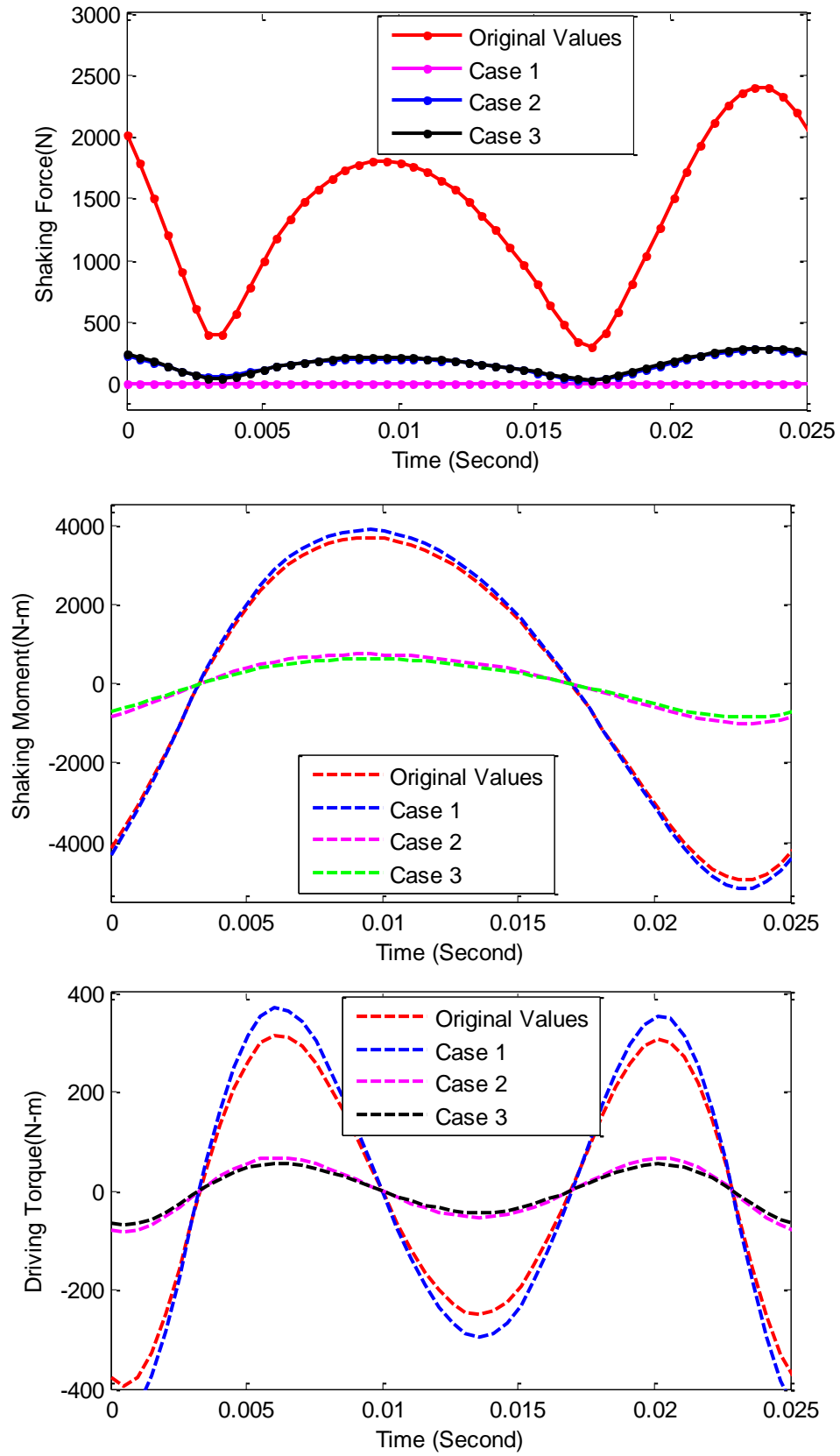


Fig. 3.7. Variation in shaking force, shaking moment and driving torque with time for one cycle of the crank

The optimum solutions obtained by Jaya algorithm are used to determine inertial properties for the balanced mechanism using Eqs (3.2-3.5); those are presented in Table 3.2. The RMS values of the shaking force, shaking moment, and input torque for each case are compared with those of the original mechanism as presented in Table 3.3. Fig.3.7 shows the variation in shaking force, shaking moment, and input torque for one cycle of the crank. The optimized results of the balanced mechanism for case 1 satisfy the analytical conditions given by Eqs. (3.22-3.23). But, the total force balancing (case 1) enhances the RMS values of driving torque and shaking moment up to 16.48% and 4.53%, respectively. The best solutions achieved in case 3 decrease the RMS values of driving torque, shaking force, and shaking moment up to 82.48%, 87.76%, and 83.037%, respectively. Hence, combined shaking force and shaking moment balancing is more effective compared to the total force balancing.

Table 3.3. The RMS values of the balanced mechanism for all cases

Cases	Type of balancing	Input Torque (N-m)	Shaking force (N)	Shaking moment (N-m)
	Original Mechanism	226.96	1503.00	3067.80
Case 1	Force balance	264.36 (+16.48%)	0.00078 (-99%)	3207.6 (+4.53%)
Case 2	Force balance	48.4011 (-78.67%)	176.52 (-88.25%)	616.645 (-79.92%)
Case 3	Force and moment balance	39.7603 (-82.48%)	184.0337 (-87.76%)	520.39 (-83.037%)

A priori based optimization algorithm requires the combination of the weighting factors to objective functions. This process increases the computational time of the algorithm. Moreover, the user should be aware of the importance of each objective in advance that is a difficult task due to the uncertain scenario. Therefore, a posteriori based optimization algorithm NSJAYA is applied to balance the cleaning mechanism of the thresher machine. A posteriori based optimization problem defined in Eq. (3.19) is solved using this algorithm. It takes 100 population size and the number of iterations equal to 200 for finding the non-dominated solutions (Pareto front). The non-dominated solutions are shown in Fig.3.8 and reported in Table 3.4. Two solutions based on the importance of objective function are considered from Table 3.4, i.e., solution 1 gives a maximum priority to the minimization of the shaking moment, while solution 2 gives a

maximum priority to the minimization of the shaking force. The RMS values of shaking force, shaking moment and input-torque based on solutions 1 and 2 are likened with those of original mechanism and the RMS values obtained by the priori based algorithm (GA and Jaya) for combined balancing (case 3), as presented in Table 3.5. Table 3.5 shows that the dynamics quantities of the cleaning mechanism are significantly improved. It is observed that solution 2 increases other dynamic quantities, i.e., shaking moment, driving torque compared to those of solution 1. Thus, solution 1 is considered for balancing of the cleaning mechanism. The variations of the driving-torque, shaking force and the shaking moment for one cycle of the crank are shown in Fig.3.9.

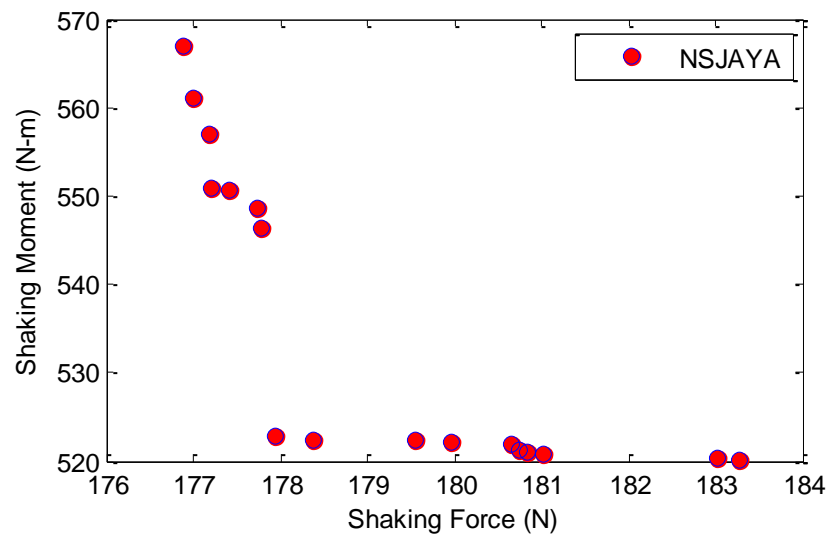


Fig. 3.8. Pareto front for a cleaning mechanism

Table 3.4. Non-dominated solutions acquired by NSJAYA

S.No.	Shaking Force (N)	Shaking Moment (N-m)
1	183.2835	520.18
2	176.8933	566.8568
3	177.9425	522.9164
4	177.7891	546.2714
5	183.0148	520.3353
6	181.0292	520.6953
7	178.3699	522.4427
8	179.5543	522.3287
9	177.0098	561.0005
10	177.1737	557.0936
11	179.9554	522.1247
12	177.2174	550.8455
13	180.655	521.977
14	177.7325	548.6091
15	177.4087	550.5608
16	180.8469	521.0591

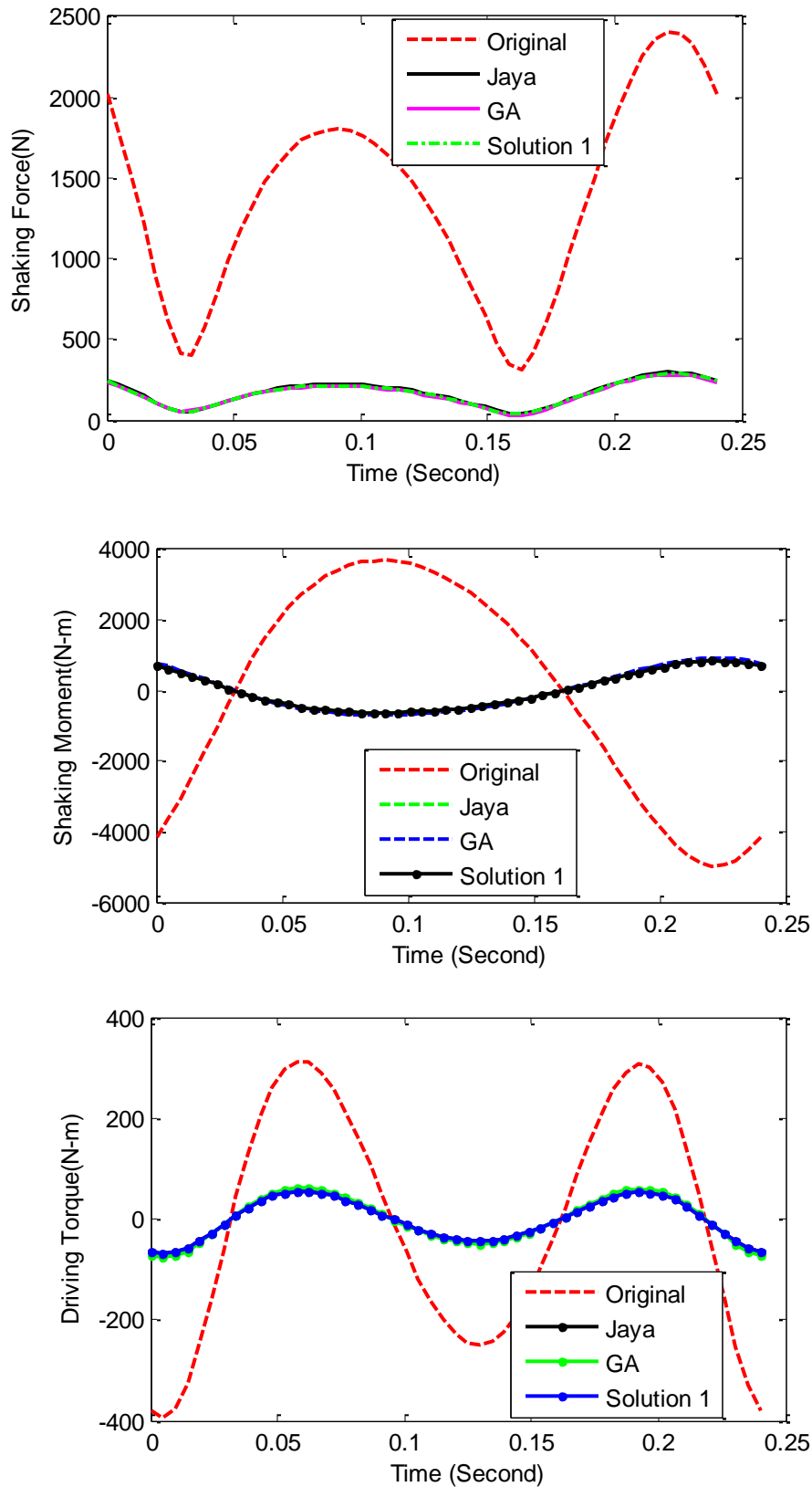


Fig. 3.9. Dynamic performance of the cleaning mechanism for one cycle of the crank.

Solution 1 minimizes the RMS values of the shaking force, the shaking moment and driving torque up to 87.80%, 83.05%, and 82.50%, respectively, compared to those

of the original mechanism. It shows that NSJAYA algorithm provides better results than those of GA and Jaya. Moreover, the multiple optimal solutions are obtained by running the GA and Jaya algorithm multiple time with a different combination of weights. As a result of that, the computation time of algorithms will increase. Hence, NSJAYA is computationally more efficient than the GA and JAYA algorithm.

The balancing of the cleaning mechanism is done numerically using optimum redistribution of link masses by choosing any solution out of the optimal Pareto set of solutions. The optimum parameters of the cleaning mechanism are determined by applying the condition of the dynamic equivalent point mass system and presented in Table 3.6.

Table 3.5. The RMS values of the dynamic quantities for the cleaning mechanism

	Driving torque (N-m)	Shaking force (N)	Shaking moment (N- m)
Original Mechanism	226.96	1503.00	3067.80
GA	44.3056	175.86	575.2204
($w_1 = 0.5, w_2 = 0.5$)	(-80.45%)	(-88.16%)	(-81.25%)
Jaya algorithm	39.7603	184.0337	520.39
($w_1 = 0.5, w_2 = 0.5$)	(-82.48%)	(-87.76%)	(-83.037%)
Solution 1	39.71	183.28	520.18
	(-82.50)	(-87.80%)	(-83.05%)
Solution 2	43.69	176.89	566.86
	(-82.50)	(-87.80%)	(-83.05%)

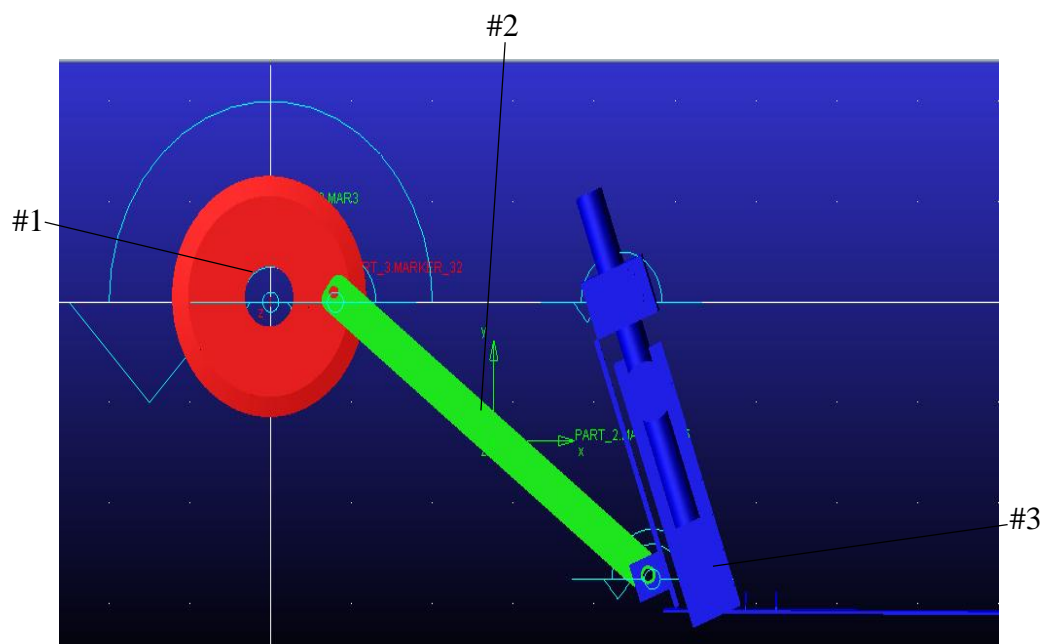


Fig. 3.10. Simulation of the cleaning mechanism using ADAMS software

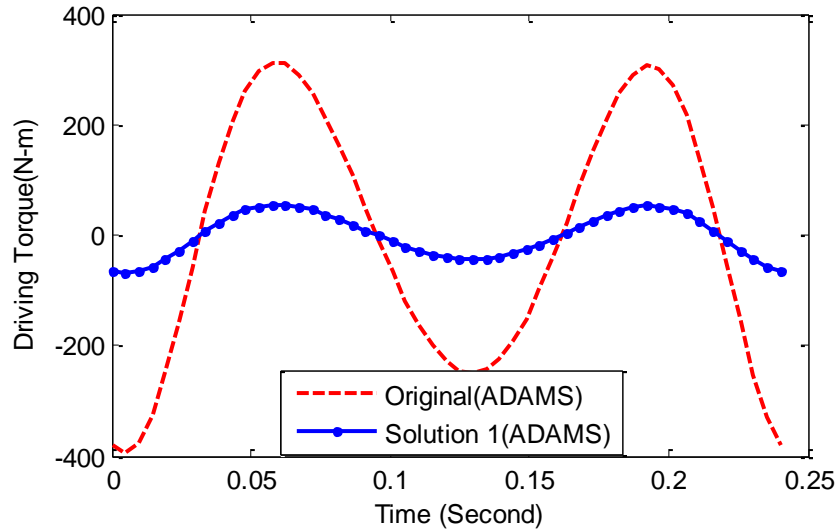


Fig. 3.11. Input torque variation in the original and optimized cleaning mechanism for the complete cycle of crank

The original and optimized configurations of the mechanism is modelled and simulated using MSC ADAMS software as shown in Fig.3.10. The input torque variation of the original and optimized mechanism is obtained using ADAMS Software as shown in Fig. 3.11. Generally, the input torque is generated by PTO of the tractor to run the mechanism. Therefore, it must be minimum for better performance of the thresher machine. It is found that the optimized mechanism reduces the maximum and minimum values of input torque about 82.49 %, and 82.50% corresponds to the original value.

Table 3.6. Optimum parameters of the balanced cleaning mechanism using solution 1

Link i	r_i (m)	m_i (kg)	I_i^G (kg- m^2)	Mass center location	
				r_{ig} (m)	θ_i (deg)
#1	0.020	0.668	4.62e-05	0.0089	245.445
#2	0.239	0.302	7.12e-04	0.1244	359.78
#3	0.139	6.022	5.1683	0.3761	84.1161
#0	0.216	-	-	-	-

3.7. Summary

In this chapter, the multi-objective optimization problem with minimization of the shaking force and shaking moment developed in the cleaning mechanism of the thresher machine is formulated using the concept of the dynamically equivalent point mass

system. The parameters of this system define the inertial properties of each moving link and are also considered as the design variables in the formulated optimization problem. The priori approach based algorithms like GA and Jaya, and the posteriori approach based algorithm as non-dominated sorting Jaya algorithm (NSJAYA) are applied to solve the formulated problem. It is observed that NSJAYA is computationally more efficient than the GA and Jaya algorithm. In this study, NSJAYA is used the first time for the balancing problem of the planar mechanism. The RMS Values of the shaking force and shaking moment are reduced using the optimum design parameters of solution 1 up to 87.80%, and 83.05%, respectively correspond to those of the original mechanism.

The original and optimized configurations of the mechanism is modelled and simulated using MSC ADAMS software. It is found that a balanced mechanism reduces the maximum and minimum values of input torque about 82.49 %, and 82.5% corresponds to the original value.

The proposed approach helps the designer for choosing the more alternatives based on the importance of objectives. Other planar and spatial mechanisms can also be balanced efficiently using the proposed approach.

Mixed variable optimization algorithms

The mixed variable optimization problems consist of the continuous, integer, and discrete variables generally used in various engineering optimization problems. These variables increase the computational cost and complexity of optimization problems due to the handling of variables. Moreover, there are few optimization algorithms which give a globally optimal solution for non-differential and non-convex objective functions.

This chapter proposes a modified Jaya algorithm for solving mixed variable optimization problems. Initially, the Jaya algorithm was developed by Rao in 2016, but it is used for continuous variable optimization problems. Therefore, the Jaya algorithm is further extended to solve mixed variable optimization problems. In the proposed algorithm, continuous variables remain in the continuous domain while continuous domains of discrete and integer variables are converted into discrete and integer domains applying bound constraint of the middle point of corresponding two consecutive values of discrete and integer variables. The effectiveness of the proposed algorithm is evaluated through examples of mixed variable optimization problems taken from the previous research works, and optimum solutions are validated with other mixed variable optimization algorithms.

4.1. Formulation of mixed variable optimization problems

This section describes the formulation of mixed variable optimization problems. Generally, these optimization problems are formulated similarly as general optimization problems, the only difference that variables may be any form of an integer, discrete and continuous. The optimization problems based on continuous and discrete variables are known as continuous and discrete optimization problems, respectively, while problems associated with discrete, integer, and continuous variables are known as mixed variable optimization problems.

The mixed variable optimization problems are defined in mathematical form as:

$$\begin{aligned}
& \text{Minimize/Maximize } Z(\mathbf{x}), \\
& \text{subjected to: } g_s(\mathbf{x}) \leq 0, \quad s = 1, 2, \dots, c_{in} \\
\mathbf{x} &= [\mathbf{x}^c, \mathbf{x}^{int}, \mathbf{x}^d]^T \\
&= [x_1^c, x_2^c, \dots, x_{nc}^c, x_1^{int}, x_2^{int}, \dots, x_{n_{int}}^{int}, x_1^d, x_2^d, \dots, x_{n_d}^d] \\
& \quad (4.1) \\
x_i^d &\in D_i, \quad D_i(d_{i1}, d_{i2}, d_{i3}, \dots, d_{im_i}), \quad i = 1, \dots, n_d \\
x_i^{int} &\in G_i, \quad G_i(g_{i1}, g_{i2}, g_{i3}, \dots, g_{iq_i}), \quad i = 1, \dots, n_{int} \\
x_i^{cL} &\leq x_i^c \leq x_i^{cU}; \quad x_i^{dL} \leq x_i^d \leq x_i^{dU}; \quad x_i^{intL} \leq x_i^{int} \leq x_i^{intU}
\end{aligned}$$

Where $Z(\mathbf{x})$ and $g(\mathbf{x})$ denotes the objective functions and non-equality constraints, respectively. ‘ c_{in} ’ denotes the numbers of total inequality constraints. $\mathbf{x} = [\mathbf{x}^c, \mathbf{x}^{int}, \mathbf{x}^d]^T$ is the vector of design variables. $\mathbf{x}^c, \mathbf{x}^d$, and \mathbf{x}^{int} present the vector of continuous, discrete and integer variables, respectively. nc, n_d and n_{int} present the number of continuous, discrete and integer variables, respectively. The total number of variables is given as $n = nc + n_{int} + n_d$. d_{ij} and g_{ij} are the j^{th} discrete and integer values for i^{th} variable, respectively. m_i and q_i are the number of discrete and integer values for i^{th} variable, respectively. D_i and G_i are set of discrete values and integer values for the i^{th} variable, respectively. Although, the number of discrete values may be different for each variable. x_i^{cL}, x_i^{dL} , and x_i^{intL} are the lower bounds of the i^{th} continuous, discrete and integer variables, respectively. x_i^{cU}, x_i^{dU} , and x_i^{intU} are the upper bounds of the i^{th} continuous, discrete and integer variables, respectively. If there are any equality constraints in the optimization problem, these can be converted into inequality constrained.

The constraint optimization problem described in Eq. (4.1) is changed into an unconstrained optimization problem using penalty function (Singh et al., 2017). The objective function is penalized for an infeasible solution for each constraint violation. Hence, the global optimum solutions, those satisfy all the constraints, are obtained. Finally, the unconstrained optimization problem is posed as a combination of the objective function and penalty function.

$$\varphi(x) = f(x) + \sum_{j=1}^{c_{in}} C_j * p^j \quad (4.2)$$

$$x_i^L \leq x_i \leq x_i^U \quad (4.3)$$

Where p^j ($j = 1, \dots, c_{in}$) presents the penalty value of 10^5 assigned to objective function for constraint violation. The Boolean Function (Mundo et al., 2009) is represented by C_j defined as:

$$C_j = \begin{cases} 0 & \text{if } g_s(x) \leq 0 \\ 1 & \text{otherwise} \end{cases} \quad (4.4)$$

4.2. A modified Jaya algorithm for mixed variable optimization problems

This section describes a modified Jaya algorithm proposed for mixed variable optimization problems. However, the original Jaya algorithm was developed by Rao in 2016. It is a population-based evolutionary algorithm that does not require any parameter for its convergence. It works only on one phase compared to TLBO that works on two phases (teacher and learner phases). This algorithm converges rapidly toward the optimal solution in each iteration (Rao, 2019; Rao, R. V. et al., 2017). The readers may use reference (Rao, 2016) for the flowchart of original Jaya algorithm. Generally, continuous variables are converted into noncontiguous variables using rounding off operators. Rounding-off operation of a continuous variable may violate the constraints due to the existence of optimal continuous solutions on the boundaries of the functional constraints. Further, researchers are checked optimum values for integer and discrete variables before rounding-off corresponding continuous variables so that constraints are not violated after rounding-off. Although, this operation increases calculation time.

Therefore, the original Jaya algorithm is modified to handle the various design variables in optimization problems without violation of constraints. This algorithm has not any algorithmic parameter and converges fast to the optimal global solution. This algorithm begins with the initialization of parameters. Initial solutions of all variables are generated in continuous space randomly. Further, continuous variables remain in continuous space while the continuous domain of discrete/ integer variables is converted into discrete domain by using the bound constraint of a middle point of corresponding two consecutive values of discrete and integer variables. Such as continuous solutions of discrete and integer variables (x_{ij}^{nc}) lie between corresponding discrete values $x^{nc}(k)$ and $x^{nc}(k+1)$. Then, continuous solutions of discrete and integer variables are converted into discrete solutions as If $x_{ij}^{nc} \leq \frac{x^{nc}(k)+x^{nc}(k+1)}{2}$, $x_{ij}^{nc} = x^{nc}(k)$ otherwise $x_{ij}^{nc} = x^{nc}(k+1)$ as shown in Fig.4.1. Further, best and worse solutions of the objective function are compared with

previous solutions at each iteration. Thus, the best solution is stored, and the worse solution is updated in each iteration.

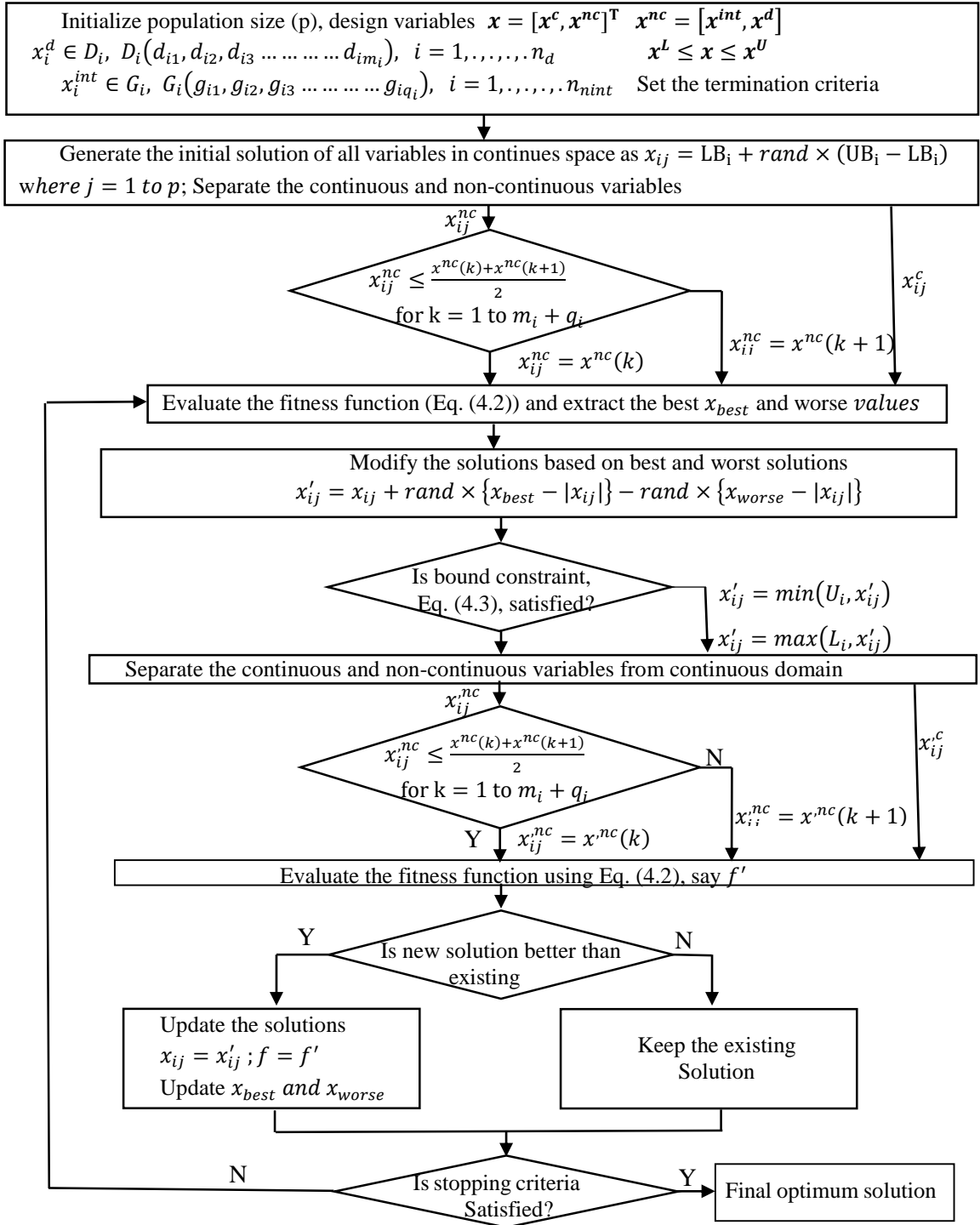


Fig. 4.1. A modified Jaya algorithm for mixed variable optimization problems

The procedure of this algorithm continues until the termination criteria are satisfied. The termination criteria are described by function evaluations and the number of

generations. The number of function evaluations is the product of the number of iterations and initial populations or population size, i.e. (The number of function evaluations= population size \times no. of iterations). So, function evaluations are not affected by design variables, but the computational time of the algorithm can be increased. Generally, an algorithm is efficient if it takes the fewer number of function evaluations. The detailed procedure of this algorithm is explained by the flowchart as shown in Fig. 4.1. Moreover, this algorithm reduces the computational effort than the other mixed optimization algorithms. However, the first time it is applied to mixed optimization problems in this study.

4.3. Design Problems

In this section, the effectiveness of the proposed algorithm as described in the previous section is validated through five design problems taken from literature. These design problems have been tested by other evolutionary mixed optimization algorithms such as EP(Cao, 2000), EA(Deb, 1997), GAs(Coello et al., 2001; Rajeev and Krishnamoorthy, 1992; Rao and Xiong, 2005; Wu and Chow, 1995), PSOs(Guo et al., 2004; He et al., 2004; Kitayama et al., 2006; Li et al., 2009; Nema et al., 2008; Sun et al., 2011), ACO (Camp and Bichon, 2004) , ABC (Sonmez, 2011), TLBO (Dede, 2014), SAs(Kripka, 2004; Zhang and Wang, 1993). However, these problems involve continuous, discrete, and integer variables. Moreover, the optimum results of the proposed algorithm are validated to optimum solutions achieved by other algorithms. This algorithm is implemented in Mat Lab 2014.

4.3.1. Example 1: Design of a welded beam

This design problem includes mixed variables taken from (Lampinen and Zelinka, 1999; Nema et al., 2008) and objective of this example is to determine the minimum cost of the welded beam design shown in Fig.4.2. There are seven nonlinear and linear constraints. The length of the welded joint (l), the thickness of the weld (h), the bar breadth (b) and the bar thickness (t) are taken as design variables. l and h are two integer variables, while b and t are two discrete variables whose values are multiples of 0.5. These design variables are defined in vector form as $\mathbf{x} = [x_1, x_2, x_3, x_4]^T = [l, h, b, t]^T$

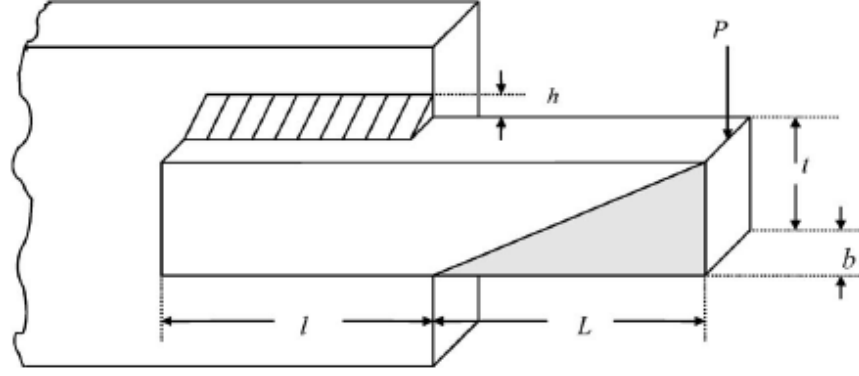


Fig. 4.2. Welded beam design (Nema et al., 2008)

The optimization formulation of this design is given as

$$\min f(x) = 1.10471x_2^2x_1 + 0.04811x_3x_4(14 + x_1)$$

Subjected to

$$\left. \begin{aligned} g_1(x) &= x_2 - x_3 \leq 0 \\ g_2(x) &= \sigma(x) - \sigma_{max} \leq 0 \\ g_3(x) &= 0.10471x_2^2 + 0.04811x_3x_4(14 + x_1) - 5.0 \leq 0 \\ g_4(x) &= \tau(x) - \tau_{max} \leq 0 \\ g_5(x) &= 0.125 - x_2 \leq 0 \\ g_6(x) &= P - P_c(x) \leq 0 \\ g_7(x) &= \delta(x) - \delta_{max} \leq 0 \\ 0.1 &\leq x_1 \leq 10.0 \\ 0.1 &\leq x_2 \leq 2.0 \\ x_3, x_4 &\in [0.5, 1.0, 1.5, 2.0, 2.5, \dots, 9.5, 10.0] \end{aligned} \right\} \quad (4.5)$$

Where

$$\delta(x) = \frac{4PL^3}{Ex_4^3x_3}, \sigma(x) = \frac{6PL}{x_3x_4^2}, P_c(x) = \frac{4.013\sqrt{EG(x_4^2x_3^6/36)}}{l^2} \left(1 - \frac{x_4}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P \left(L + \frac{x_1}{2} \right), R = \frac{x_1^2}{4} + \left(\frac{x_2 + x_4}{2} \right)^2$$

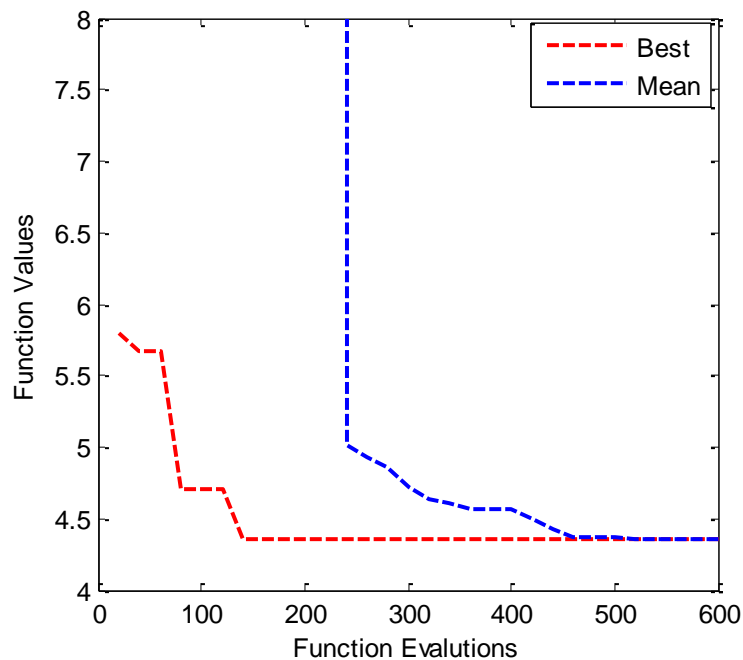
$$J = 2\sqrt{2}x_1x_2 \left(\frac{x_1^2}{12} + \left(\frac{x_2 + x_4}{2} \right)^2 \right)$$

The design parameters are taken as

$$\begin{aligned} P &= 6000lb, E = 30 \times 10^6 lbf/in^2, G \\ &= 12 \times 10^6 lbf/in^2, L = 14 in, \tau_{max} = 13600 lbf/in^2, \delta_{max} \\ &= 0.25in, \sigma_{max} = 30000 lbf/in^2 \end{aligned}$$

Table 4.1. Optimum designs of the welded beam

Design variables (\mathbf{x})	MDHGA (Rao and Xiong, 2005)	HPB (Nema et al., 2008)	MPSO (Sun et al., 2011)	This study
x_1	2.0	1	1	1
x_2	1.0	1	1	1
x_3	1.0	1	1	1
x_4	4.5	4.5	4.5	4.5
$f(x)$	5.67334	4.3521	4.3521	4.3521
Function Evaluations	-	800	6750	600
Constraints violation	None	None	None	None

**Fig. 4.3.** Convergence graph of best and mean values of objective function for welded beam design

20 initial populations and 30 iterations are chosen for this design example. The best objective function values and the mean of all function values corresponding to the best run are obtained in 20 independent runs. Convergence performance of best and mean values of the objective function is presented in Fig.4.3. The best and mean values of the objective function are obtained in 600 function evaluations as 4.3521 and 4.3521,

respectively. The optimum solutions of the welded beam design are validated with the optimum solutions of other algorithms as shown in Table 4.1. Table 4.1 presents that the proposed algorithm gives a better optimum design than that of (Rao and Xiong, 2005) and equal to that of (Nema et al., 2008; Sun et al., 2011). But, the proposed algorithm takes less function evaluation for finding the best objective function value compared to other evolutionary optimization algorithms.

4.3.2. Example 2: Pressure Vessel Design

This design deals with the pressure vessel design taken from the literature (Sandgren, 1990) as shown in Fig.4.4. The objective of this design is to minimize the manufacturing cost of the pressure vessel with specific design constraints. The design variables are taken as shell thickness (T_s), spherical head thickness (T_h), radius (R), and shell length (L). T_s and T_h are the discrete variables whose values are multiples of 0.0625 inches while R and L are defined as continuous variables. Design variables are described in vector form as

$$\mathbf{x} = [x_1, x_2, x_3, x_4]^T = [T_h, T_s, l, R]^T$$

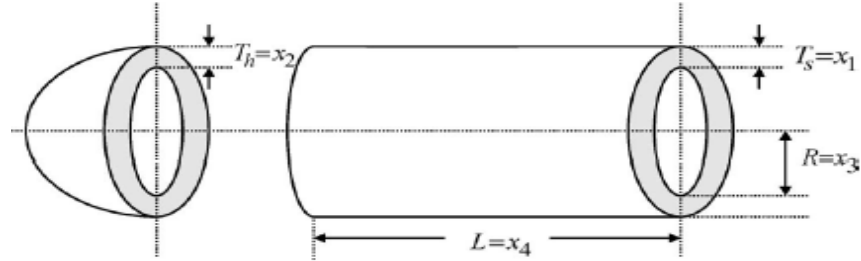


Fig. 4.4. Pressure vessel design (Sandgren, 1990)

The optimization problem is posed (Sandgren, 1990) as:

$$\min f(\mathbf{x}) = 0.6224x_2x_3x_4 + 1.7781x_1x_4^2 + 3.1661x_2^2x_3 + 19.84x_2^2x_4$$

Subjected to

$$\left. \begin{aligned} g_1(\mathbf{x}) &= 0.0193x_4 - x_2 \leq 0 \\ g_2(\mathbf{x}) &= 0.00954x_4 - x_1 \leq 0 \\ g_3(\mathbf{x}) &= 1296,000 - \pi x_4^2 x_3 - \frac{4}{3} \pi x_4^2 \leq 0 \\ g_4(\mathbf{x}) &= x_3 - 240 \leq 0 \\ 10 &\leq x_3 \leq 200 \\ 10 &\leq x_4 \leq 200 \\ x_1, x_2 &\in [0.0625, 0.1250, 0.1875, 0.2500, \dots, \dots, 6.00, 6.0625, 6.1250, 6.1875] \end{aligned} \right\} (4.6)$$

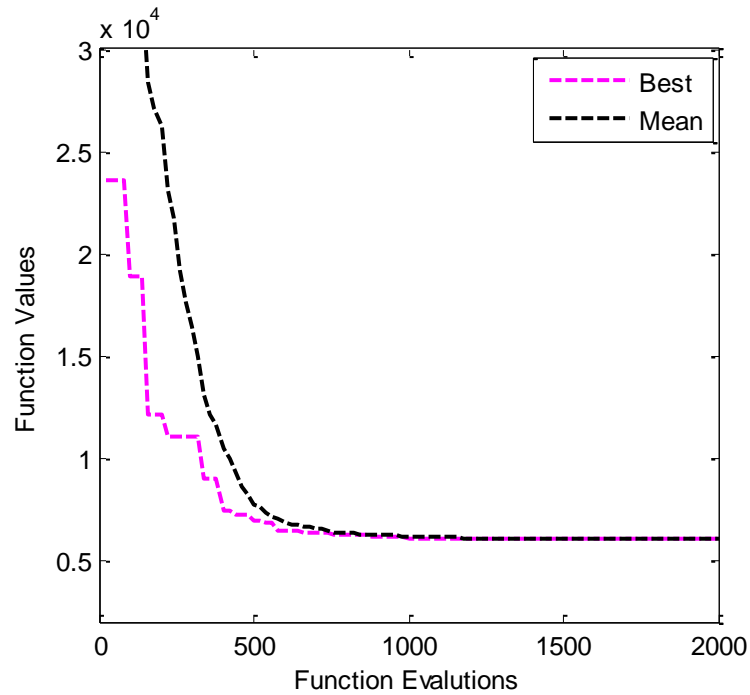


Fig. 4.5. Convergence of best and mean values of objective function for pressure vessel design

Table 4.2. Optimum solutions for design of pressure vessel

Design variables (x)	EP (Cao, 2000)	EA (Deb, 1997)	GA (Coello et al., 2001)	PSO (He et al., 2004)	HPB (Nema et al., 2008)	MPSO (Sun et al., 2011)	This Study
x_1	0.625	0.5	0.4375	0.4375	0.4375	0.4375	0.4375
x_2	1	0.9345	0.8125	0.8125	0.8125	0.8125	0.8125
x_3	90.7821	112.679	176.654	176.6366	176.6366	176.636792	176.636792
x_4	51.1958	48.329	40.0974	42.09845	42.09845	42.098446	42.098446
$f(x)$	7108.616	6410.381	6059.946	6059.714	6059.714	6059.718932	6059.700
Function Evaluations	100000	42000	30000	30,000	4013	4,00000	2000
Constraints violation	None	None	None	None	None	None	None

The number of iterations and initial populations are considered as 100 and 20 respectively for this example. Ten independent runs are chosen to find the best values of the objective function and mean of all objective function value corresponding to the best run. The convergence rate of best and mean objective function values are shown

in Fig. 4.5. The best and mean values of the objective function are obtained in 2000 function evaluations as 6059.70 and 6059.74, respectively. The optimum results for the design of pressure vessel are shown in Table 4.2. Table 4.2 shows that the optimum value of the objective function is better than that of other optimization algorithms. Moreover, the proposed algorithm takes less function evaluation for finding the best objective function value compared to different evolutionary optimization algorithms.

4.3.3. Example 3: 10-Bar Planar Truss design

A 10 bar planar truss design is taken as an optimization problem shown in Fig.4.6 (Rajeev and Krishnamoorthy, 1992). In this truss structure design, a minimization optimization problem is formulated by considering the weight of the truss as the objective function with the constraints of displacements at each nodal point and the stress induced in each member. This problem is based on discrete optimization problems in which cross-sectional areas of each member are discrete variables. The nodes 2 and 4 are subjected to a vertical nodal load of 100 kips. Modulus of elasticity of the material of each bar and density are considered as $E = 10,000$ ksi and $\rho = 0.1$ lb/in³ respectively. The allowable displacements for the free nodes and the allowable stress for all members are taken as ± 2 inch. in both directions and ± 25 ksi respectively. 10 discrete design variables and their values are selected from the standard values $D = \{1.62, 2.38, 1.99, 1.80, 2.13, 2.62, 2.88, 3.09, 2.93, , 2.63, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 14.2, 13.9, 15.5, 16.0, 16.9, 18.8, 1.99, 22.0, 22.9, 26.5, 30.0, 33.5\}$ (inch²) [10]. The vector form of design variables is expressed as:

$$x = [x_i] = [A_i] \quad i = 1 \text{ to } 10$$

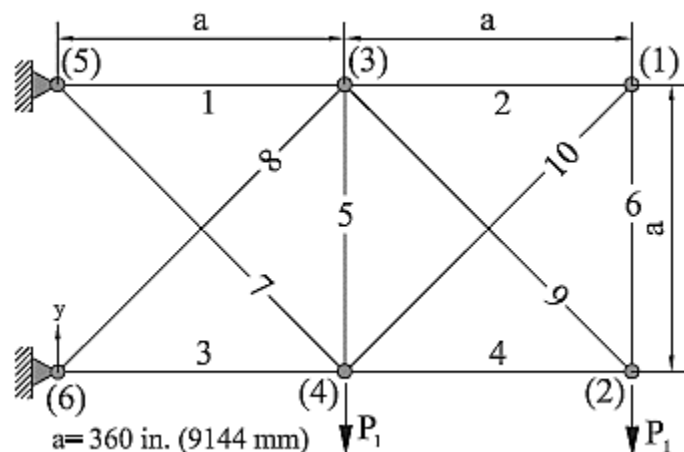


Fig. 4.6. A planar 10 bar truss structure(Rajeev and Krishnamoorthy, 1992)

The formulation of the problem is presented as:

$$\begin{aligned} \text{minimize weight (f)} &= \sum_{i=1}^n \rho x_i L_i \\ \text{Subjected to} & \\ \left. \begin{aligned} \frac{\sigma_i}{\sigma_a} - 1 &\leq 0 \\ \frac{u_i}{u_a} - 1 &\leq 0 \end{aligned} \right\} & \end{aligned} \quad (4.7)$$

Where L_i , σ_i , u_i and A_i are the length, stresses, deflection and of cross-sectional area i th member respectively.

For this problem, the initial populations and the number of iterations are set to 10 and 95, respectively. 10 independent runs are performed to find out the best values of the objective function, and the mean of all objective function values correspond to the best run. The best and mean values of the objective function in 10 runs are 5,490.74 and 5,493.54, respectively. This algorithm takes 950 function evaluations. Convergence plot of best and mean values of the objective function is shown in Fig.4.7. The comparison of optimum solutions for planar 10 bar truss is presented in Table 4.3.

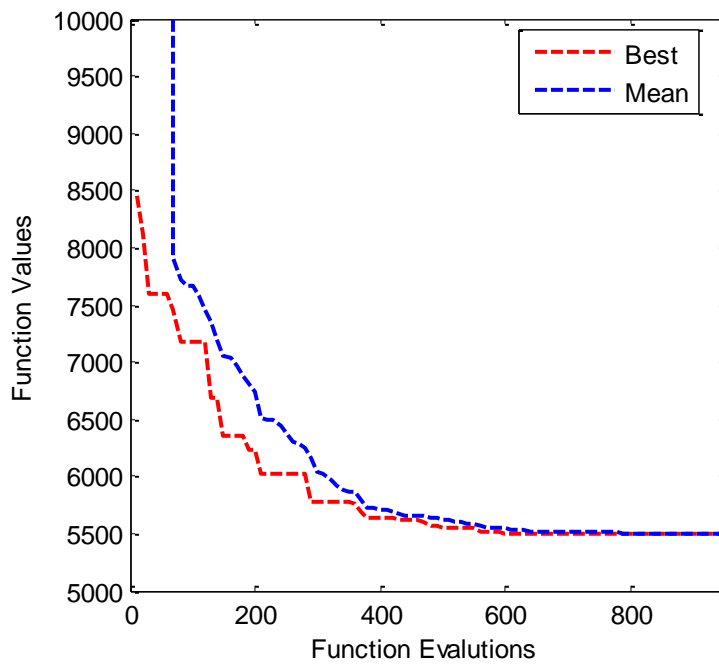


Fig. 4.7. Convergence of best and mean values of objective function for planar 10-bar planar truss

Table 4.3. Comparison of Optimum solutions for planar 10-bar truss structure

Design variables (in ²)	GA (Rajeev and Krishna moorthy, 1992)	SA (Kripka, 2004)	HPSO (Li et al., 2009)	ACO (Camp and Bichon, 2004)	ABC (Sonmez, 2011)	TLBO (Dede, 2014)	Jaya (Rao, 2016)	This study
A ₁	33.5	33.5	30	33.5	33.5	33.5	33.5	33.5
A ₂	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A ₃	22	22.9	22.9	22.9	22.9	22.9	22.9	22.9
A ₄	15.5	14.2	13.5	14.2	14.2	14.2	14.2	14.2
A ₅	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A ₆	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A ₇	14.2	7.97	7.97	7.97	7.97	7.97	7.97	7.97
A ₈	19.9	22.9	26.5	22.9	22.9	22.9	22.9	22.9
A ₉	19.9	22	22	22	22	22	22	22
A ₁₀	2.62	1.62	1.8	1.62	1.62	1.62	1.62	1.62
W(lb)	5,613.84	5,490.74	5,531.9	5,490.74	5,490.74	5,490.74	5,490.74	5,490.74
Function Evaluation	N/A	N/A	50,000	10,000	25,800	1,000	950	950
Constraints violation	none	none	none	none	none	none	none	none

Table 4.3 shows that the optimum objective function value is better than that of (Rajeev and Krishnamoorthy, 1992), (Li et al., 2009) and close to that of (Camp and Bichon, 2004), (Sonmez, 2011), (Dede, 2014), (Kripka, 2004), and (Rao, 2016) . But, the proposed algorithm takes less function evaluation for finding the best objective function value compared to other evolutionary optimization algorithms. Thus, the proposed approach is more efficient and reduces the computational efforts.

4.3.4. Example 4: A helical spring design

This example consists of the design of helical spring under the constant and axial load as shown in Fig.4.8 (Nema et al., 2008). The minimization of spring weight is considered as an objective function with certain inequality constraints. The number of spring coils (N), the outside diameter of the spring (D), and the spring wire diameter (d) are the design variables. This problem involves integer, discrete and continuous variables. Where, the number of coils (N) is an integer variable, outside diameter of the spring (D) is a continuous variable and the spring wire diameter is a discrete variable, whose standard values are chosen. Design variables are in vector form as:

$$\mathbf{x} = [x_1, x_2, x_3]^T = [D, N, d]^T$$

The formulation of the optimization problem is posed as:

$$\begin{aligned}
\min f(x) &= \frac{\pi^2 x_2 x_1^2 (x_3 + 2)}{4} \\
\text{Subjected to} \\
g_1(x) &= \frac{8C_f F_{\max} x_2}{\pi x_3^3} - S \leq 0 \\
g_2(x) &= l_f - l_{\max} \leq 0 \\
g_3(x) &= d_{\min} - x_3 \leq 0 \\
g_4(x) &= x_2 - D_{\max} \leq 0 \\
g_5(x) &= 3.0 - \frac{x_1}{x_3} \leq 0 \\
g_6(x) &= \delta - \delta_m \leq 0 \\
g_7(x) &= \delta + \frac{F_{\max} - F}{K} + 1.05(x_2 + 2)x_3 - l_f \leq 0 \\
g_8(x) &= \delta_w - \frac{F_{\max} - F}{K} \leq 0 \\
0.6 &\leq x_1 \leq 3 \\
1 &\leq x_2 \leq 70 \\
x_3 &\in [0.009, 0.0095, 0.0104, 0.0118, 0.0128, 0.0132, 0.014, 0.015, 0.0162, 0.0173, 0.018, 0.020, 0.023, \\
&0.025, 0.028, 0.032, 0.035, 0.041, 0.047, 0.054, 0.063, 0.072, 0.080, 0.092, 0.105, 0.120, 0.135, 0.148 \\
&0.162, 0.177, 0.192, 0.207, 0.225, 0.244, 0.263, 0.283, 0.307, 0.331, 0.362, 0.394, 0.4375, 0.500]
\end{aligned} \tag{4.8}$$

Where

$$C_f = \frac{4 \left(\frac{x_1}{x_3} \right) - 1}{4 \left(\frac{x_1}{x_3} \right) - 4} + \frac{0.615x_3}{x_1}, K = \frac{Gx_3^4}{8x_2x_1^3}, \delta = \frac{F}{K}, l_f = \frac{F_{\max}}{K} + 1.05(x_2 + 2)x_3$$

The values of predefined parameters of spring are given as

$$\begin{aligned}
F_{\max} &= 1000.0 \text{ lb}, l_{\max} = 14.0 \text{ in}, d_{\min} = 0.2 \text{ in}, S = 189000.0 \text{ lbf/in}^2, d_{\max} = 3.0 \text{ in}, F \\
&= 300.0 \text{ lb}
\end{aligned}$$

$$\delta_m = 6.0 \text{ in}, \delta_w = 1.25 \text{ in}, G = 11.5 \times 10^6 \text{ lbf/in}^2$$

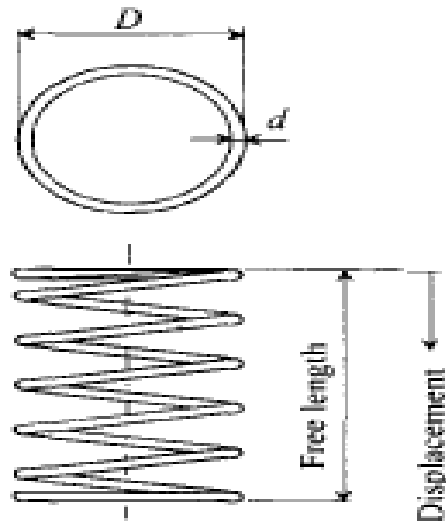
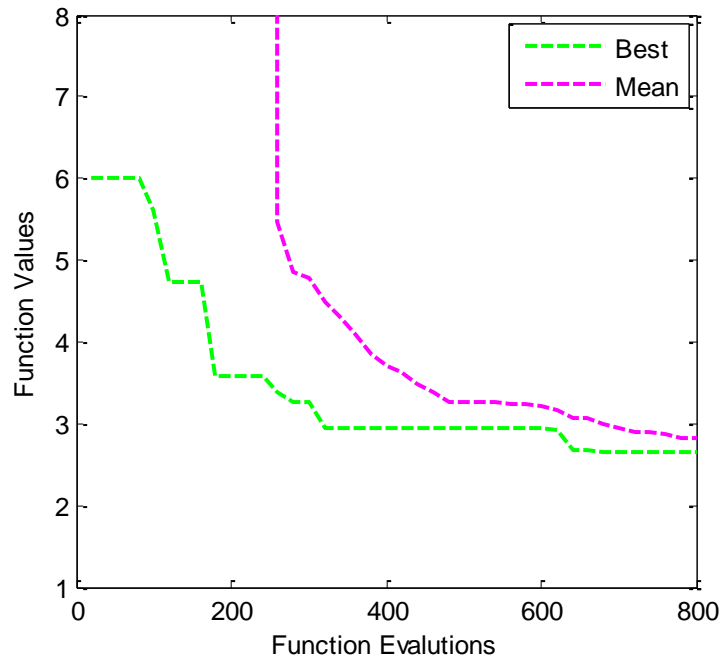


Fig. 4.8. Design of helical spring (Nema et al., 2008)

Table 4.4. Optimal solutions comparison for the spring design

Design variables (\mathbf{x})	EA (Deb, 1997)	DE(Lampinen and Zelinka, 1999)	PSO (He et al., 2004)	HPB(Nema et al., 2008)	MPSO (Sun et al., 2011) [6]	This study
x_1	1.226	1.223041	1.223041	1.223041	1.223041	1.2231
x_2	9	9	9	9	9	9
x_3	0.283	0.283	0.283	0.283	0.283	0.283
$f(x)$	2.665	2.65856	2.65856	2.6585	2.6585	2.6586
Function Evaluations	30000	30000	30,000	835	10,000	800
Constraints violation	None	None	None	None	None	None

**Fig. 4.9.** Convergence characteristic of objective function for spring design

For this design problem, the number of iterations and initial populations are set to 40 and 20, respectively. Twenty independent runs are chosen to find the best values of the objective function and mean of all objective function values corresponding to the best run. Convergence performance of best and mean objective function values are shown in Fig.4.9. The best and mean values of the objective function are obtained in 800 function evaluations as 2.6585 and 2.7746, respectively. The optimum design of the spring is validated with the results obtained by other algorithms as shown in Table 4.4.

Table 4.4 shows that the proposed algorithm gives a better optimum value of the objective function than that of (Deb, 1997) and equal to that of (Lampinen and Zelinka, 1999), (He et al., 2004; Nema et al., 2008; Sun et al., 2011). But, the proposed algorithm takes less function evaluation for finding the best objective function value compared to other evolutionary optimization algorithms.

4.3.5. Example 5: Compound gear train design

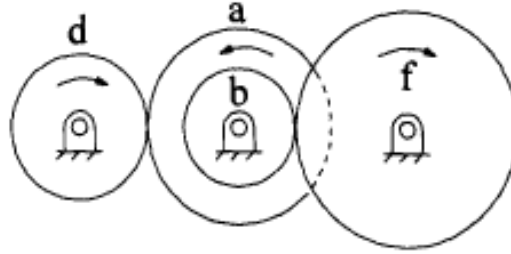


Fig. 4.10. Gear train design (Guo et al., 2004)

The purpose of this design is to obtain the optimum gear ratio of the gear train arrangement as presented in Fig.4.10 (Guo et al., 2004). The ratio of the output shaft angular velocity to the input shaft angular velocity is known as the gear ratio of the gear train. The effective overall gear ratio G_r is expressed as;

$$G_r = \frac{\omega_{out}}{\omega_{in}} = \frac{T_b T_d}{T_a T_f}$$

Where ω_{in} and ω_{out} represent input and output shafts angular velocities, respectively, and, the number of teeth of each gear is represented by 'T'. Each gear's teeth number is taken as design variables. However, all design variables are integers whose values lie between 12 and 60. A vector form of design variables is expressed as

$$\mathbf{x} = [T_b \ T_d \ T_a \ T_f]^T = [x_1 \ x_2 \ x_3 \ x_4]^T$$

The optimization problem is posed as:

$$\text{Min } f(\mathbf{x}) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2 \quad (4.9)$$

$$12 \leq x_i \leq 60, \quad i = 1 \text{ to } 4$$

The initial populations of 150 and the number of iterations of 100 are decided for this example. The best value of the objective function and the mean of all function values corresponding to the best run are obtained for 30 independent runs. Convergence performance of objective function values is shown in Fig. 4.11. The best and mean values of the objective function are obtained in 15000 function evaluations as $2.7 \times$

10^{-12} and 1.4311×10^{-7} , respectively. The optimum design of the gear train is validated with the design achieved by other algorithms as shown in Table 4.5. Table 4.5 presents that the proposed algorithm gives nearly same optimum solution of the objective function to that of the different evolutionary algorithm. But, it minimizes the percentage of error known as the difference between the mean and best values of objective functions, compared to other optimization algorithms. Thus, the proposed algorithm can be effectively applied to integer optimization problems.

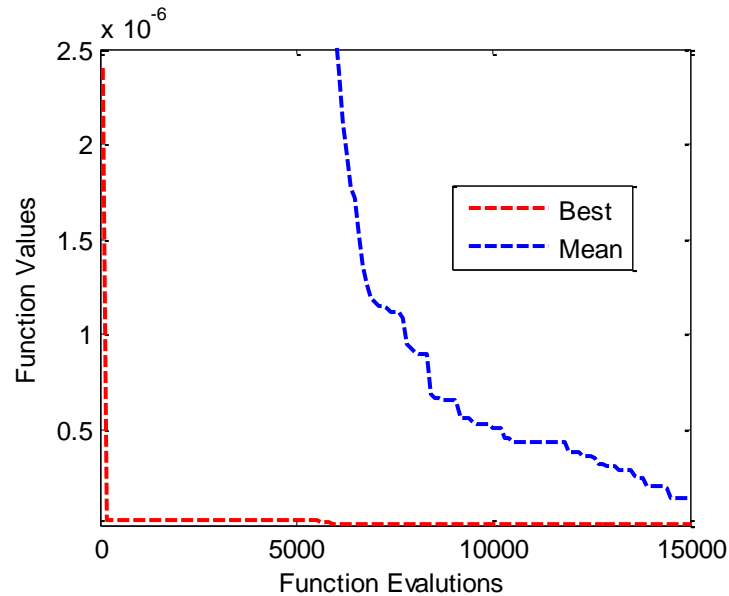


Fig. 4.11. Convergence graph of best and mean values of objective function for gear train design

Table 4.5. Optimal design for the compound gear train

Design variables (\mathbf{x})	SA(Zhang and Wang, 1993)	EP(Cao, 2000)	HSIA(Guo et al., 2004)	MPSO (Sun et al., 2011)	Jaya (Rao, 2016)	This study
x_1	30	30	16	16	16	16
x_2	15	15	19	19	19	19
x_3	52	52	43	43	43	43
x_4	60	60	49	49	49	49
$f(x)$	2.36×10^{-6}	2.36×10^{-6}	2.7×10^{-12}	2.7×10^{-12}	2.7×10^{-12}	2.7×10^{-12}
Function Evaluations	-	-	-	4,00000	18000	15000
% error	0.033	0.033	0.0011	0	9.8×10^{-13}	9.8×10^{-13}
Gear ratio	0.14423	0.14423	0.14428	0.14428	0.14428	0.14428

4.4. Summary

In this chapter, a modified Jaya algorithm is proposed for the mixed variable optimization problems. Original Jaya algorithm has been developed for continuous optimization problems. Therefore, Jaya algorithm is further extended for solving the mixed variable optimization problems. In the proposed algorithm, continuous variables remain in the continuous domain while continuous domains of discrete and integer variables are converted into discrete and integer domains applying bound constraint of the middle point of corresponding two consecutive values of discrete and integer variables.

Furthermore, the efficiency of a modified Jaya algorithm is demonstrated by five design problem taken from literature. Moreover, the optimum results obtained from the proposed algorithm are compared with those of well-known optimization algorithms. The results show that it takes fewer function evaluations without violation of the design constraints and gives the better and nearly close results compared to other optimization algorithms. Other mixed, continuous, and discrete variable optimization problems can also be effectively solved using this algorithm. Hence, a modified Jaya algorithm may be an essential tool for a wide range of mixed variables problems.

Optimal two-plane discrete balancing

This chapter presents the optimum two-plane discrete balancing procedure for the rigid rotor. The discrete two-plane balancing in which rotor is balanced to minimize the residual effects or the reactions on the bearing supports using discrete parameters such as masses and their angular positions on two balancing planes. A multi-objective optimization problem is formulated by considering reaction forces on the bearing supports as multi-objective functions and discrete parameters as the design variables. These multi-objective functions are converted into a single-objective function using appropriate weighting factors. The formulated problem is solved using a modified Jaya algorithm explained in chapter 4. It is found that the modified Jaya algorithm is computationally more efficient than the genetic algorithm (GA). A number of masses per plane are used to balance the rotor. A comparison of reaction forces using the number of masses per plane is also investigated. The effectiveness of the proposed methodology is tested by the balancing problem of rotor available in the literature. It is also applied for the balancing of the threshing drum. Automated dynamic analysis of mechanical systems (ADAMS) software and experimental tests are used for validation of a developed balancing approach.

5.1. Dynamic model of rigid rotor

The balancing of the rotor using the numbers of masses at a corresponding angular position is investigated in this section. The rigid rotor requires two planes for its balancing, while the flexible rotor requires multiple planes for its balancing. Further, reaction forces acting on the bearings are calculated using Newton-Euler equation. A rigid rotor of mass (M) is mounted on bearings P and Q shown in Fig.5.1. A coordinate system (x, y, z) is fixed coordinate system while rotating coordinate system (x_r, y_r, z_r) is attached to the shaft rotating at a constant angular velocity ω about the z -axis. The origin of the coordinate system is denoted by the point 'O.' Two balance planes are considered which are centered at points c_1 and c_2 , respectively. The center of mass of the rotor 'G' is eccentric at a distance of 'e' from the axis of rotation due to unbalance of the rotor. F_P and F_Q are the unbalance reaction forces acting at angle θ_P and θ_Q on the bearings P and Q, respectively. l_P and l_Q are lengths from 'O' to the bearings P and

Q respectively. Numbers of discrete masses m_{ij} ($j = 1$ to N_i) placed at radius R_i and angular positions α_{ij} measured from the x -axis, as shown in Fig. 5.1 on balance plane i ($i = 1, 2$) for rigid rotor, while ($i = 1, \dots, p$) for flexible rotor. Where p represents the number of planes.

The center of mass ('G') position is given as:

$$OG(e_x, e_y)$$

where $e_x = \cos(\theta + \omega t)$; $e_y = \sin(\theta + \omega t)$

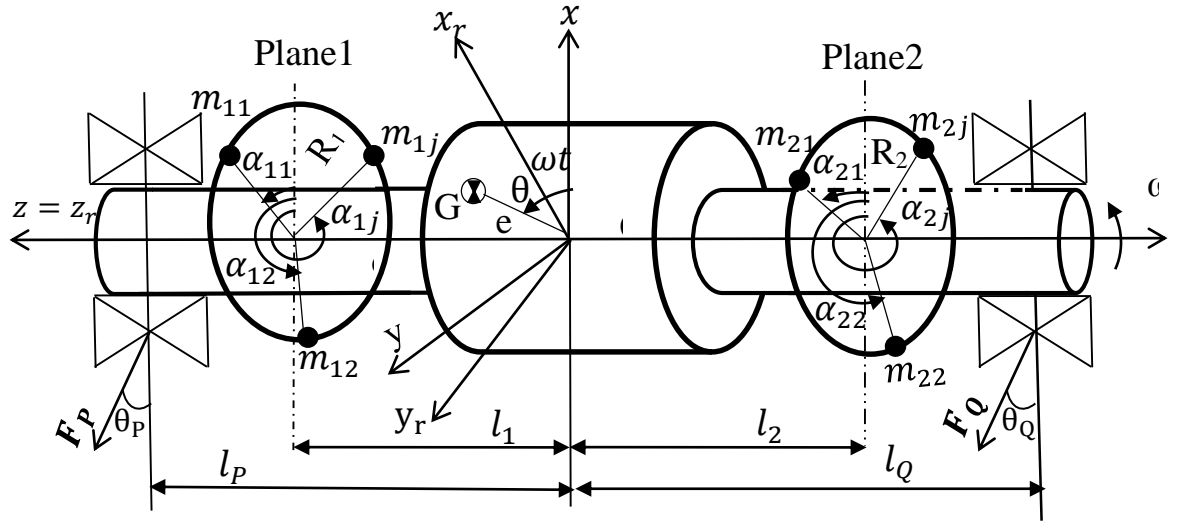


Fig. 5.1. Rigid rotor model

Newton-Euler equations of motion (Chaudhary and Saha, 2009) for the rigid body are written as

$$M\dot{\mathbf{v}} = \mathbf{f}_0 \quad (5.1)$$

$$I_G \dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}} I_G \boldsymbol{\omega} = \mathbf{M}_0 \quad (5.2)$$

Where \mathbf{f}_0 is the resultant of external forces which acting at supports, $\dot{\mathbf{v}}$ is linear acceleration of center of mass, $\dot{\boldsymbol{\omega}}$ is angular acceleration and \mathbf{M}_0 is the resultant of all the external moments about point O. I_G is Inertia tensor of the rotor given as

$$I_G = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{yz} & I_{yz} & I_{zz} \end{bmatrix} \quad (5.3)$$

The components of the reaction forces in x and y directions at supports of the rotor are determined using Eqs. (5.1) and (5.2), given in matrix form as

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ l_P & 0 & l_Q & 0 \\ 0 & l_P & 0 & l_Q \end{bmatrix} \begin{bmatrix} F_{Px} \\ F_{Py} \\ F_{Qx} \\ F_{Qy} \end{bmatrix} = \begin{bmatrix} M\omega^2 e_x + \omega^2 \sum_{i=1}^p \sum_{j=1}^{N_i} (m_{ij} R_i \cos \alpha_{ij}) \\ M\omega^2 e_y + \omega^2 \sum_{i=1}^p \sum_{j=1}^{N_i} (m_{ij} R_i \sin \alpha_{ij}) \\ -I_{yz} \omega^2 + \omega^2 \sum_{i=1}^p \sum_{j=1}^{N_i} (m_{ij} R_i \cos \alpha_{ij}) l_i \\ I_{xz} \omega^2 + \omega^2 \sum_{i=1}^p \sum_{j=1}^{N_i} (m_{ij} R_i \sin \alpha_{ij}) l_i \end{bmatrix} \quad (5.4)$$

Note that m_{ij} is the j^{th} mass at the i^{th} plane and α_{ij} is the angular position of m_{ij}

Now, the reaction forces at supports P and Q of the balanced rotor are obtained as

$$F_P = \sqrt{F_{Px}^2 + F_{Py}^2} \quad (5.5)$$

$$F_Q = \sqrt{F_{Qx}^2 + F_{Qy}^2} \quad (5.6)$$

Where F_{Px} , F_{Py} , F_{Qx} , and F_{Qy} are the component forces at supports P and Q in x and y directions, respectively, for balanced rotor. Further, the reaction forces at supports P and Q for unbalance rotor (when the number of balance mass per plane is zero) are calculated using Eq. (5.4) as

$$F_{PO} = \sqrt{F_{PxO}^2 + F_{PyO}^2} \quad (5.7)$$

$$F_{QO} = \sqrt{F_{QxO}^2 + F_{QyO}^2} \quad (5.8)$$

Where F_{PxO} , F_{PyO} , F_{QxO} , and F_{QyO} are the component forces at supports P and Q, respectively, for unbalanced rotor. The reaction forces at supports P and Q are normalized in order to formulate the objective function of the optimization problem. The normalized forces $F_{P,\text{norm}}$ and $F_{Q,\text{norm}}$ with respect to F_{PO} and F_{QO} are written respectively, as:

$$F_{P,\text{norm}} = \frac{F_P}{F_{PO}} \quad (5.9)$$

$$F_{Q,\text{norm}} = \frac{F_Q}{F_{QO}} \quad (5.10)$$

5.2. Formulation of discrete optimization problem

Balancing of the rigid rotor using masses per balance plane is formulated as the optimization problem in this section. The main purpose of balancing is to minimize vibration effects. Therefore, the minimization of the reaction forces F_P and F_Q acting at supports is expressed as multi-objective functions with discrete constraints on design variables. These multi-objective functions are transformed into a single objective function using appropriate weighting factors. The rotor is balanced by placing the masses at different angular positions at a fixed radius taken from the finite sets of masses and available angular positions, on each balancing plane. These masses and angular position per plane are taken as design variables, a $2N_i$ -vector, \mathbf{X}_i of design variable for the i^{th} plane is defined as:

$$\mathbf{X}_i = [m_{i1} \ \alpha_{i1} \ m_{i2} \ \alpha_{i2} \ m_{i3} \ \alpha_{i3} \ \dots \ m_{iN_i} \ \alpha_{iN_i}]^T \quad (5.11)$$

Where m_{ij} and α_{ij} are j^{th} mass and corresponding angular position in the i^{th} plane, respectively. Hence, the design variable $2 \sum_{i=1}^p N_i$ -vector, \mathbf{X} , for a rotor having p balancing planes in case of the flexible rotor is given by:

$$\mathbf{X} = [\mathbf{X}_1^T \ \mathbf{X}_2^T \ \dots \ \mathbf{X}_p^T]^T \quad (5.12)$$

However, the rigid rotor requires two planes for its balancing. There are total $2 \sum_{i=1}^2 N_i$ design variables for the rigid rotor. They are expressed in vector form as

$$\mathbf{X} = [\mathbf{X}_1^T \ \mathbf{X}_2^T]^T \quad (5.13)$$

Hence, the balancing procedure and the optimization formulation of the rotor will remain the same when the condition of the rotor changes from the rigid rotor to the flexible rotor. But, the design variables will be different for both rotors.

Now, the optimization problem is posed as a weighted sum of the reaction forces acting at supports given as:

$$\text{Minimize } Z = w_1 F_{P,norm} + w_2 F_{Q,norm}$$

Discrete constraints defined as

$$\left. \begin{aligned} X_{ij} \in d_{ij}; \ d_{ij} = [d_{ij1}, d_{ij2}, d_{ij3}, \dots, d_{ijD_{ij}}] ; \ d \in \{m_{ij} \ \alpha_{ij}\} \\ m_{ij} \in \{m_1; m_2; m_3 \dots \dots \dots; m_{D_1}\} \\ \alpha_{ij} \in \{\alpha_1; \alpha_2; \alpha_3 \dots \dots \dots; \alpha_{D_2}\} \end{aligned} \right\} \text{ for } i = 1,2 \text{ and } j = 1,2, \dots, N_i \quad (5.14)$$

$$LB_{ij} \leq X_{ij} \leq UB_{ij} \quad (5.15)$$

Where d_{ij} represents the set of discrete values for X_{ij} design variable, D_{ij} is the number of discrete values for X_{ij} design variable. d_{ijK} represents the k^{th} discrete values for X_{ij}

design variable taken from the available discrete values of masses and corresponding angular positions. D_1 and D_2 are total numbers of discrete values of masses and discrete values of corresponding angular position, N_i is the number of balancing masse for i^{th} plane, LB_{ij} and UB_{ij} are the lower and upper bounds on the X_{ij} design variable, w_1 and w_2 are the weighting factors assigned to forces acting at supports. The number of objective functions transforms into a single function using this approach. The various approaches for selection of the weighting factors are presented by (Marler and Arora, 2004, 2010). The weights represent the relative importance of the various objectives. However, both the objective functions have equal importance. Therefore, $w_1 = 0.5$ and $w_2 = 0.5$ are chosen for this study.

5.3. Results and Discussions

In this section, the effectiveness of the proposed methodology is tested through the numerical problems of static and dynamic unbalanced rigid rotor taken from the ref. (Messenger and Pyrz, 2013) and, it is also applied to the unbalanced threshing drum.

5.3.1. Numerical problem of the rigid rotor

In this problem, a 60 kg rotor is rotating at a constant speed of 1500 rpm about the z-axis. Dimensions and parameters of the static and dynamic unbalance rigid rotor are shown in Tables 5.1 and 5.2, respectively. The total discrete values of masses and corresponding angular positions are $D_1 = 7$ and $D_2 = 12$, respectively as

$$m_{ij}(\text{grams}) \in [0; 10; 20; 50; 100; 200; 300]$$

$$\alpha_{ij}(\text{Degrees}) \in [30; 60; 90; 120; 150; 180; 210; 240; 270; 300; 330; 360]$$

Table 5.1. Dimensions of the rigid rotor in (m)

l_p	l_Q	l_1	l_2	R_1	R_2
-0.2	0.3	-0.25	0.35	0.15	0.15

Table 5.2. Parameters of a rigid rotor

S.No.	Unbalanced Problem	e (mm)	θ (degree)	I_{xz} (kg-m ²)	I_{yz} (kg-m ²)	F_{p0} (N)	F_{Q0} (N)
1	Static	1.04	14	0	0	915.6	610.4
2	Dynamic	1.04	14	-0.004	-0.01	742.3	1017.3

The rotor can be balanced using a set of D_1 predefined mass values and placed at D_2 their angular positions on balance planes. Thus, static and dynamic unbalanced rotor

have been analyzed using the number of masses N_1 and N_2 at corresponding angular positions placed on the balancing planes 1 and 2, respectively. Total numbers of design variables depend upon the number of masses. Lower and upper limits of design variables are considered as lower value and higher value of available discrete values. Moreover, the number of balancing masses N_i for the i^{th} plane is predefined by the user. Then, this balancing problem has $(D_1 \times D_2)^{\sum_{i=1}^2 N_i}$ number of possible combinations. For example, $D_1 = 8$, $D_2 = 10$ and $N_1 = N_2 = 2$ two masses per plane, there are more than 4×10^7 possible combinations. The identification of the best combination of masses and corresponding angular positions is complicated and time-consuming due to the high number of possible combinations. Therefore, the modified Jaya algorithm explained in section 5.2 is applied to solve the formulated discrete optimization problem. This algorithm is coded in MATLAB 2014. The effectiveness of the algorithm is compared with GA for the same problems using three cases as case I ($N_1 = N_2 = 1$), case II ($N_1 = N_2 = 2$), and case III ($N_1 = N_2 = 3$). The parameters of GA and modified Jaya algorithm for optimization procedure are shown in Table 5.3. However, GA requires its control parameters whereas Jaya is a parameter-less technique. In the case of GA, the population size in three cases is taken as 50, 150, and 300 while the number of iterations are taken as 400, 1600, and 4000, respectively for unbalanced problems. 20 independent runs of GA have been carried out to find out the best value of the objective function corresponding to the design variables. The function evaluations for the three cases are 4×10^5 , 4.8×10^6 and 2.4×10^7 , respectively. However, the modified Jaya algorithm takes population size for three cases as 20, 100 and 200 while the number of iterations are taken as 100, 500 and 3000, respectively for unbalanced problems. 20 independent runs of Jaya algorithm have been carried out to find out the best value of the objective function corresponding to the design variables as shown in Figs. 5.2 and 5.3. The function evaluations for the three cases are 2×10^5 , 2×10^6 and 1.5×10^7 , respectively. The function evaluations of modified Jaya algorithm for three cases are compared with those of GA as presented in Table 5.3. Moreover, Jaya algorithm requires 95%, 58% and 38% fewer function evaluations for all three cases than those required by GA. Hence, the computational efficiency of the modified Jaya algorithm is better than the GA algorithm.

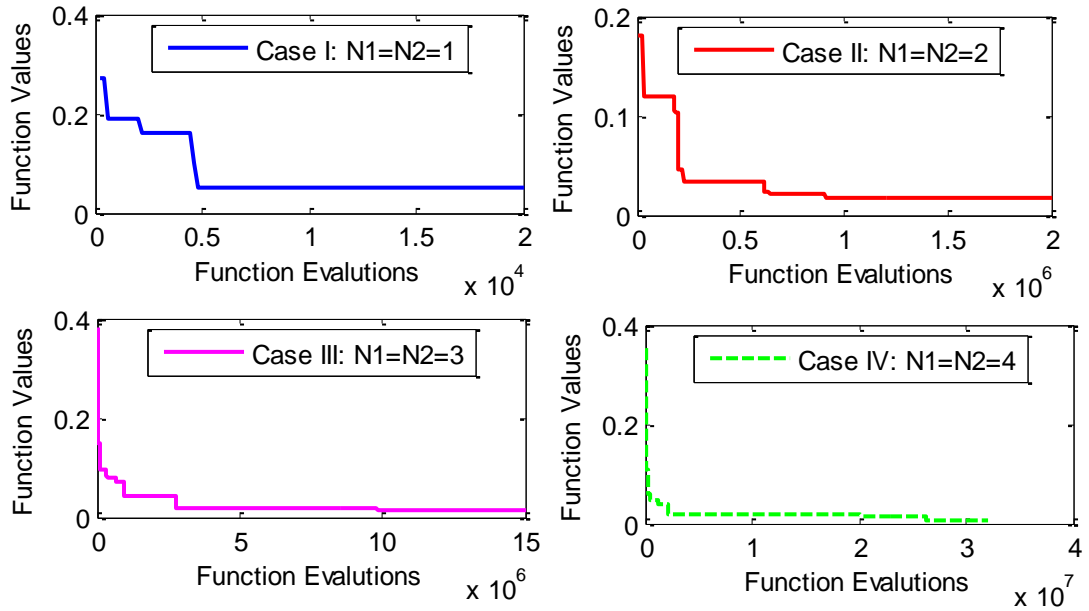


Fig. 5.2. Convergence of the best objective function values in different cases of static balancing

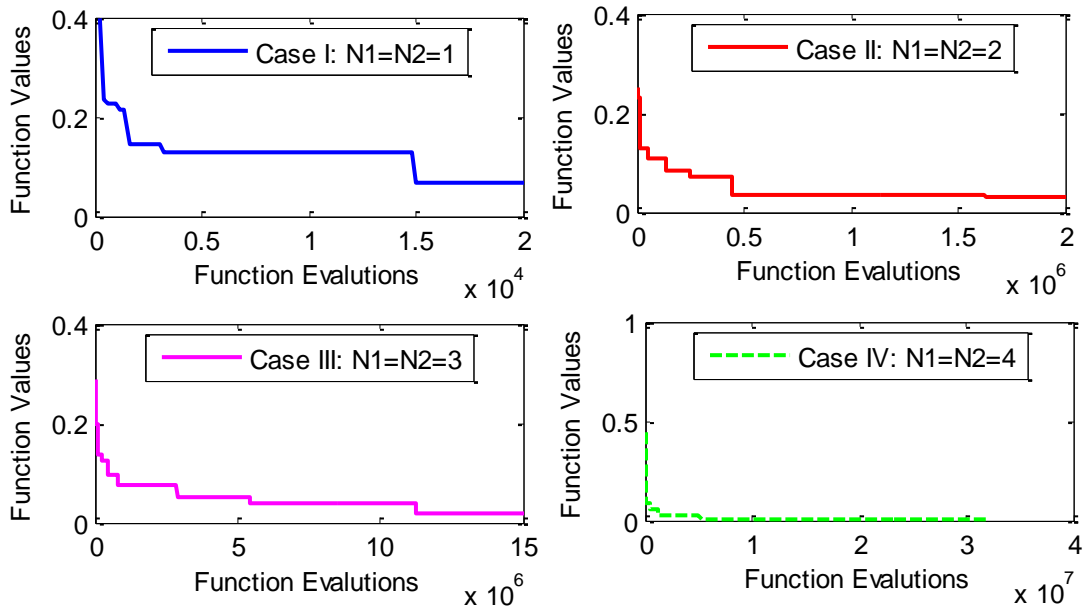


Fig. 5.3. Convergence of the best objective function values in different cases of dynamic balancing

Further, the number of masses ($N_1 = N_2 = 4$) as a case IV is analyzed for the same problems of static and dynamic unbalance. Optimization parameters for this case are taken as the population size of 300 and 3000 number of iterations. 20 independent runs of Jaya algorithm have been carried out to find out the best value of the objective function corresponding to the design variables. The best value of the objective function

is compared with those of the previous cases. Thus, the reaction forces acting on the support are further reduced for static and dynamic balancing problems. Hence, the optimal solutions depend on the number of masses, so reaction forces decrease as the numbers of balance mass increase. The convergence to the optimum solution for static and dynamic balancing for all cases is shown by the plots between function values and function evaluations in Figs.5.2 and 5.3, respectively and optimum discrete balancing solutions for all cases of static and dynamic balancing are presented in Table 5.4.

Using optimum values of design variables obtained by modified Jaya algorithm, the reaction forces at supports are analyzed. The reaction forces at supports P and Q are reduced by 70.85% and 65.70 % for case I, 92.13% and 93.62 % for case II, 98.76% and 99.69% for case III, and, 99.85% and 99.56% for case IV, respectively, correspond to those of the static unbalanced rotor. Similarly, 72% and 80 % for case I, 92.7% and 93.6 for case II, 99.3% and 98.7% for case III, and, 99.4% and 99% for case IV, respectively, correspond to those of the dynamic unbalanced rotor as shown in Table 5.4.

Table 5.3. Parameters of GA and Discrete Jaya algorithm for numerical problems

Algorithm	Number of masses	Population size	Number of Iteration	Number of Function Evaluations	Possible solutions
GA	$N_1 = N_2 = 1$	50	400	4×10^5	7056
	$N_1 = N_2 = 2$	150	1600	4.8×10^6	49.8×10^6
	$N_1 = N_2 = 3$	300	4000	2.4×10^7	3.51×10^{11}
Modified Jaya	$N_1 = N_2 = 1$	20	100	2×10^4 (-95%)	7056
	$N_1 = N_2 = 2$	100	500	2×10^6 (-58%)	49.8×10^6
	$N_1 = N_2 = 3$	200	3000	1.5×10^7 (-38%)	3.51×10^{11}
	$N_1 = N_2 = 4$	300	3000	3.2×10^7	2.48×10^{15}

Using optimum values of design variables obtained by modified Jaya algorithm, the reaction forces at supports are analyzed. The reaction forces at supports P and Q are reduced by 70.85% and 65.70 % for case I, 92.13% and 93.62 % for case II, 98.76% and 99.69% for case III, and, 99.85% and 99.56% for case IV, respectively, correspond to those of the static unbalanced rotor. Similarly, 72% and 80 % for case I, 92.7% and 93.6 for case II, 99.3% and 98.7% for case III, and, 99.4% and 99% for case IV,

respectively, correspond to those of the dynamic unbalanced rotor as shown in Table 5.4.

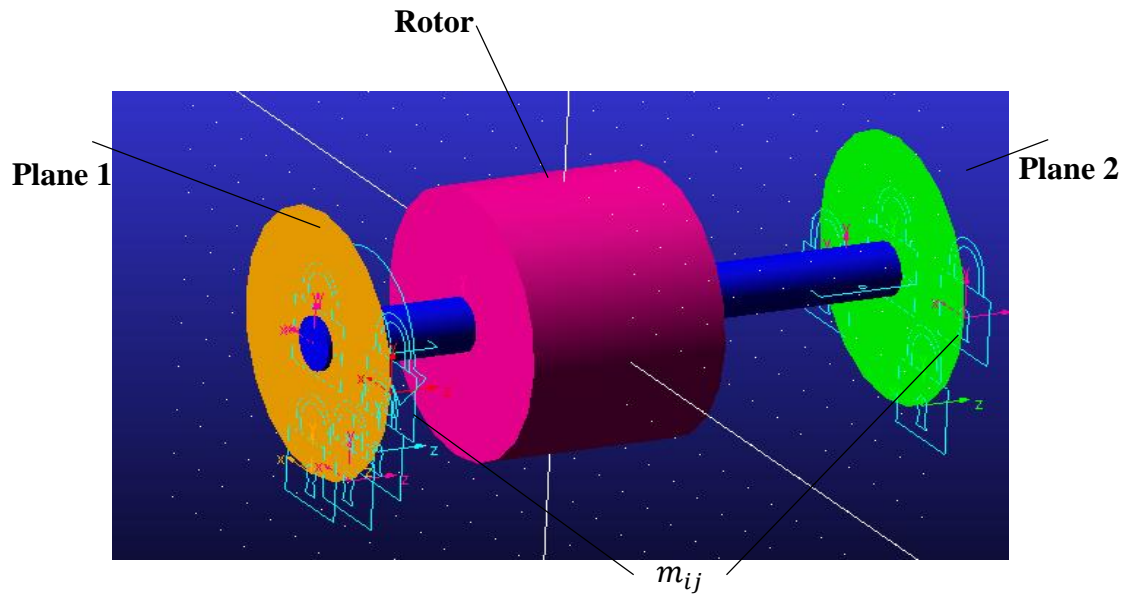


Fig. 5.4. Modeling of the rigid rotor in MSC ADAMS

A static and dynamic unbalance rigid rotor is modeled and simulated in MSC ADAMS using optimum values of design variables for all cases as shown in Fig. 5.4. ADAMS is a multi-body dynamic analysis program that is ideal for modeling of a rigid body.

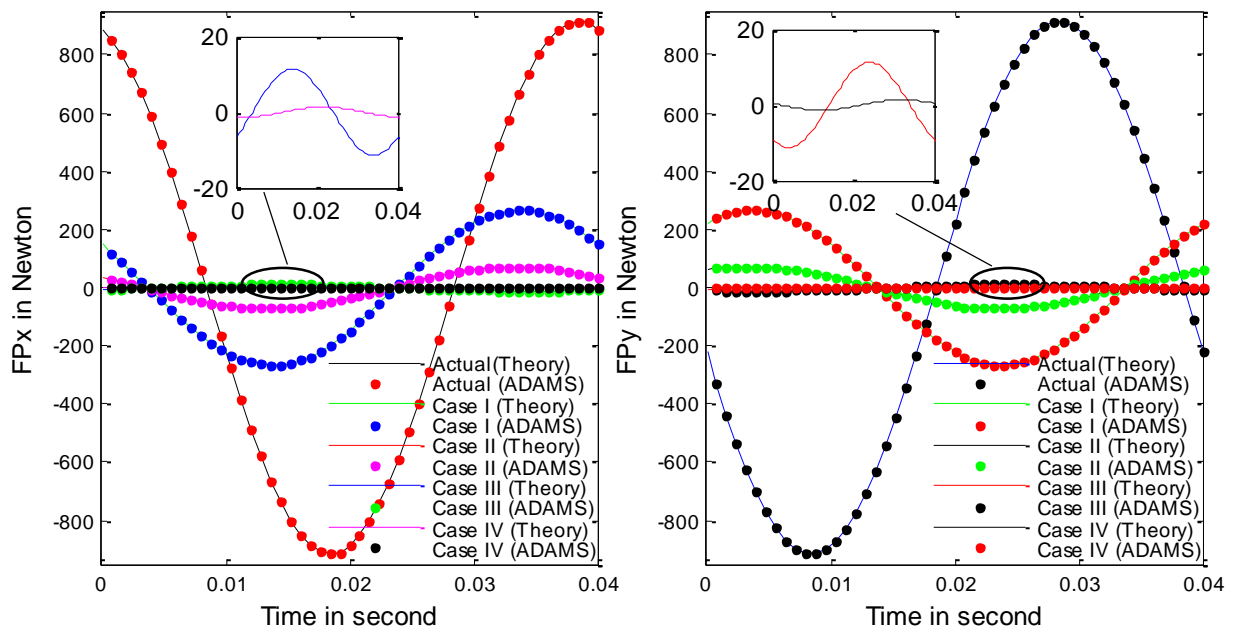


Fig. 5.5. Validation of reaction forces in x and y -directions at supports P using ADAMS for static balancing

Table 5.4. Comparison of modified Jaya algorithm to GA (Messenger and Pyrz, 2013) for all cases

Algorithm	Number of masses		Plane1		Plane2		F _P (N)	F _Q (N)
	N ₁	N ₂	m _{1j} (grams)	α _{1j} (degree)	m _{2j} (grams)	α _{2j} (degree)		
Static unbalance							915.6	610.4
GA	1	1	200	180	200	180	266.89	209.36
	2	2	200	180	200	180	72.03	38.97
			50	240	50	300		
	3	3	200	180	200	180	11.32	1.879
50			240	20	240			
		20	240	50	330			
Modified Jaya	1	1	200	180	200	180	266.89 (-70.85%)	209.36 (-65.70%)
	2	2	200	180	50	300	72.0 (-92.13%)	38.97 (-93.62%)
			50	240	200	240		
	3	3	50	240	50	330	11.32 (-98.76%)	1.879 (-99.69%)
			200	180	200	180		
			20	240	20	240		
	4	4	10	210	50	330	1.370 (-99.85%)	2.673 (-99.56%)
			200	180	10	240		
10			270	10	240			
		50	240	200	180			
Dynamic unbalance							742.3	1017.3
GA	1	1	200	150	300	210	208.01	204.15
	2	2	100	150	50	360	15.16	13.11
			100	180	300	210		
	3	3	200	150	20	120	4.473	13.14
50			240	300	180			
		10	330	200	300			
Modified Jaya	1	1	200	150	300	210	208.01 (-72%)	204.15 (-80%)
	2	2	100	180	300	210	15.16 (-92.7%)	13.11 (-93.6%)
			100	150	50	360		
	3	3	200	150	20	120	4.473 (-99.3%)	13.14 (-98.7%)
			50	240	200	300		
			10	330	300	180		
	4	4	20	360	10	150	4.042 (-99.4 %)	10.01 (-99%)
			10	270	300	210		
100			300	50	360			
		300	150	10	300			

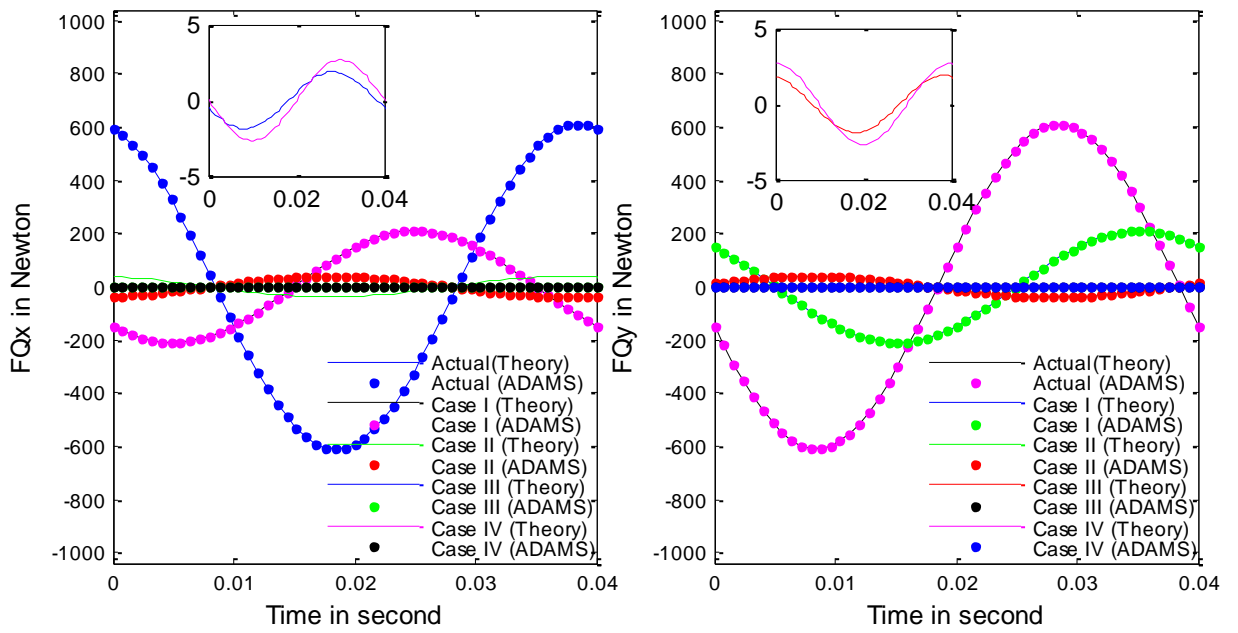


Fig. 5.6. Validation of reaction forces in x and y -directions at supports Q using ADAMS for static balancing

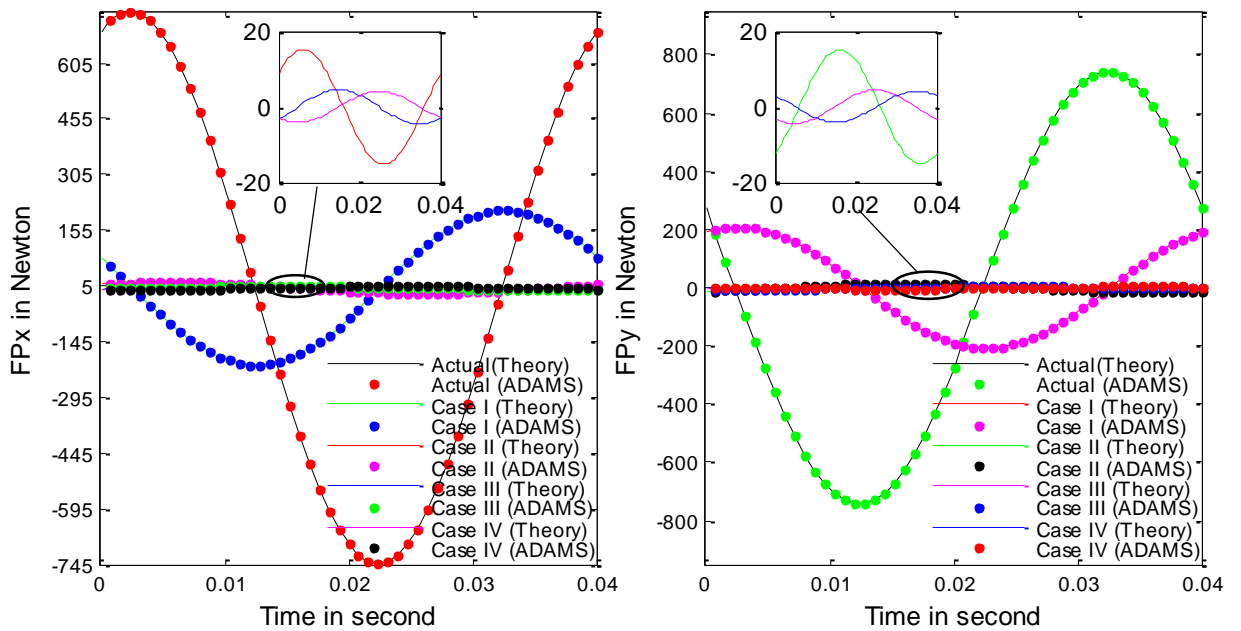


Fig. 5.7. Validation of reaction forces in x and y -directions at supports P using ADAMS for dynamic balancing

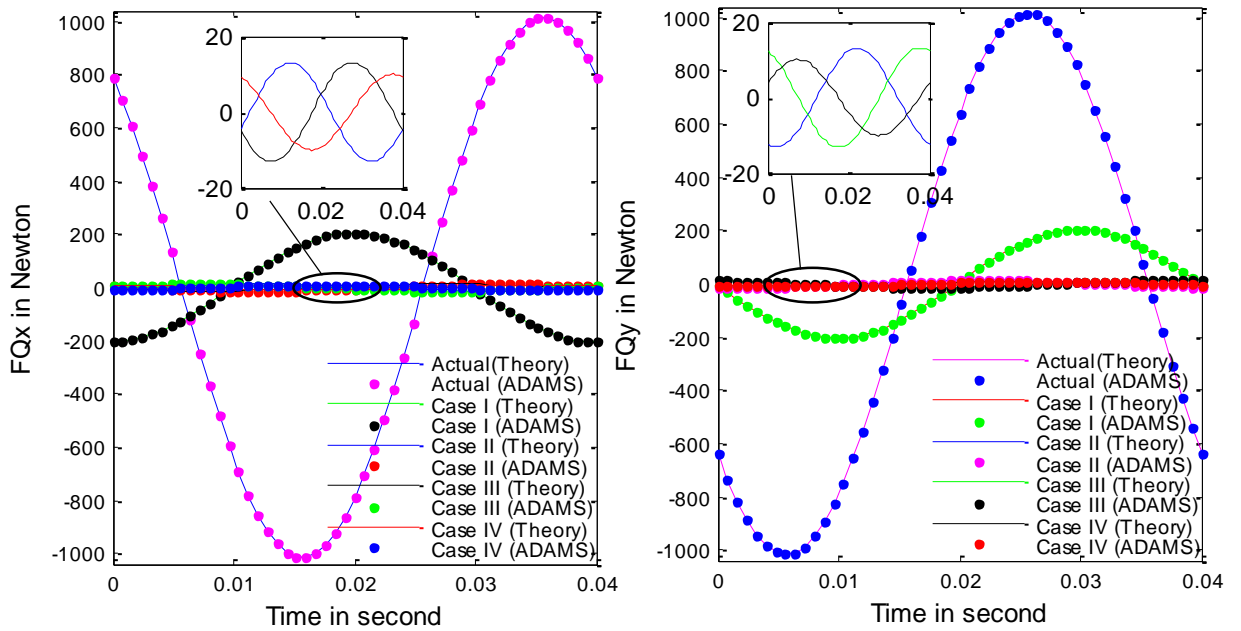


Fig. 5.8. Validation of reaction forces in x and y -directions at supports Q using ADAMS for dynamic balancing

Moreover, the reaction forces in x and y directions at supports P and Q as F_{Px} , F_{Py} , F_{Qx} , and F_{Qy} are validated using MSC ADAMS software for all cases in static and dynamic balancing as shown in Figs. 5.5, 5.6, 5.7, and 5.8, respectively. Figures 5.5 and 5.6 show that the values of F_{Px} , F_{Py} , F_{Qx} , and F_{Qy} for the unbalanced rotor (actual case), case I, case II, case III, and case IV obtained from theoretical model match precisely to those obtained from simulation with MSC ADAMS software, respectively, for static balancing. It is observed that the maximum values of F_{Px} and F_{Py} are reduced up to 70.85% and 70.85% for case I, 92.13% and 92.13% for case II, 98.76% and 98.76% for case III, and, 99.85% and 99.85% for case IV, respectively, correspond to those of the static unbalanced rotor as shown in Fig.5.5. While the reduction in maximum values of F_{Qx} and F_{Qy} is 65.70 % and 65.70 % for case I, 93.62 % and 93.62 % for case II, 99.69% and 99.69% for case III, and, 99.56% and 99.56% for case IV, respectively, correspond to those of the static unbalanced rotor as shown in Fig.5.6. Similarly, Figures 5.7 and 5.8 present that the values of F_{Px} , F_{Py} , F_{Qx} , and F_{Qy} for all cases obtained from simulation with MSC ADAMS software and theoretical model are in good agreement for dynamic balancing, respectively, and the maximum values of F_{Px} and F_{Py} are reduced by 72% and 72% for case I, 92.7% and 92.7% for case II,

99.3% and 99.3% for case III, and, 99.4% and 99.4% for case IV, respectively, correspond to those of the dynamic unbalanced rotor as shown in Fig.5.7, while the maximum values of F_{Qx} and F_{Qy} are reduced up to 80 % and 80 % for case I, 93.6 % and 93.6 % for case II, 98.7% and 98.7% for case III, and 99 % and 99 % for case IV, respectively, correspond to those of the dynamic unbalanced rotor as shown in Fig.5.8. Hence, theoretical and simulation results show that the proposed methodology gives a certain decrease in reaction forces at supports.

5.3.2. Application - Balancing of the threshing drum

The methodology developed in this chapter is applied for balancing of the threshing drum (Prashad and Sharma, 1985). The dimension of the threshing drum is presented in Table 5.5. It rotates at a constant speed of 460 rpm about the z-axis and has a total mass of 120 kg.

Table 5.5. Dimensions of the threshing drum in (m)

l_p	l_Q	l_1	l_2	R_1	R_2
-0.48	0.44	-0.35	0.33	0.33	0.33

Balance masses and corresponding angular positions are chosen from the set $D_1 = 7$ and $D_2 = 18$, respectively, given as

$$m_{ij}(\text{grams}) \in [0; 10; 20; 50; 100; 200; 300],$$

$$\alpha_{ij}(\text{Degrees}) \in [20; 20; 360]$$

The threshing drum is balanced statically by placing the number of masses N_1 and N_2 at corresponding angular positions on the balancing planes 1 and 2, respectively. The eccentricity values (e_x, e_y) in mm and products of inertia (I_{xz}, I_{yz}) in $\text{kg} - \text{m}^2$ for the threshing drum are given as $(-0.154, 1.12)$ and $(0,0)$, respectively.

A modified Jaya and GA algorithms are used to find the optimal balancing masses and corresponding angular positions on two planes for three different cases of masses as case 1 ($N_1 = N_2 = 1$), case2 ($N_1 = N_2 = 2$), and case 3($N_1 = N_2 = 3$). In case of GA, the population size in three cases is taken as 20, 150, and 500 while the number of iterations is chosen as 100, 100, and 3000, respectively for this unbalanced problem. Thus, GA takes 2000, 15×10^3 and 15×10^5 function evaluations for three cases, respectively to find the optimal results. While, modified Jaya algorithm takes population size of 10, 100 and 300, and the number of iterations similar to 100, 100 and 3000 for all three cases, respectively. Hence, the number of function evaluations for three cases equal to 1×10^3 , 10×10^3 and 9×10^5 , respectively, are required by the

modified Jaya algorithm. The convergence of the best objective function values in both algorithms is shown in Fig.5.9. Figure 5.9 indicates that the modified Jaya algorithm provides better optimal solutions by taking 50%, 33.33% and 50% fewer number of function evaluations for case 1, case 2 and case 3, respectively, than those of GA.

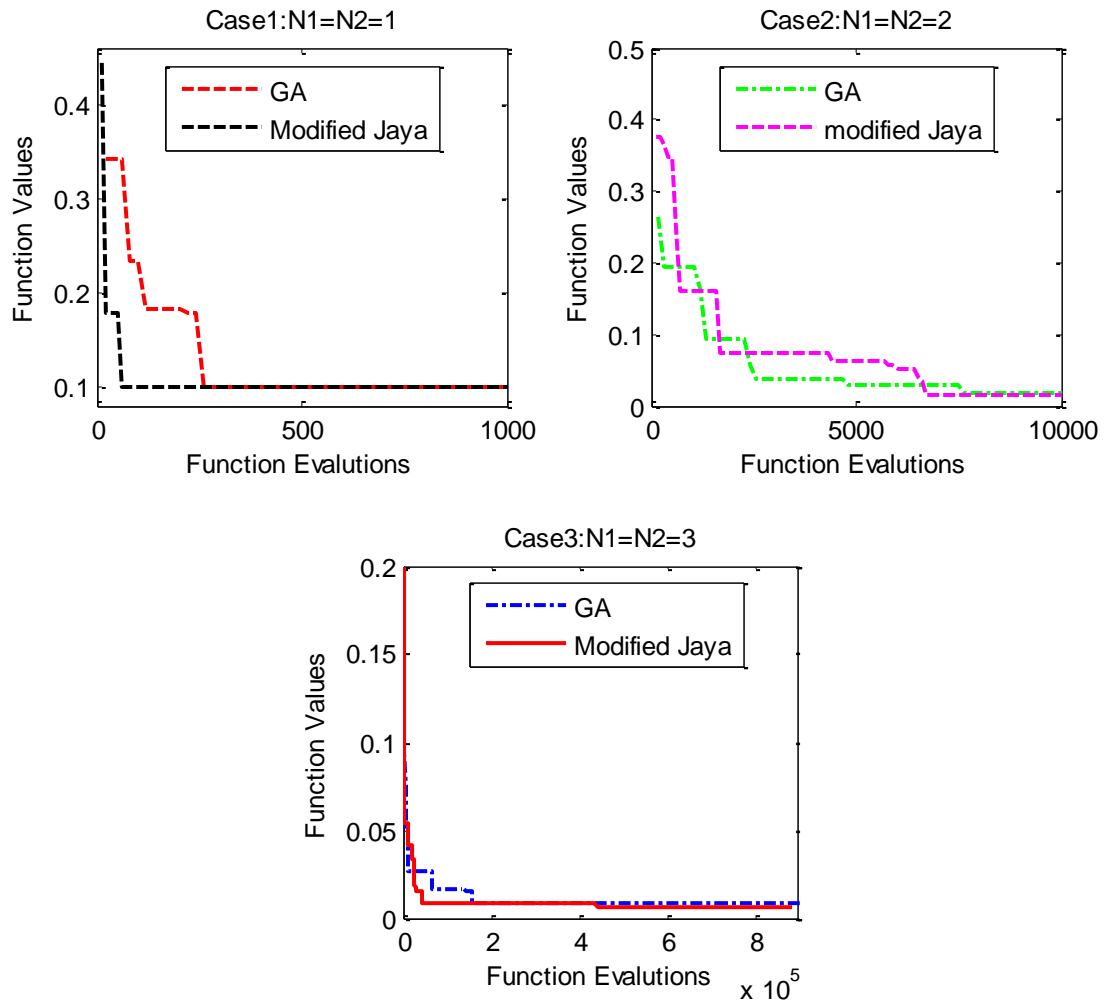


Fig. 5.9. Convergence of the best objective function values in different cases for GA and modified Jaya algorithm

Using optimum values of design variables obtained by discrete Jaya algorithm, the reaction forces at supports are analyzed. The reaction forces at supports P and Q are reduced by 89.88% and 89.87 % for case 1, 98.34% and 98.37 % for case 2, 99.45% and 99.27% for case 3, respectively, correspond to those of the unbalanced drum (Original case) as shown in Table 5.6. Moreover, the variation of the reaction forces in x and y directions at supports P and Q as F_{Px} , F_{Py} , F_{Qx} , and F_{Qy} for all cases are shown in Figs. 5.10 and 5.11, respectively. Figure 5.10 shows that the maximum values of F_{Px}

and F_{Py} are reduced up to 89.86% and 89.86% for case 1, 98.34% and 98.34% for case 2, and, 99.45% and 98.46% for case 3, respectively likened with those of the unbalanced drum. While the reduction in maximum values of F_{Qx} and F_{Qy} is 89.87 % and 89.87 % for case 1, 98.37 % and 98.36 % for case 2, and, 99.26% and 99.27% for case 3, respectively compared with those of the unbalanced drum as shown in Fig.5.11.

Table 5.6. Comparison of modified Jaya algorithm to GA for all cases

Algorithm	Number of masses		Plane1		Plane2		F_p (N)	F_Q (N)
	N_1	N_2	m_{1j} (grams)	α_{1j} (degree)	m_{2j} (grams)	α_{2j} (degree)		
Unbalance drum							161.11	175.77
GA	1	1	200	260	200	260	16.31	17.79
	2	2	200	260	20	280	2.63	2.87
			20	280	200	260		
	3	3	200	260	10	320	1.19	1.26
20			260	200	260			
Modified Jaya	1	1	200	260	200	260	16.31 (-89.88%)	17.79 (-89.87%)
	2	2	200	260	20	280	2.63 (-98.34%)	2.87 (-98.37%)
			20	280	200	260		
	3	3	200	260	200	260	0.88 (-99.45%)	1.28 (-99.27%)
10			360	20	260			

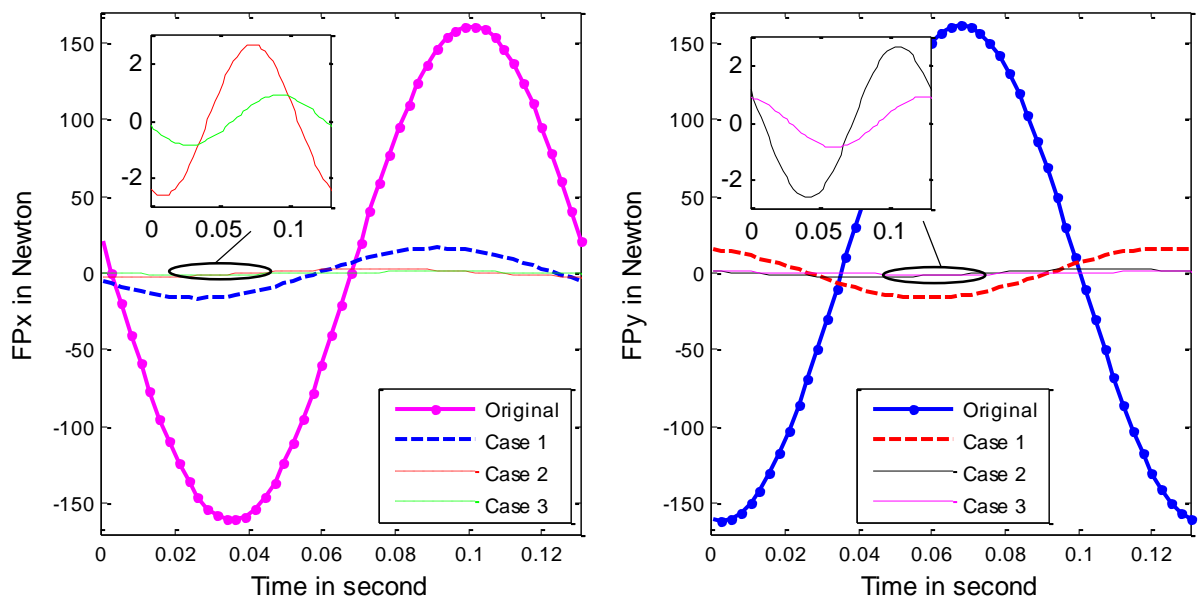


Fig. 5.10. Variation of reaction forces in x and y -directions at support P for the balancing of the threshing drum

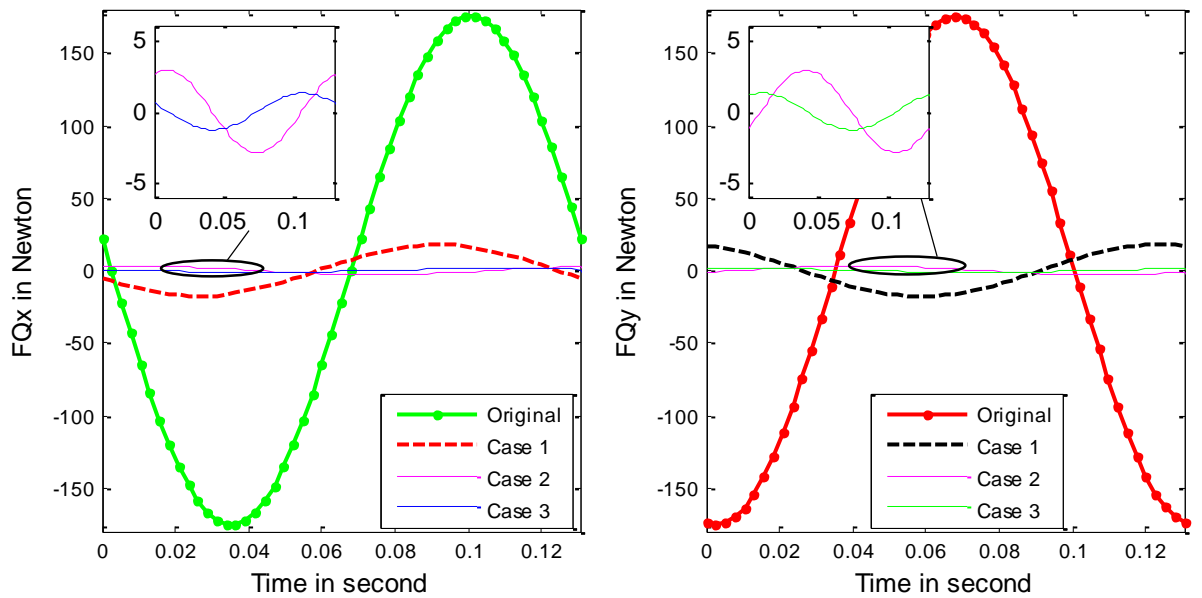


Fig. 5.11. Variation of reaction forces in x and y -directions at support Q for the balancing of the threshing drum

An unbalance rigid threshing drum is also modeled and simulated in MSC ADAMS using optimum values of design variables for all cases. ADAMS is a multi-body dynamic analysis program that is ideal for modeling of a rigid body (ADAMS, 2014). Results obtained from the theoretical model match precisely to those obtained from simulation with MSC ADAMS software. Hence, the proposed methodology gives a certain decrease in reaction forces at supports.

5.3.2.1. Experimental study

This section presents the robustness and effectiveness of the developed approach evaluated using the experimental study of the threshing drum. The experimental study is held at the thresher manufacturer workshop (Prashad and Sharma, 1985). Then, a smart field balancer as shown in Fig.5.12 consists of two accelerometers for vibration measurement and optical KeyPhasor for angular position and speed measurement, manufactured by Schenck Rotech India limited is used to measure the balancing masses at corresponding angular positions on the two balance planes for an unbalanced threshing drum. Threshing drum of 120 kg mass supported at two bearings and is rotated at 460 rpm by v-belt pulley through PTO shaft that runs by Massey Ferguson 245 DI. Two balancing planes are considered at the threshing drum. The balancing radius is taken as 0.33 m.

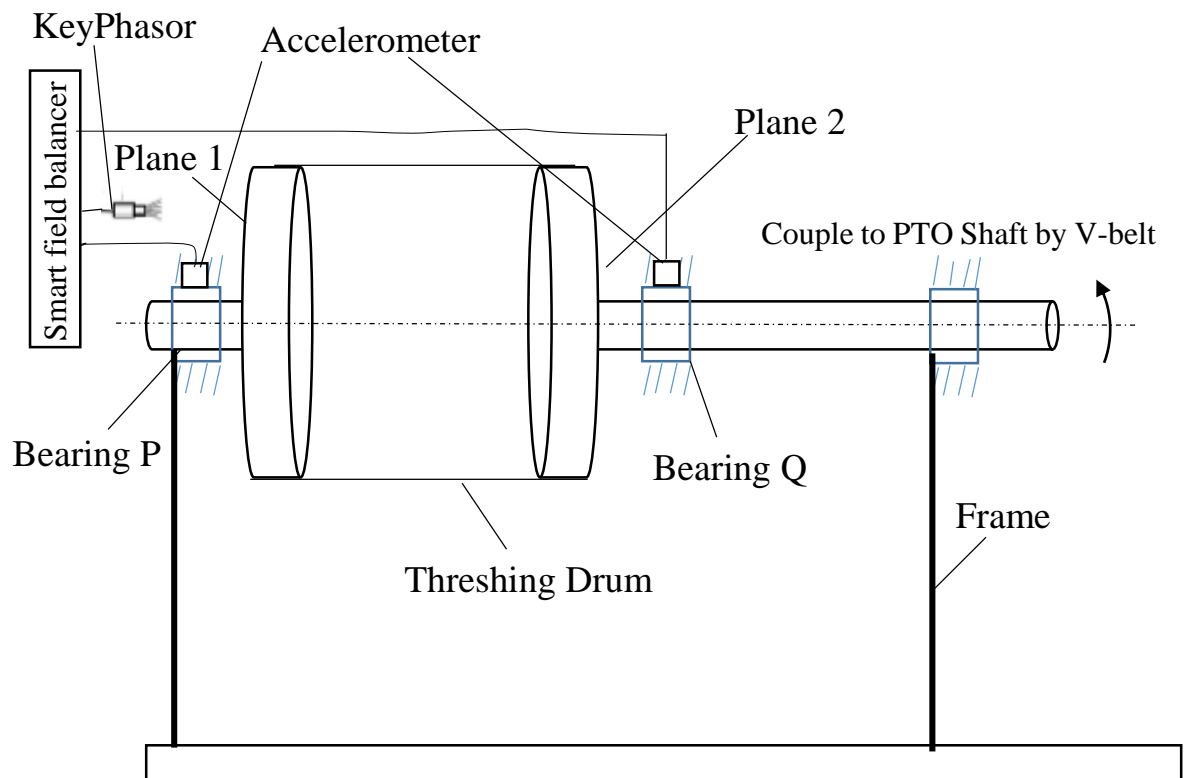


Fig. 5.12. The experiment setup for the unbalanced threshing drum

At starting, two accelerometers are attached to the bearings P and Q as presented in Fig.5.12. These measure vibration amplitudes at bearings P and Q as $V_{P,rms}$ and $V_{Q,rms}$ and, optical key-phasor measures the angular position of the balance mass and the angular speed of the unbalanced drum. Accelerometer and optical KeyPhasor are linked to the smart field balancer. This balancer indicates vibration amplitudes at bearings P

and Q as $V_{P,rms}$ and $V_{Q,rms}$ and the continuous balancing values for unbalanced drum such as $\bar{m}_1, \bar{\alpha}_1, \bar{m}_2,$ and $\bar{\alpha}_2$. Initially, unbalanced threshing rotates at its balance speed of 460 rpm. The smart field balancer provides the initial vibration amplitudes $V_{P0,rms} = 3.109$ and $V_{Q0,rms}=4.785$ (in mm/s). After that, trial weights 400 gram at 0° are applied at both planes 1 and 2 for the calibration of the smart field balancer. Finally, this balancer gives the continuous balancing solutions as $\bar{m}_1 = 170.2$ (gram), $\bar{\alpha}_1 = 70^\circ$, $\bar{m}_2 = 292.2$ (gram), and $\bar{\alpha}_2 = 113^\circ$ and vibration amplitudes at bearing P and Q as $V_{P,rms} = 0.685$ (mm/s) and $V_{Q,rms} = 0.545$ (mm/s), respectively.

In the analysis of this practical problem, a discrete multi-objective optimization problem with minimization of the difference between the balancing components ($m_{ij}R_i$) and corresponding continuous balancing components (\bar{m}_iR_i) in x and y directions, respectively, is formulated. The masses and corresponding angular position per plane are taken as design variables, and are expressed in Eq. (5.11). The objective functions are expressed as

$$\begin{aligned} \text{Minimize } f_1(\mathbf{x}) &= \sum_{i=1}^2 \sum_{j=1}^{N_i} |(m_{ij}R_i \cos \alpha_{ij} - \bar{m}_i R_i \cos \bar{\alpha}_i)| \\ \text{Minimize } f_2(\mathbf{x}) &= \sum_{i=1}^2 \sum_{j=1}^{N_i} |(m_{ij}R_i \sin \alpha_{ij} - \bar{m}_i R_i \sin \bar{\alpha}_i)| \end{aligned} \quad (5.16)$$

Where, $i = 1, 2$ and $j = 1, 2 \dots N_i$

The number of objective functions transforms into a single function using the weighting factors (Marler and Arora, 2004, 2010). Finally, optimization problem is formulated as:

$$\text{Minimize } f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) \quad (5.17)$$

Where weighting factors w_1 and w_2 represent the relative importance of the various objectives. However, both the objective functions have equal importance. Therefore, $w_1 = 0.5$ and $w_2 = 0.5$ are chosen for this practical problem.

Discrete constraints to design variables are defined by Eq. (5.14). Total discrete values of masses and corresponding angular positions are chosen from given discrete sets

$$m_{ij}(\text{grams}) \in [0; 10; 20; 50; 100; 200; 300]$$

$$\alpha_{ij}(\text{Degrees}) \in [20; 20; 360]$$

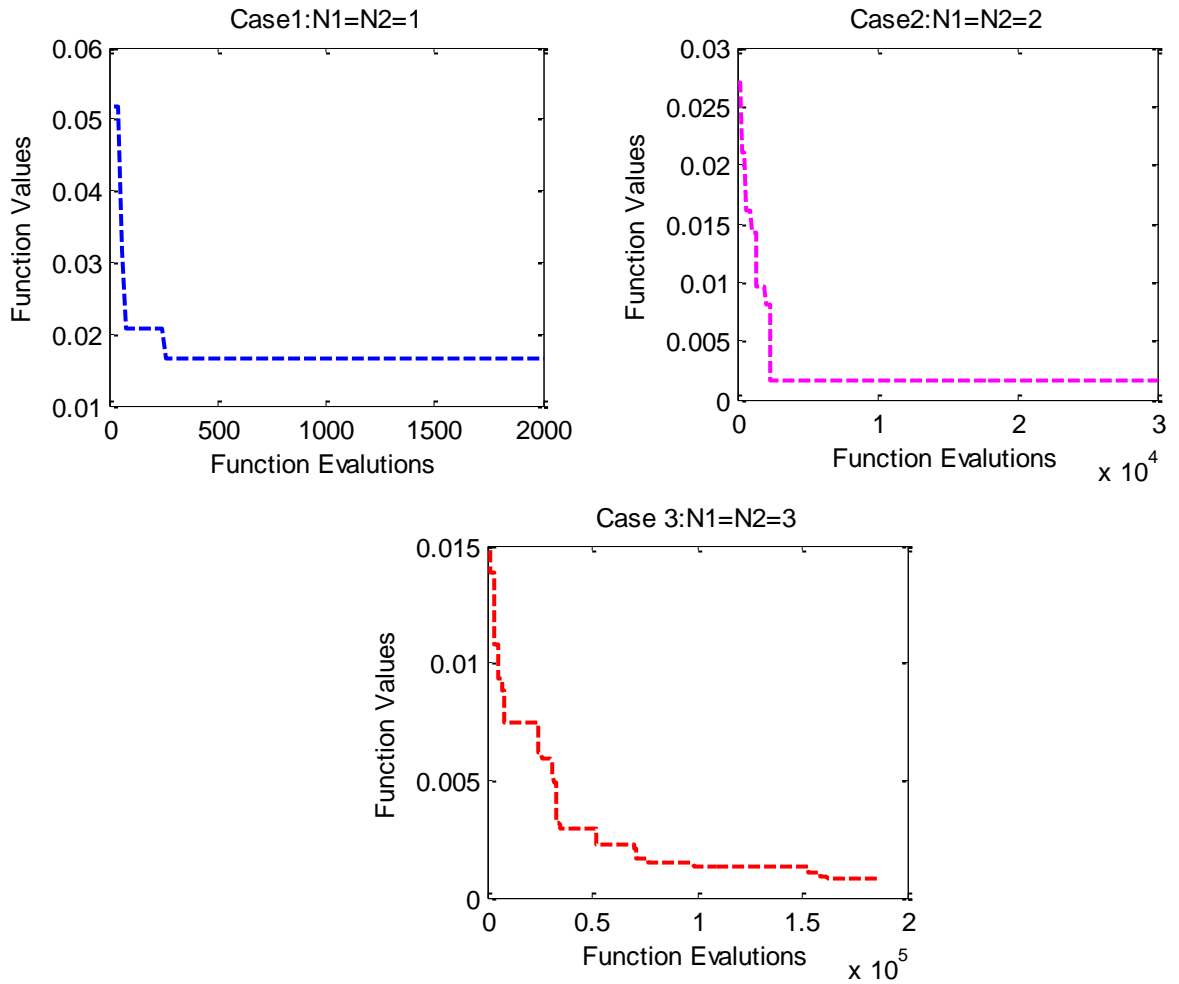


Fig. 5.13. Convergence of the best objective function values in different cases

Table 5.7. Optimal solutions for all cases

Number of masses		Plane1		Plane2		$f(x)$	$V_{P,rms}$ (mm/s)	$V_{Q,rms}$ (mm/s)
N_1	N_2	m_{1j} (grams)	α_{1j} (degree)	m_{2j} (grams)	α_{2j} (degree)			
Unbalanced Drum							3.109	4.875
1	1	0.2	60	0.3	120	0.01650	0.542 (-82.57)	0.457 (-90.63)
2	2	0.1	100	0.2	120	0.00161	0.319 (-89.74%)	0.265 (-94.56)
		0.1	40	0.1	100			
3	3	0.02	40	0.02	360	0.00088	0.179 (-94.24%)	0.164 (-96.64)
		0.05	280	0.02	40			
		0.2	80	0.3	120			

It is observed that the modified Jaya algorithm is computationally more efficient than GA. Therefore, the modified Jaya algorithm is applied to solve the formulated optimization problem using different cases of masses as case 1 ($N_1 = N_2 = 1$), case2 ($N_1 = N_2 = 2$), and case 3($N_1 = N_2 = 3$). To find the optimal solutions, it takes the population size of 20, 100 and 300, and the number of iterations similar to 100, 100 and 2500 for all three cases, respectively. Hence, the number of function evaluations for three cases equal to 2×10^3 , 1×10^4 and $75. \times 10^5$, respectively, are taken by this algorithm. Convergence of the best objective function values is shown in Fig.5.13. The optimum solutions for all cases are presented in Table 5.7.

For all cases, the optimum values of discrete masses at the corresponding angular position are placed at the balance planes 1 and 2 of the threshing drum. Smart field balancer is used to measure vibration amplitudes of the bearings P and Q as $V_{P,rms}$ and $V_{Q,rms}$ (in mm/s), respectively. The vibration amplitudes of bearings P and Q are reduced up to 82.57% and 90.63% for case 1, 89.74% and 94.56% for case 2, and 94.62% and 96.24% for case 3, respectively with respect to those of the unbalanced thresher drum as presented in Table 5.7. Hence, the number of masses per plane improves the vibration amplitudes of the threshing drum. As a result of that, the fuel consumption and dynamic performance of machines will improve and also human accident will minimize.

5.4. Summary

In this chapter, optimum two-plane discrete balancing procedure is developed for the rigid rotor. The multi-objective problem is formulated to minimize the reaction forces at supports, and discrete parameters such as masses and corresponding angular positions on each balancing plane are considered as design variables. The effectiveness of the proposed approach is demonstrated by a numerical problem taken from literature, and it is also applied to the threshing drum model of the thresher machine. The proposed modified Jaya algorithm and GA are used as solvers for these balancing problems. It is observed that a modified Jaya algorithm takes fewer function evaluations and gives the better and nearly close results than those of GA for all cases. The reaction forces at supports P and Q are reduced up to 70.85% and 65.70 % for case I, 92.13% and 93.62 % for case II, 98.76% and 99.69% for case III, and 99.85% and 99.56% for case IV corresponding to the static unbalanced rotor, respectively.

Similarly, 72% and 80 % for case I, 92.7% and 93.6 for case II, 99.3% and 98.7% for case III, and 99.4% and 99% for case IV correspond to the dynamic unbalanced rotor, respectively. Hence the reaction forces at supports decrease, as the number of balance mass per plane increases.

In case of the threshing drum balancing problem, reaction forces at supports P and Q are reduced by 65% and 87.32 % for case 1, 98.28% and 98.15 % for case 2, 99.78% and 98.74% for case 3, respectively, correspond to those of the unbalanced threshing drum.

The unbalanced and balanced rotors are simulated using MSC ADAMS software. It is found that the simulation and theoretical results of the component forces are in good agreement without variation. Besides the analytical study of the threshing drum, the experimental study is also investigated. The experimental study shows that the number of discrete masses per plane decreases the vibration amplitudes of the unbalanced threshing drum. Thus the dynamic performance of the thresher machine can be improved. The proposed approach is quite general and equally applicable to the rigid and flexible rotors.

Optimal shape synthesis of the flywheel

This chapter presents the optimal shape synthesis procedure of the flywheel using a cubic B-spline curve. The flywheel plays a vital role in storing kinetic energy in modern machines. Thus, the kinetic energy is an essential parameter to measure flywheel performance and can be improved by optimal thickness distribution of the flywheel, generally known as shape optimization. Therefore, the shape optimization model of the flywheel with maximization of the kinetic energy is formulated using a cubic spline curve under the design constraints like the mass of the flywheel and maximum values of Von Mises stresses. A flow chart is proposed to solve the two-point boundary value differential equation for calculation of Von Mises stress at each point between the inner and outer radius of the flywheel. The control points of the cubic B-spline curve are taken as design variables. Then the formulated problem is solved using particle swarm algorithm (PSO), genetic algorithm (GA), and Jaya algorithm. The effectiveness of the proposed approach is investigated through the design of flywheel taken from the literature and the flywheel design of the thresher machine.

6.1. A shape optimization model of the flywheel

In this section, the thickness profile of the flywheel is represented by the cubic B-spline curve as shown in Fig.6.1. R_1 and R_2 are the inner radius and outer radius of the flywheel respectively. The flywheel is axial symmetric in X and Y directions, and thickness distribution along the radial direction is represented by cubic B-splines curve. The control points describe the vertices of the polygon of the B-spline curve. A set of control points as p_1, p_2, \dots, p_n approximates or interpolates the curve as defined in Eq. (6.1) (Mortenson, 2006; Zeid, 1991). X and Y coordinates of control points represent the radial and thicknesses points, respectively.

$$p(u) = \sum_{i=1}^n N_{i,k}(u)p_i, \quad 0 \leq u \leq S \quad (6.1)$$

Where parameter k controls the degree of curve and the value of k for a cubic B-spline curve is 4. $N_{i,k}(u)$'s are B-spline blending functions, which are defined by the following expression:

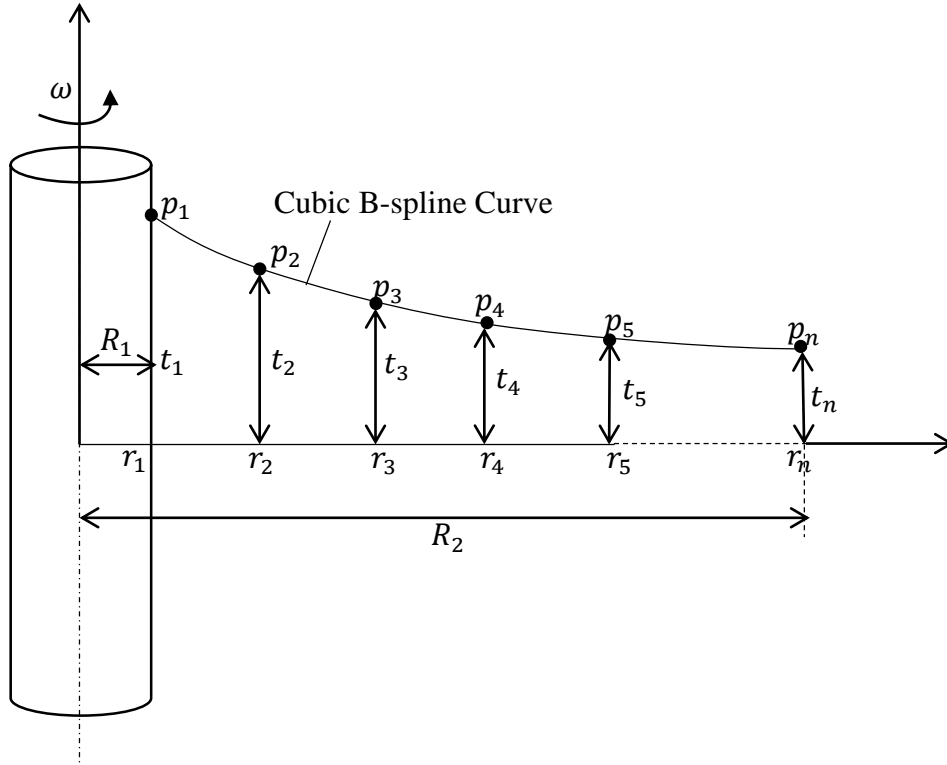


Fig. 6.1. 2D symmetric model of flywheel

Moreover, a composite sequence of curve segments ($S = n - k + 1$) defines the cubic B-spline curve that connected with C^2 continuity. Further, it blends two curve segments with the same curvature as shown in Fig.6.2.

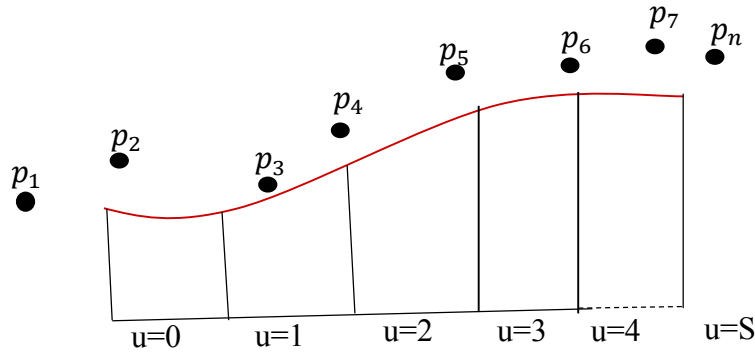


Fig. 6.2. B-spline curve segments

$$N_{i,k} = (u - u_i) \frac{N_{i,k-1}(u)}{u_{i+k-1} - u_i} + (u_{i+k} - u) \frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \quad (6.2)$$

$$N_{i,1} = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{elsewhere} \end{cases} \quad (6.3)$$

Where, $N_{i,1}$ is a unit step function and u_i are called the parametric knots. The values of u_i depend on whether the B-spline curve is an open (non-periodic) or periodic curve. The periodic B-spline curve does not pass through any of control points, while a non-

periodic B-spline curve passes through the two end points. The knot values for periodic and non-periodic B-spline curves are given by:

$$\text{For periodic B – spline curve } u_i = i - 1, \quad i - 1 \leq S \quad (6.4)$$

$$\text{For non – periodic B – spline } u_i = \begin{cases} 0, & i < k \\ i - k + 1, & k \leq i < n \\ n - k + 1, & i > n \end{cases} \quad (6.5)$$

Where $1 \leq i \leq n + k$

The coordinates of any point on the i^{th} segment of the curve for periodic B-spline curves are determined using Eqs. (6.1-6.4) and are given as:

$$r^i(u) = \frac{a_1 r_i + a_2 r_{i+1} + a_3 r_{i+2} + a_4 r_{i+3}}{6} \quad (6.6)$$

$$t^i(u) = \frac{a_1 t_i + a_2 t_{i+1} + a_3 t_{i+2} + a_4 t_{i+3}}{6} \quad (6.7)$$

Where

$$\left. \begin{aligned} a_1 &= -u^3 + 3u^2i - 3ui^2 + u^3 \\ a_2 &= 3u^3 + u^2(3 - 9i) + u(9i^2 - 6i - 3) - 3i^3 + 3i^2 + 3i + 1 \\ a_3 &= -3u^3 + u^2(9i - 6) + u(-9i^2 + 12i) + 3i^3 - 6i^2 + 4 \\ a_4 &= u^3 + u^2(-3i + 3) + u(3i^2 - 6i + 3) - i^3 + 3i^2 - 3i + 4 \end{aligned} \right\} \begin{array}{l} i - 1 \leq u \\ \leq i \end{array} \quad (6.8)$$

The coordinates of any point on the i^{th} segments of the curve for non-periodic B-spline curves are determined using Eqs. (6.1-6.3) and 6.5 and are given as:

$$r^i(u) = a_1^i r_i + a_2^i r_{i+1} + a_3^i r_{i+2} + a_4^i r_{i+3} \quad (6.9)$$

$$t^i(u) = a_1^i t_i + a_2^i t_{i+1} + a_3^i t_{i+2} + a_4^i t_{i+3} \quad (6.10)$$

Where a_1^i, a_2^i, a_3^i and a_4^i are obtained by solving Eqs. (6.2-6.3) and (6.5) for each segment in MAT LAB.

6.1.1. Parameters of the flywheel

The volume (v), mass (m), and kinetic energy (e_k) are the parameters of the flywheel, which also describe the performance of flywheel. These parameters, for each segment, are calculated using the expression of $r^i(u)$ and $t^i(u)$ of periodic B-spline as:

$$v^i = 2\pi\rho \int_{u_{i-1}}^{u_i} t^i(u)r^i(u) \frac{dr^i(u)}{du} du \quad (6.11)$$

$$m^i = 2\pi\rho \int_{u_{i-1}}^{u_i} t^i(u)r^i(u) \frac{dr^i(u)}{du} du \quad (6.12)$$

$$e_K^i = \pi \rho \omega^2 \int_{u_{i-1}}^{u_i} t^i(u) \left(r^i(u) \right)^3 \frac{dr^i(u)}{du} du \quad (6.13)$$

Where ρ is the density of the material and ω represents the angular velocity of the flywheel.

Further, the whole volume (V), mass (M), and kinetic energy (E_K) of flywheel are calculated by the summing of volume, mass, and kinetic energy of each segment, respectively as:

$$V = \sum_{i=1}^{n-k+1} v^i \quad (6.14)$$

$$M = \sum_{i=1}^{n-k+1} m^i \quad (6.15)$$

$$E_K = \sum_{i=1}^{n-k+1} e_K^i \quad (6.16)$$

6.1.2. Stress analysis of flywheel

A non-periodic uniform B-spline cubic curve is used for stress analysis of the flywheel. It passes through the end points. Stresses at the inner radius and outer radius are calculated using this curve. While a periodic uniform B-spline cubic curve does not pass through the endpoints, it does not give information about stresses at an inner and outer radius of the flywheel. However, tangential and radial stresses occur due to centrifugal forces during the operation of the flywheel. The following relationship between tangential and radial stresses for each segment has been obtained, based on the force balance on each small element of the flywheel (Timoshenko and Goodier, 1970).

$$\frac{d}{dr^i(u)} (t^i(u) r^i(u) \sigma_r^i) - t^i(u) \sigma_\theta^i + \rho (r^i(u))^2 t^i(u) \omega^2 = 0 \quad (6.17)$$

$$(\sigma_\theta^i - \sigma_r^i)(1 + \nu) + r \frac{d\sigma_\theta^i}{dr^i(u)} - r^i(u) \nu \frac{d\sigma_r^i}{dr^i(u)} = 0 \quad (6.18)$$

Where σ_r^i and σ_θ^i are tangential and radial stresses for each segment, respectively. $t^i(u)$ and $r^i(u)$ are the thickness and radius of the flywheel for each segment given in Eqs. (6.9) and (6.10) for non-periodic B-spline.

Let us define a stress function F^i for each segment as:

$$F^i = t^i(u) r^i(u) \sigma_r^i \quad (6.19)$$

$$\sigma_r^i = \frac{F^i}{t^i(u) r^i(u)} \quad (6.20)$$

Substituting Eq. (6.20) in Eq. (6.17) and solving for σ_θ as

$$\sigma_\theta^i = \frac{1}{t^i(u)} \left(\frac{dF^i}{dr^i(u)} + \rho(r^i(u))^2 \omega^2 t^i(u) \right) \quad (6.21)$$

A second ordinary differential equation is obtained by solving the Eqs. (6.18), (6.20) and (6.21) as:

$$r^2 \frac{d^2 F}{dr^2} + r \frac{dF}{dr} - F + (3 + \nu) \rho r^3 \omega^2 t - \frac{r}{t} \frac{dt}{dr} \left(r \frac{dF}{dr} - \nu F \right) = 0 \quad (6.22)$$

Thus, Eqs. (6.21) and (6.22) are written in parametric form for each segment by converting the independent variable r into u by using chain rule of differentiation as

$$\sigma_\theta^i = \frac{1}{t^i(u)} \left(\frac{\frac{dF^i}{du}}{\frac{dr^i(u)}{du}} + \rho(r^i(u))^2 \omega^2 t^i(u) \right) \quad (6.23)$$

$$\begin{aligned} & (r^i(u))^2 \frac{dr^i(u)}{du} \frac{d^2 F^i}{du^2} \\ & + \left\{ r^i(u) \left(\frac{dr^i(u)}{du} \right)^2 - (r^i(u))^2 \frac{d^2 r^i(u)}{du^2} \right. \\ & \left. - \frac{(r^i(u))^2}{t^i(u)} \frac{dr^i(u)}{du} \frac{dt^i(u)}{du} \right\} \frac{dF^i}{du} \\ & + \left\{ \nu \frac{r^i(u)}{t^i(u)} \frac{dt^i(u)}{du} \left(\frac{dr^i(u)}{du} \right)^2 - \left(\frac{dr^i(u)}{du} \right)^3 \right\} F^i \\ & + (3 + \nu) \rho \omega^2 t^i(u) (r^i(u))^3 \left(\frac{dr^i(u)}{du} \right)^3 = 0 \end{aligned} \quad (6.24)$$

The Eq. (6.24) is the second order differential equation with the independent variable 'u' and dependent variable 'F'. In case of flywheel design, two boundary conditions as radial stresses equal to zero at the inner radius and outer radius are usually known. Then, the Equation (6.24) becomes a two-point boundary value problem with a second order differential equation. If the equation is represented for the whole range, the number of numerical methods are available to solve the equation such as the finite difference method and the shooting method (Ahsan and Farrukh, 2013; Kharab and Guenther, 2012). Furthermore, a MATLAB inbuilt function as ode45 is directly used to solve this equation. But, Equation (6.24) is piecewise equation which different for each segment or sub-range due to the variation of 'u'. Then the equation is difficult to solve by the above methods, and fewer researchers have been reported to solve for this

type of equations. Therefore, a flowchart is proposed for solving the piecewise equation with two-point boundary value condition to find out the tangential and radial stresses at each point of each segment along the radial direction, is shown in Fig.6.3. The detailed procedure for solving is the piece wise equation with two-point boundary value is described as:

Step 1: set the n, k, S, tolerance $\epsilon = 10^{-3}$ and two boundary conditions as radial stresses are zero at the inner and outer radius. From Eq. (6.19)

$$F^1|_{u=0} = t^1(u)r^1(u)\sigma_r^1 = 0; \quad r^1(u)|_{u=0} = R_1 \quad \sigma_r^1 = 0$$

$$F^S|_{u=S} = t^S(u)r^S(u)\sigma_r^S = 0; \quad r^S(u)|_{u=S} = R_2 \quad \sigma_r^S = 0$$

Step 2: guess the initial slop as $\frac{dF^1}{du}|_{u=0} = s_1$

Step 3: divide each segment into the number of points by taking step size as h. Solve the differential Eq. (6.24) for each segment for slope s_1 using Runge Kutta method by applying B-Spline properties as

$$F_1^i|_{u=i} = F_1^{i+1}|_{u=i}$$

$$\frac{dF^i}{du}|_{u=i} = \frac{dF^{i+1}}{du}|_{u=i}; \text{ for } i = 1 \text{ to } S$$

Step 4: if $F^S|_{u=S} \leq F_1^S|_{u=S}$, set slope as $\frac{dF^1}{du}|_{u=0} = s_2 = \frac{s_1 + \frac{dF^S}{du}|_{u=S}}{2}$

Otherwise $\frac{dF^1}{du}|_{u=0} = s_2 = \frac{s_1 - \frac{dF^S}{du}|_{u=S}}{2}$

Step 5: Solve the differential Eq. (6.24) for each segment for slope s_2 using Runge-Kutta method by applying B-Spline properties as:

$$F_2^i|_{u=i} = F_2^{i+1}|_{u=i}$$

$$\frac{dF^i}{du}|_{u=i} = \frac{dF^{i+1}}{du}|_{u=i}; \text{ for } i = 1 \text{ to } S$$

Step 6: update the stress function and slope

$$s_3 = s_1 + (s_2 - s_1) \times \frac{F^S - F_1^S}{F_2^S - F_1^S}|_{u=m}$$

$$s_1 = s_2; \quad s_2 = s_3; \quad F_1^S = F_2^S|_{u=S}$$

Step 7: If $abs(F^S - F_2^S)|_{u=S} \leq \epsilon$

Radial and tangential stresses are calculated at each point of each segment using Eqs. (6.20) and (6.23) from known values of stress function $F_2^i|_{u=i}$ and slope $\frac{dF^i}{du}|_{u=i}$

Otherwise, go to step 5

Once radial and tangential stresses are known at each point of each segment, the Von Mises stresses at each point are calculated using the application of distortion energy theory as:

$$\sigma_t = (\sigma_r^2 + \sigma_\theta^2 - \sigma_r \sigma_\theta)^{1/2} \quad (6.25)$$

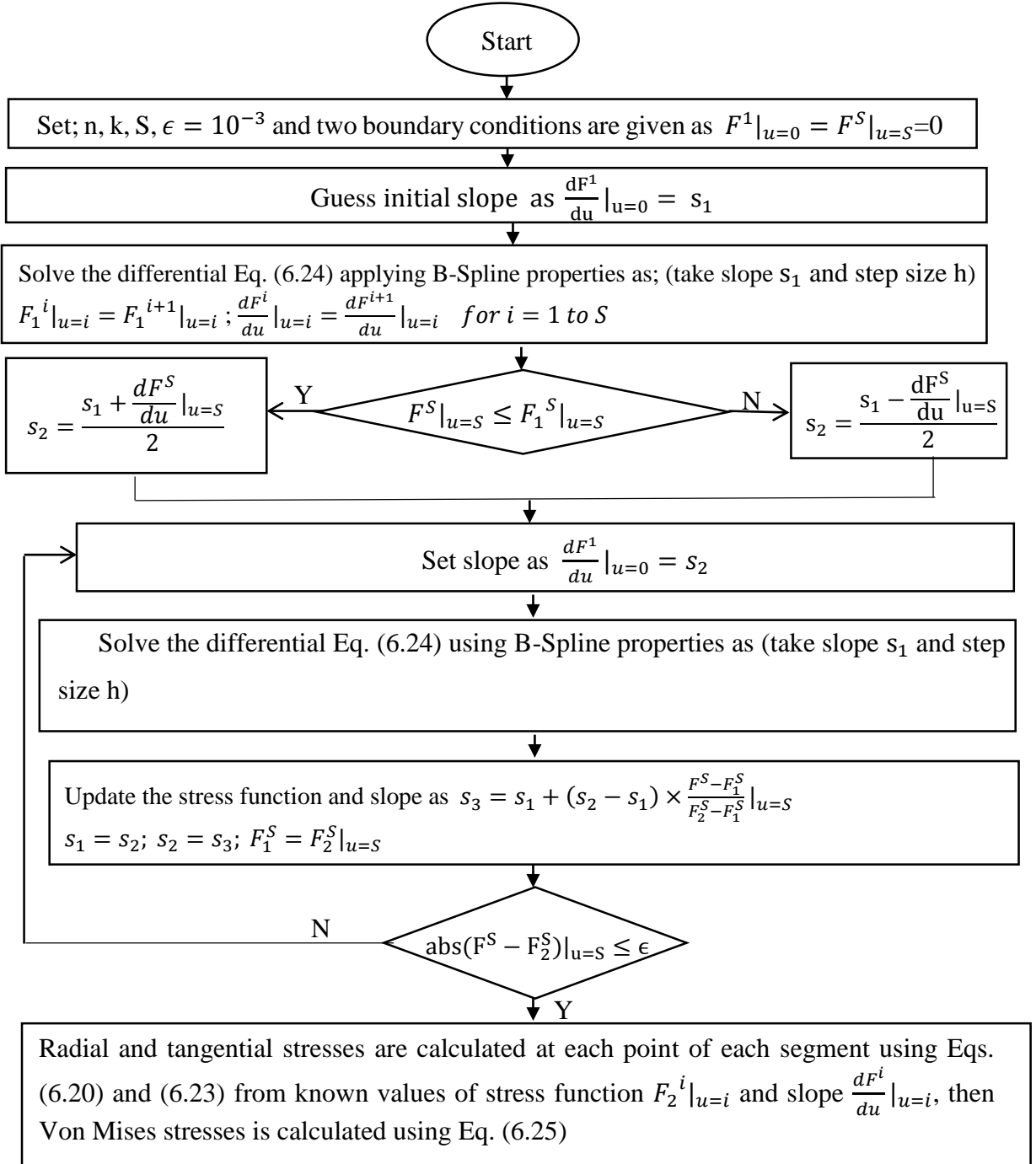


Fig. 6.3. A flow chart for solving the piece wise equation with two-point boundary value condition

6.2. Formulation of the optimization problem

A nonlinear constrained shape optimization problem for optimal distribution of thickness along the radial direction is formulated using the cubic B-spline. The various design parameters of the flywheel and global coordinate system, OXY are described in Fig.6.1. Radial points (r_1, r_2, \dots, r_n) and thickness points (t_1, t_2, \dots, t_n) are defined in the X and Y directions, respectively.

6.2.1. Design variables

The radial points of the flywheel between R_i to R_o are equally spaced. Thus, the radial points are fixed and given by Equation as $r_i = r_1 + \frac{(r_2-r_1)(i-1)}{n-1}$ where $i = 1, \dots, n$. Furthermore, thickness points (Y coordinates of the control points) are considered as the design variables and are defined in vector form as:

$$\mathbf{x} = [t_1 \ t_2 \ t_3 \ \dots \ \dots \ \dots \ t_n]^T \quad (6.26)$$

Where n represents the control points or design variables spaced between R_1 to R_2 .

6.2.2. Objective function and constraints

In general, the flywheel should store as much kinetic energy as possible for its better performance. Although the kinetic energy stored in the flywheel can be improved by the shape of the flywheel. Therefore, the kinetic energy is considered as objective function and the mass of the flywheel and the maximum value of Von Mises stresses are taken as the design constraints in this shape optimization problem.

Thus, the kinetic energy of the flywheel is determined by using Eq. (6.16) and design constraints as mass and the maximum value of Von Mises stress are determined using Eqs. (6.15) and (6.25), respectively. However, a mass constraint is defined as the mass of the flywheel should be less than the maximum mass (M_{max}) of the flywheel

$$g_1(\mathbf{x}) = M \leq M_{max} \quad (6.27)$$

While stress constraint is defined as the maximum value of Von Mises stress at all points of each segment in the radial direction should be less than the allowable stress (σ_a).

$$g_2(\mathbf{x}) = \max(\sigma_t) \leq \sigma_a \quad (6.28)$$

Finally, the optimization problem is posed as

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) = -E_K \\ &\text{Subjected to} \end{aligned} \quad (6.29)$$

$$\begin{aligned}
g_1(\mathbf{x}) &= M - M_{\max} \leq 0 \\
g_2(\mathbf{x}) &= \max(\sigma_t) - \sigma_a \leq 0 \\
LB_i &\leq x_i \leq UB_i \quad i = 1, \dots, \dots, n
\end{aligned} \tag{6.30}$$

The negative sign in the expression for the objective function implies that the function E_k is being maximized. LB_i and UB_i are the lower and upper bounds on the i^{th} design variable, and n represents the number of design variables. To obtain an optimum solution, the constrained problem, as defined in Eq. (6.29), is converted into an unconstrained problem by using a penalty function (Singh et al., 2017). A significant penalty value is added to the objective function for each constraint violation. As a result, the objective function proceeds toward an infeasible solution. Hence, the global optimum solution is obtained by satisfying all the constraints using a suitable optimization algorithm. The original constrained optimization problem is then stated as an unconstrained optimization problem in which the first and second term describes the objective function and the penalty function, respectively. Finally, the shape optimization problem of the flywheel is formulated as:

$$\text{Minimize } f(\mathbf{x}) = -E_K + \sum_{B=1}^2 C_B (C)^B \tag{6.31}$$

$$LB_i \leq x_i \leq UB_i \quad i = 1, \dots, \dots, n \tag{6.32}$$

Where C is the penalty value of the order of 10^6 which assign to objective function if the constraints are not satisfied and C_B is defined as (Singh et al., 2017)

$$C_B = \begin{cases} 1 & \text{if } g_B(\mathbf{x}) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

6.3. Optimization algorithm

After formulating the optimization problem, it can be solved either by classical methods or nature-inspired optimization algorithms. The traditional methods involve the computational complexity in calculating the gradient of the objective function correspond to the design variables. Moreover, these methods give only a local optimum solution (Arora, 2004).

Nature-inspired optimization algorithms have less computational complexity compared to classical methods. There are some nature-inspired optimization algorithms, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and

Ant Colony Optimization (ACO). Moreover, these algorithms converge to a global minimum, but there is no guarantee of an optimal global solution.

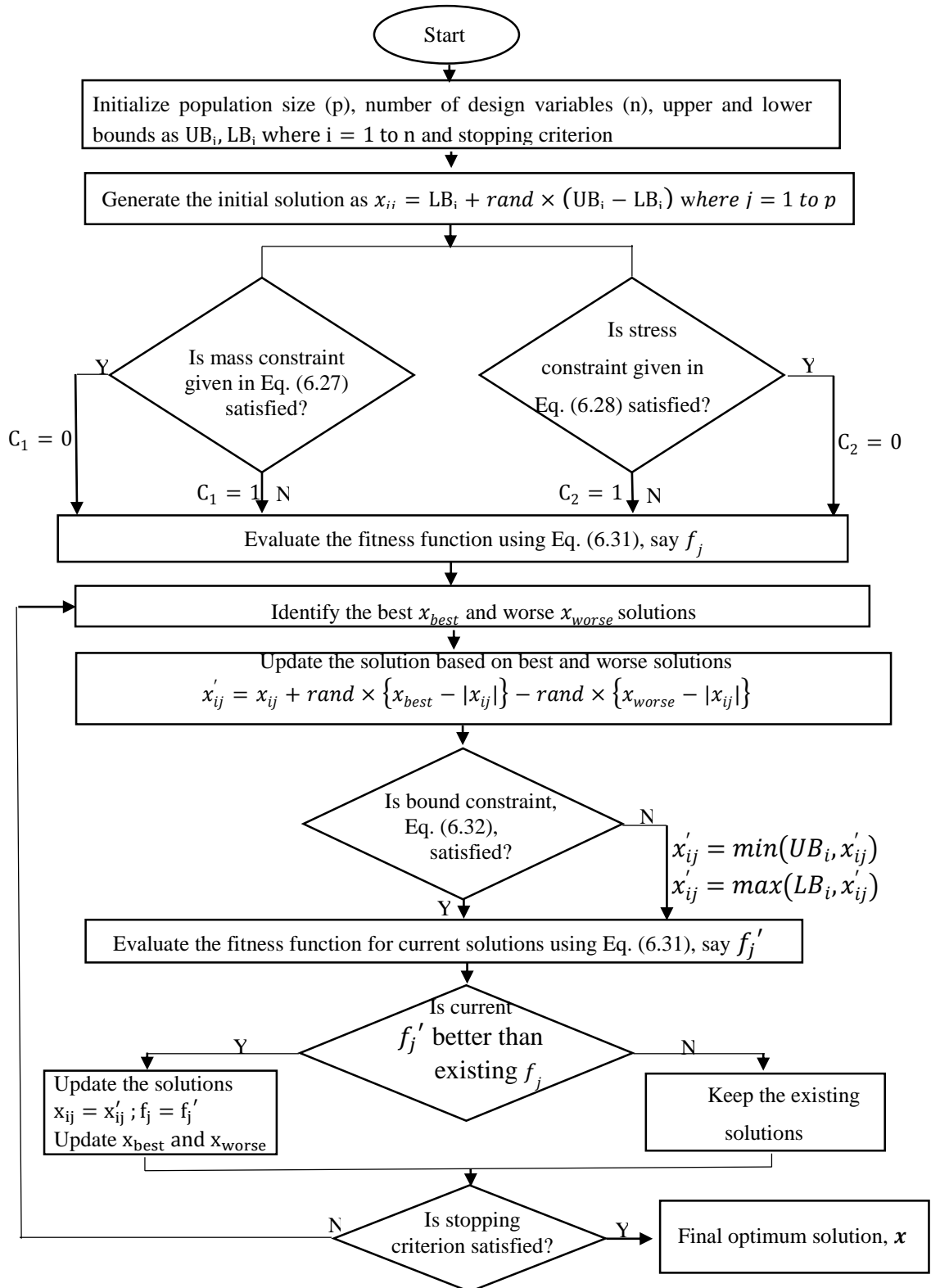


Fig. 6.4. A flow chart of Jaya algorithm for the shape of the flywheel

Some of the algorithms such as GA, PSO, and ACO have algorithmic control parameters, which affect the performance of algorithms (Rao and Savsani, 2012). Furthermore, the Jaya algorithm is a parameter-less algorithm that has only one phase, unlike the two phases of the TLBO algorithm (Chaudhary and Chaudhary, 2015b; Rao, 2015). It gives the optimal solution rapidly and updates the worst solution (Rao, 2016; Venkata Rao and Waghmare, 2016). A flow chart of Jaya algorithm for the shape of the flywheel is shown in Fig. 6.4. Here, the stopping criterion for Jaya algorithm is considered as the number of iterations and number of function evaluations.

The number of function evaluations depends on population size and the number of iterations. The number of function evaluations is equal to the product of the iteration numbers and population size. Therefore, design variables do not affect the function evaluations but may affect the computational time of the algorithm. The detailed procedure of GA and PSO applied in this study is explained by (Chaudhary and Chaudhary, 2014) and (Pathak et al., 2017), respectively. However, the first time it is implemented for the optimal design of a flywheel in this study.

Notations used in the Jaya algorithm are defined as follows:

p = population size

LB_i, UB_i = Lower and upper bounds for the i th design variables

x_i = i th design variable

x_{ij} = i th design variable for j th population

f_j = An objective function value of the j th population Eq. (6.31)

rand = Any random number in the range of 0 and 1

6.4. Flywheel design problems

The proposed optimization procedure for the shape of the flywheel is applied for a wide variety of problem formulations. The effectiveness of this approach is illustrated by the numerical example of the flywheel design taken from the literature (Jiang et al., 2016) and the flywheel design of the thresher machine. Design variables are equal to the control points that are divided between the inner and outer radius. The eight control points are taken for the optimal shape of the flywheel. Von Mises stresses at each point, as expressed in Eq. (6.25), are solved by taking the number of segments ($S=5$) and step size (h) = 0.1. A total of 51 points are taken between the inner radius and outer radius for this study. Moreover, the homogenous material is considered for the design of flywheel.

6.4.1. Numerical example of the flywheel design

A numerical problem of flywheel design (Jiang et al., 2016) is solved using the proposed method. Material properties and design parameters of the flywheel are given in Tables 6.1 and 6.2, respectively.

The upper and lower limits of the design variables are taken as $0.050 \leq x_i \leq 0.200$ where $i = 1, \dots, 8$.

Table 6.1. Material properties of the flywheel

Material	Density (kg/m^3)	Elastic modulus (Gpa)	Poisson's ratio
Low alloy carbon steel	7850	210	0.3

Table 6.2. Design parameters of the flywheel

Control points n	Inner radius R_1 (m)	outer radius R_2 (m)	Angular velocity ω (rad/sec)	M_{max} (kg)	σ_a (N/mm^2)
8	0.150	0.800	188.50	1849	120

The nonlinear shape optimization problem as explained in Eq. (6.31) is solved using GA, PSO, and Jaya algorithm. These algorithms are coded in MAT LAB. In order to find the best objective values, population size and the number of iterations in these algorithms are chosen as 10 and 100, respectively. However, the controlling parameters are required for their convergence in case of PSO and GA while Jaya algorithm is not required such controlling parameters. These algorithms are run for 1000 function evaluations to obtain the best objective function value and corresponding design variables. GA, PSO, and Jaya reach to the optimum value of the objective function at $-1.246E+07$, $-1.308E+07$ and $-1.373E+07$, respectively, as shown in Fig. 6.5. These optimum results obtained from these algorithms are compared with those of the conventional method (Jiang et al., 2016) as given in Table 6.3, and the best value of the objective function is shown in boldface. Table 6.3 shows that the Jaya algorithm gives better results compared with those of the conventional method, GA, and PSO.

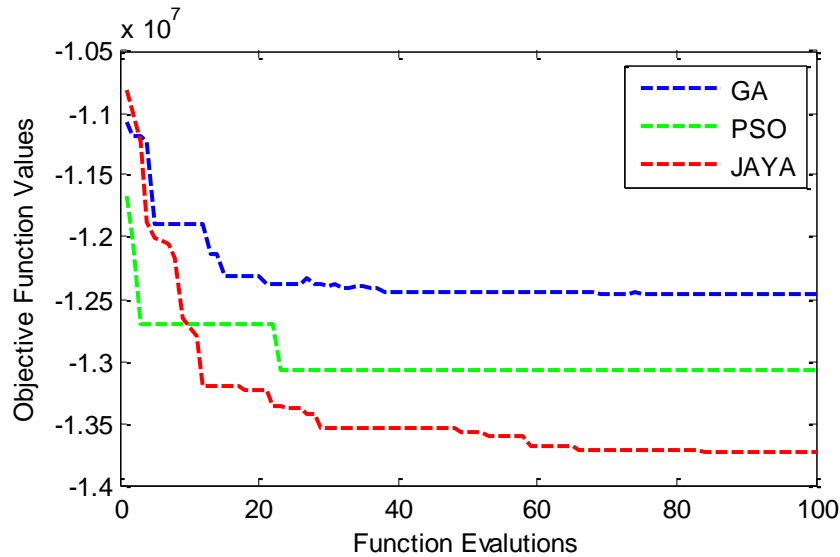


Fig. 6.5. Convergence of the best objective function value in PSO, GA, and Jaya.

The optimized shape of the flywheel using the optimum design parameters obtained from the Jaya algorithm stores the maximum energy compared to that of the conventional method. The optimal thickness distribution of the flywheel with respect to the Jaya algorithm and the conventional method is presented in Fig. 6.6 and the comparison of the optimized shape with that of the conventional method is shown in Fig. 6.7. Figure 6.7 shows that the profile of the flywheel is thin in the middle section and thicker at the inner and outer sections.

Table 6.3. The optimum value for the shape of the flywheel

Optimum Values	Conventional Method (Jiang et al., 2016)	GA	PSO	Jaya
$t_1(m)$	0.122	0.0585	0.200	0.083
$t_2(m)$	0.065	0.1413	0.050	0.050
$t_3(m)$	0.056	0.0508	0.050	0.050
$t_4(m)$	0.052	0.0653	0.050	0.050
$t_5(m)$	0.064	0.1029	0.050	0.050
$t_6(m)$	0.133	0.1751	0.200	0.174
$t_7(m)$	0.195	0.1186	0.152	0.200
$t_8(m)$	0.197	0.1922	0.200	0.200
M (kg)	1849	1848.51	1848.81	1848.30
$E_K(J)$	1.35E+07	1.246E+07	1.308E+07	1.373E+07
$f(\mathbf{x})$	-1.35E+07	-1.246E+07	-1.308E+07	-1.373E+07

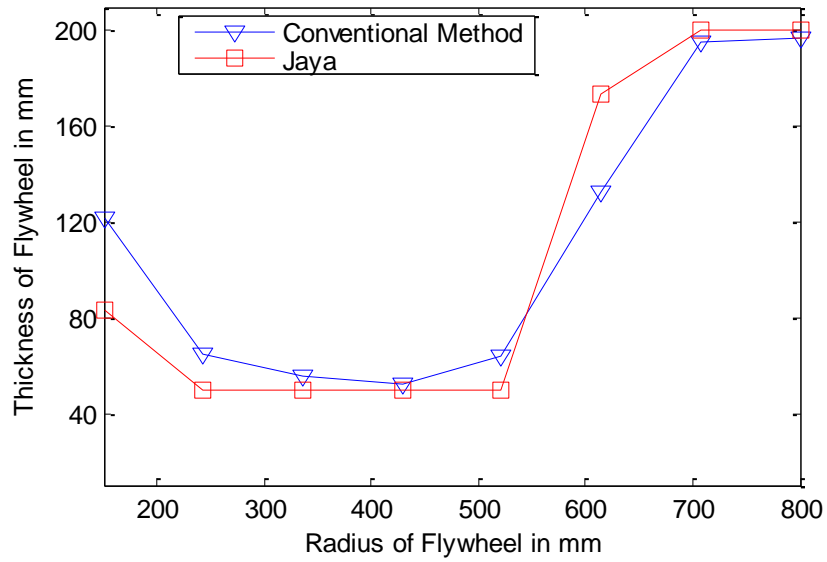


Fig. 6.6. Optimal thickness distribution along the radial direction

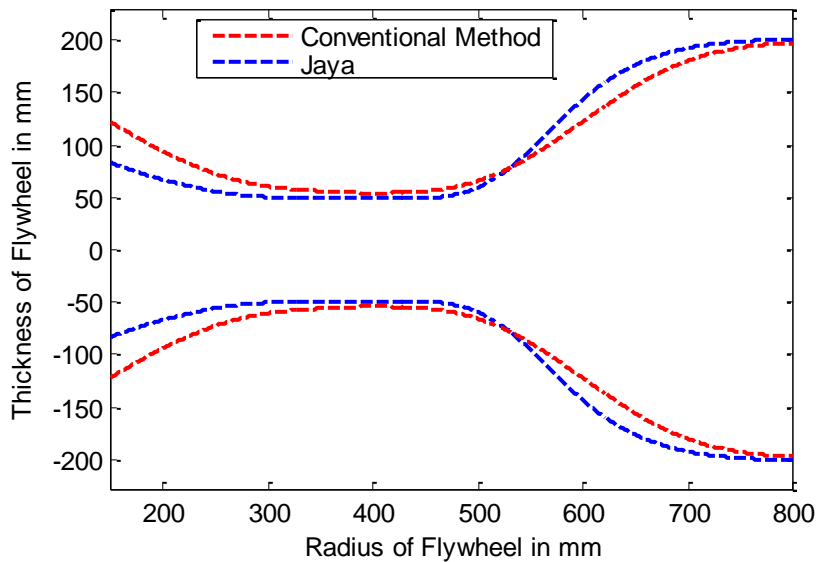


Fig. 6.7. Optimized shape correspond to Jaya and conventional method

Stress distribution along the radial direction in all sections of the optimized shape of the flywheel is shown in Fig 6.8. Generally, the maximum value of Von Mises stresses occurs near the inner section of flywheel due to the central hole. But, it is reduced by increasing the thickness in the inner section as shown in Fig.6.8. Moreover, the middle section is more stressed compared to other section of the flywheel. The stresses decrease at the outer section due to increase in thickness.

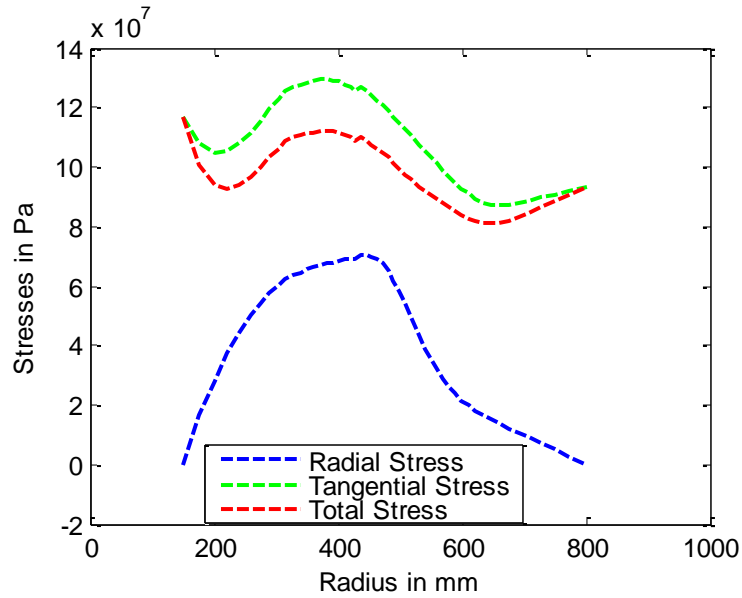


Fig. 6.8. Stress distribution along the radial direction for the optimized shape of the flywheel

6.4.2. The flywheel design of the thresher machine

The proposed approach is applied to the flywheel of the thresher machine. The function of the flywheel in a thresher machine is to minimize the variations in the speed of the PTO shaft due to torque fluctuations of the threshing drum by storing or releasing kinetic energy. The material properties and design parameters of the flywheel are given in Tables 6.4 and 6.5, respectively (Varshney, 2004). The upper and lower limits of the design variables are taken as

$$0.010 \leq x_i \leq 0.060 \quad \text{where } i = 1 \text{ to } 8$$

Table 6.4. Material properties of flywheel

Material	Density (kg/m^3)	Elastic modulus (Gpa)	Poisson's ratio
grey cast iron	7250	210	0.3

Table 6.5. Design parameters of the flywheel

Control points	Inner radius	outer radius	Angular velocity	M_{max}	$\sigma_a(N/mm^2)$
n	R_1 (m)	R_2 (m)	ω (rad/sec)	(kg)	
8	0.060	0.500	65.45	115	6.4

The Jaya algorithm was coded in MATLAB. In order to compare the efficiency of the algorithm, GA, and PSO are also applied to the same flywheel design problem. All three algorithms were run for 100 iterations to find the best objective function value and corresponding design variables. GA, PSO, and Jaya give the best of the objective function as -28950, -32896.25, and -41624.8, respectively. The convergence plots of the best objective function values are shown in Fig.6.9. The optimum results, obtained from these algorithms, are compared with the original design (Varshney, 2004) in Table 6.6, and the best value of the objective function is shown in boldface. It was found that Jaya gives better results compared to the results obtained from GA and PSO.

The optimized shape of the flywheel stores 36.55% more energy compared to the existing flywheel shape (Ghaly, 1985; Madan Lal, 2012). As a result of this, less torque and fuel consumption will be required to run the thresher machine. Comparison of the optimized shape of the flywheel correspond to the Jaya algorithm with the original shape of thresher machine flywheel is shown in Fig.6.10. The profile of the optimized flywheel is divided into three sections like thin in the middle section and thicker at inner and outer section, while the original flywheel has a constant thickness profile over the entire section. Further, the optimal values of the thicknesses are also effectively converted into physically possible shapes of the flywheel using PTC Creo 3.0 as shown in Fig.6.11. Hence, the optimized shape of flywheel with respect to Jaya algorithm stores the maximum energy compared with the existing flywheel of thresher machine.

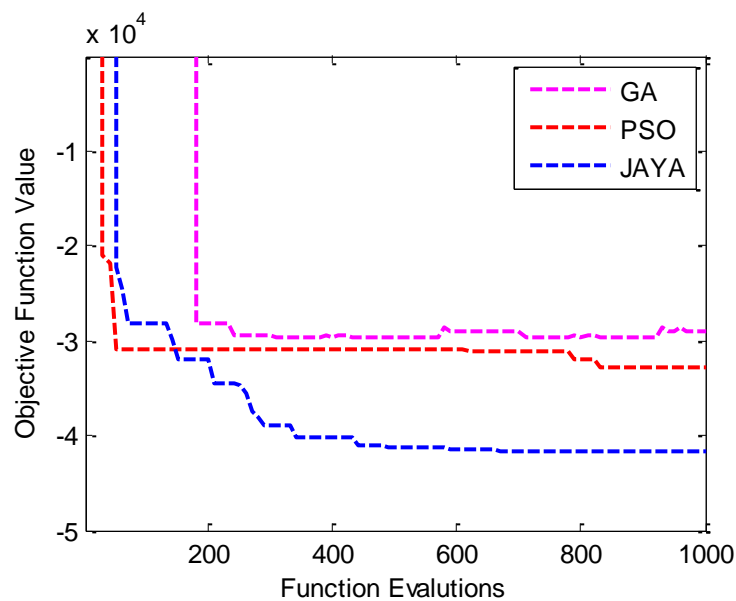


Fig. 6.9. Convergence of the best objective function value in PSO, GA, and Jaya.

Table 6.6. The optimum value for the shape of the flywheel

Optimum Values	Original (Varshney, 2004)	GA	PSO	Jaya
$t_1(m)$	0.020	0.0267	0.01	0.0189
$t_2(m)$	0.020	0.0122	0.01	0.01
$t_3(m)$	0.020	0.0228	0.06	0.01
$t_4(m)$	0.020	0.0292	0.01	0.01
$t_5(m)$	0.020	0.0248	0.01	0.01
$t_6(m)$	0.020	0.0182	0.01	0.01
$t_7(m)$	0.020	0.013	0.01	0.0246
$t_8(m)$	0.020	0.0211	0.0489	0.06
M (kg)	115	115	115	115
$E_K(J)$	30483.66	28950	32896.25	41624.88 (36.55%)
$f(\mathbf{x})$	-	-28950	-32896.25	-41624.88

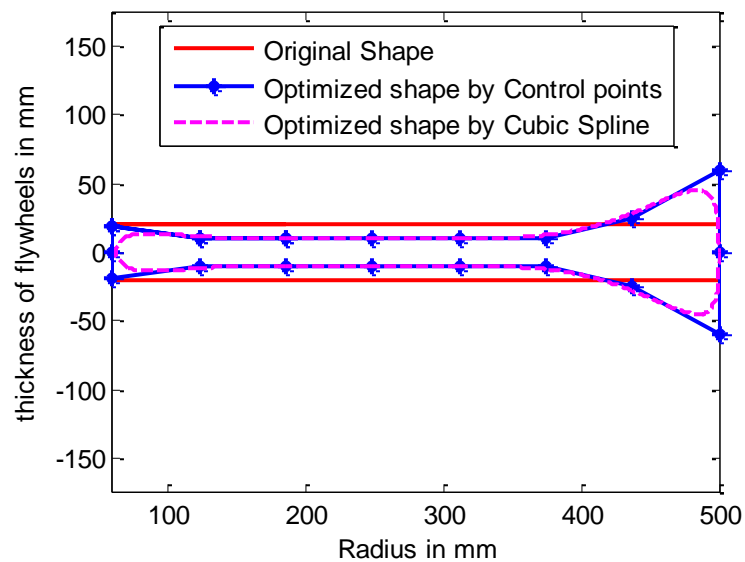


Fig. 6.10. Original shape and optimized shape of the flywheel

The stress distribution of radial, tangential, and Von Mises stresses along the radial direction in the original and optimized flywheel in all sections is shown in Fig.6.12. Figure 6.12 shows that the optimized shape of the flywheel develops less stress compared to those of the original shape of the flywheel. Thus, the material of the optimized flywheel will not fail during its operation.

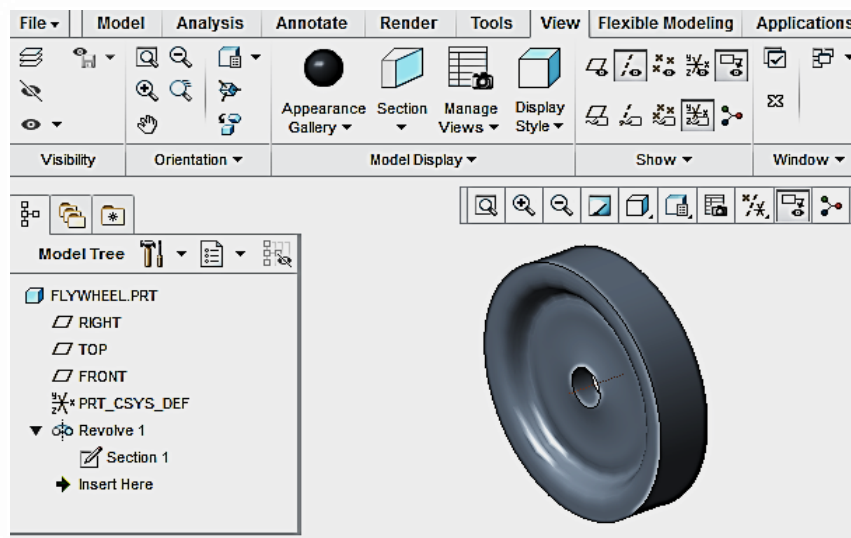


Fig. 6.11. A Cad model of the Optimized flywheel

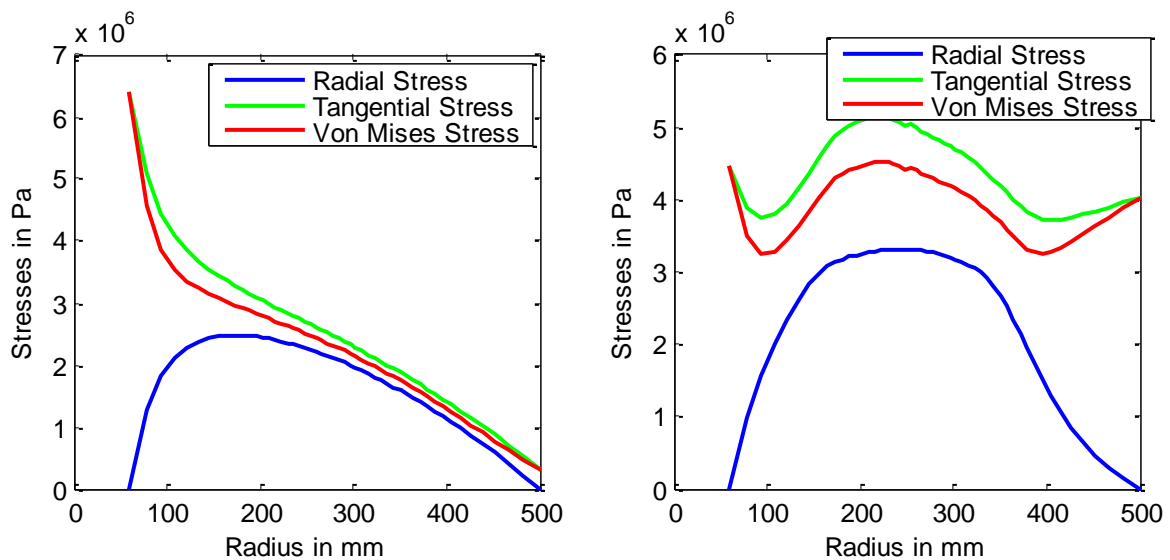


Fig. 6.12. Stress distribution along the radial direction in (a) original flywheel (b) optimized flywheel

The original and optimized configurations of the flywheel of the thresher machine was simulated using MSC ADAMS software (ADAMS, 2014). The Input torque variations for the original and optimized flywheel are shown in Fig.6.13. The input torque is generated by the PTO of a tractor to run the thresher machine. So, the input torque must be minimum for better performance of the thresher machine. It was seen

that the optimized shape of the flywheel reduced the maximum and minimum values of input torque up to 15.65%, and 16.58% compared to the original values.

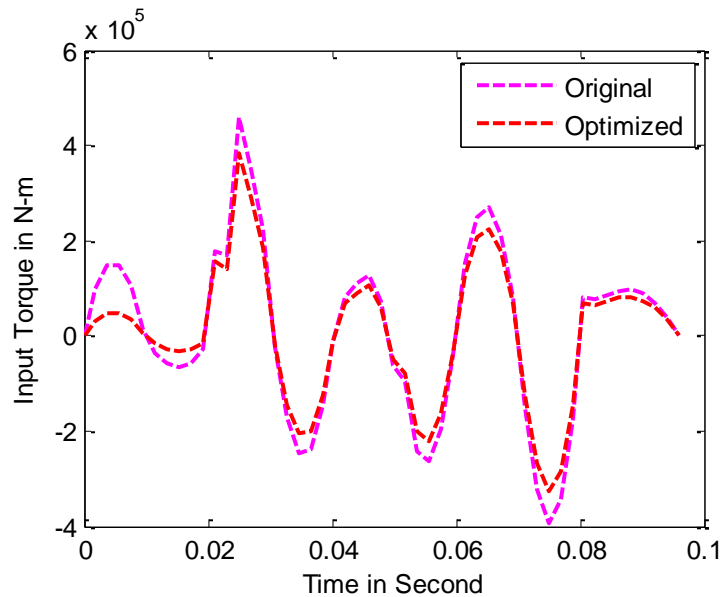


Fig. 6.13. Torque variation in the original and optimized flywheel

6.5. Summary

The shape optimization procedure of the flywheel using a cubic B-spline is described in this chapter. The shape optimization problem with maximization of the kinetic energy is formulated by considering the design constraints of the mass of the flywheel, and the maximum value of Von Mises stresses. To the analysis of stresses, a flowchart is proposed to solve the two-point boundary value differential equation for stress distribution at each radial location. The proposed approach is tested by the flywheel design taken from literature and the flywheel design of the thresher machine. The proposed method also demonstrates Jaya, PSO, and GA as a solver for the shape optimization of the flywheel. It is found that Jaya gives better results compared to results obtained using GA and PSO. The optimized shape of flywheel obtained by Jaya algorithm stores the maximum kinetic energy compared to that of the conventional method. The stress distribution in the middle section of the optimized flywheel shape is more compared to the near the shaft and outer sections. While the optimized shape of the flywheel for the thresher machine stores 36.55% more energy compared with that of the original flywheel, without potential failure of the material during its operation. The optimal value of thicknesses is also effectively converted into physically possible shapes of the flywheel using any CAD software. The MSC ADAMS software was also

used for simulation of the optimized shape of the flywheel. It was found that the maximum and minimum values of torque are reduced up to 15.65%, and 16.58% compared with the original values. Thus, farmers can use the optimized shape of the flywheel in place of the existing flywheel. As a result of this, dynamic performance (like vibrations, torque fluctuation) of the existing thresher machine will improve, and also accidents involving humans will be reduced.

Chapter 7

Conclusions

This thesis presents the dynamic performance improvement of the thresher machine to minimize fuel consumption and human accidents. The moving parts or mechanisms namely, cleaning mechanism, threshing drum, and the flywheel are considered for the improvement in the dynamic performance of the thresher machine. The multi-objective optimization problem is formulated to minimize the shaking force and shaking moment developed in the cleaning mechanism as described in Chapter 3. The cleaning mechanisms is balanced using the optimal inertial properties of each moving link. These inertial properties are represented by the dynamically equivalent point-mass system. The parameters of this system are considered as the design variables in the formulated optimization problem. The priori approach based algorithms like GA and Jaya, and the posteriori approach based algorithm as non-dominated sorting Jaya algorithm (NSJAYA) are applied to solve the formulated problem. It is established that NSJAYA is computationally more efficient than the GA and Jaya algorithm. In this study, NSJAYA is used the first time for the balancing of the planar mechanism. The RMS values of the shaking force and shaking moment are reduced using the optimum design parameters of solution 1 up to 87.80% and 83.05%, respectively correspond to those of the original mechanism. The designer can choose more alternatives based on the importance of objectives.

In chapter 4, a modified Jaya algorithm is proposed for balancing of threshing drum using mixed variable optimization problems. Original Jaya algorithm has been developed for continuous optimization problems. Thus, Jaya algorithm is further extended for solving the mixed variable optimization problems. The efficiency of the proposed algorithm is demonstrated by five design problem taken from literature. It is found that the proposed algorithm are compared with those of well-known optimization algorithms. The results show that it gives the better and nearly close results compared to other optimization algorithms with fewer function evaluations. In chapter 5, optimum two-plane discrete balancing procedure is developed for the rigid rotor to improve the dynamic performance of the thresher machine. The multi-objective problem is formulated to minimize the reaction forces at supports, and discrete parameters such as masses and corresponding angular positions on each balancing plane are considered as

design variables. The effectiveness of the proposed approach is demonstrated by a numerical problem taken from literature, and it is also applied to the threshing drum model of the thresher machine. The proposed modified Jaya algorithm and GA are used as solvers for these balancing problems. It is observed that a modified Jaya algorithm takes fewer function evaluations and gives the better and nearly close results than those of GA for all cases. The reaction forces at supports P and Q are reduced up to 70.85% and 65.70 % for case I, 92.13% and 93.62 % for case II, 98.76% and 99.69% for case III, and 99.85% and 99.56% for case IV corresponding to the static unbalanced rotor, respectively. Similarly, 72% and 80 % for case I, 92.7% and 93.6 for case II, 99.3% and 98.7% for case III, and 99.4% and 99% for case IV correspond to the dynamic unbalanced rotor, respectively. Hence the reaction forces at supports decrease, as the number of balance mass per plane increases.

In case of the threshing drum balancing problem, reaction forces at supports P and Q are reduced by 65% and 87.32 % for case 1, 98.28% and 98.15 % for case 2, 99.78% and 98.74% for case 3, respectively, correspond to those of the unbalanced threshing drum.

The unbalanced and balanced rotors are simulated using MSC ADAMS software. It is found that the simulation and theoretical results of the component forces are in good agreement without variation. Besides the analytical study of the threshing drum, the experimental study is also investigated. The experimental study shows that the number of discrete masses per plane decreases the vibration amplitudes of the unbalanced threshing drum. Thus the dynamic performance of the thresher machine can be improved. The proposed approach is quite general and equally applicable to the rigid and flexible rotors.

The optimal shape synthesis procedure of the flywheel using a cubic B-spline is other approach to improve the dynamic performance of the thresher machine as described in chapter 6. The shape optimization problem with maximization of the kinetic energy is formulated by considering the design constraints of the mass of flywheel, and the maximum value of Von Mises stresses. To analysis of stresses, a flowchart is proposed to solve the two-point boundary value differential equation for stress distribution at each radial location. The proposed approach is tested by the flywheel design taken from literature and the flywheel design of the thresher machine. The proposed method also demonstrates Jaya, PSO, and GA as a solver for the shape optimization of the flywheel. It is found that Jaya gives better results compared to

results obtained using GA and PSO. The optimized shape of flywheel obtained by Jaya algorithm stores the maximum kinetic energy compared to that of the conventional method. The stress distribution in the middle section of the optimized flywheel is more compared to the near the shaft and outer sections. While the optimized shape of the flywheel for the thresher machine stores 36.55% more energy compared with that of the original flywheel, without potential failure of the material during its operation. The optimal value of thicknesses is also effectively converted into physically possible shapes of the flywheel using CAD software. The MSC ADAMS software is also used for simulation of the optimized shape of the flywheel. It was found that the maximum and minimum values of torque are reduced up to 15.65%, and 16.58% compared with the original values. Thus, farmers can use the optimized shape of the flywheel in place of the existing flywheel. As a result of that, dynamic performance (like vibrations, torque fluctuation) of the existing thresher machine will improve, and also accidents involving humans will be reduced.

The contributions of this research work are summarized as follows:

1. A multi-objective optimization problem for the balancing of the cleaning mechanism is proposed.
2. A posteriori approach based algorithm as a non-dominated sorting Jaya algorithm (NSJAYA) is applied to find the optimal mass distribution of the links for cleaning mechanism.
3. A discrete optimization problem for two-plane balancing of the rigid rotor is proposed.
4. A modified Jaya algorithm is proposed to find the optimal discrete solutions. The results are validated by the experiments and commercial software.
5. A shape optimization problem for the flywheel using the cubic B-spline curve is proposed.
6. Particle swarm algorithm (PSO), genetic algorithm (GA), and Jaya algorithm are applied for the shape optimization problem. It is established that Jaya algorithm is computationally more efficient than GA and PSO.

Future Scope of the Work

1. The elastic deformation and clearance in joints may be considered for the links of the cleaning mechanism

2. The experimental study may be used for the simulation of the cleaning mechanism.
3. A multi-objective discrete optimization algorithm can be applied to balance the rigid rotor
4. In this research work, single objective functions are considered for the shape of the flywheel. The application of multi-objective functions may render useful results.

References

- ADAMS, M. (2014), "Automatic dynamic analysis of mechanical systems", MSC Software Corporation.
- Ahmad, S.A., Iqbal, M., Ahmad, M., Tanveer, A. and Sial, J.K. (2013), "DESIGN IMPROVEMENT OF INDIGENOUS BEATER WHEAT THRESHER IN PAKISTAN", Vol. 50 No. 4, pp. 711–721.
- Ahsan, M. and Farrukh, S. (2013), "A new type of shooting method for nonlinear boundary value problems", *Alexandria Engineering Journal*, Vol. 52 No. 4, pp. 801–805.
- Ani, O.A., Xu, H., Shen, Y., Liu, S. and Xue, K. (2013), "Modeling and multiobjective optimization of traction performance for autonomous wheeled mobile robot in rough terrain", *Journal of Zhejiang University SCIENCE C*, Vol. 14 No. 1, pp. 11–29.
- Arakelian, V. and Dahan, M. (2001), "Partial shaking moment balancing of fully force balanced linkages", *Mechanism and Machine Theory*, Vol. 36 No. 11-12, pp. 1241–1252.
- Arakelian, V.H. and Smith, M.R. (1999), "Complete shaking force and shaking moment balancing of linkages", *Mechanism and Machine Theory*, Vol. 34 No. 8, pp. 1141–1153.
- Arakelian, V.H. and Smith, M.R. (2005), "Shaking Force and Shaking Moment Balancing of Mechanisms: A Historical Review With New Examples", *Journal of Mechanical Design*, Vol. 127 No. 2, pp. 334–339.
- Arora, J. (2004), *Introduction to Optimum Design, Third Edition*, Elsevier.
- Arora, J.S. (2000), "Methods for Discrete Variable Structural Optimization", *Advanced Technology in Structural Engineering*, pp. 1–8.
- Arora, J.S., Huang, M.W. and Hsieh, C.C. (1994), "Methods for optimization of nonlinear problems with discrete variables: A review", *Structural Optimization*, Vol. 8 No. 2-3, pp. 69–85.
- Arslan, M.A. (2008), "Materials & Design Flywheel geometry design for improved energy storage using finite element analysis", Vol. 29, pp. 514–518.
- Bagci, C. (1979), "Shaking force balancing of planar linkages with force transmission irregularities using balancing idler loops", *Mechanism and Machine Theory*, Vol. 14 No. 4, pp. 267–284.

- Bagci, C. (1982), “Complete Shaking Force and Shaking Moment Balancing of Link Mechanisms Using Balancing Idler Loops”, *Journal of Mechanical Design*, Vol. 104 No. 2, p. 482.
- Beck, R., Katz, J.N., Martin, A.D. and Quinn, K.M. (1995), “A Review of Discrete Optimization Algorithms”, pp. 6–10.
- Berger, M. and Porat, I. (2017), “Optimal Design of a Rotating Disk for Kinetic Energy Storage & H”, Vol. 55 No. March 1988.
- Berkof, R.S. (1973), “Complete force and moment balancing of inline four-bar linkages”, *Mechanism and Machine Theory*, Vol. 8 No. 3, pp. 397–410.
- Berkof, R.S. and Lowen, G.G. (1969), “A New Method for Completely Force Balancing Simple Linkages”, *Journal of Engineering for Industry*, Vol. 91 No. 1, p. 21.
- Borchers, B. and Mitchell, J.E. (1994), “An improved branch and bound algorithm for mixed integer nonlinear programs”, *Computers & Operations Research*, Vol. 21 No. 4, pp. 359–367.
- Camp, C. V. and Bichon, B.J. (2004), “Design of Space Trusses Using Ant Colony Optimization”, *Journal of Structural Engineering*, Vol. 130 No. 5, pp. 741–751.
- Cao, Y.J. (2000), “An evolutionary programming approach to mixed-variable optimization problems”, Vol. 24, pp. 931–942.
- Carson, W.L. and Stephens, J.M. (1978), “Feasible parameter design spaces for force and root-mean-square moment balancing an in-line 4R 4-bar synthesized for kinematic criteria”, *Mechanism and Machine Theory*, Vol. 13 No. 6, pp. 649–658.
- Chaudhary H, Saha, S.K. (2009), *Dynamics and Balancing of Multibody Systems*, Springer Verlag, Germany.
- Chaudhary, H. and Saha, S.K. (2007a), “Dynamic performance improvement of a carpet scrapping machine”, *Journal of Scientific and Industrial Research*, Vol. 66 No. 12, pp. 1002–1010.
- Chaudhary, H. and Saha, S.K. (2007b), “Balancing of four-bar linkages using maximum recursive dynamic algorithm”, *Mechanism and Machine Theory*, Vol. 42 No. 2, pp. 216–232.
- Chaudhary, K. and Chaudhary, H. (2014), “Dynamic balancing of planar mechanisms using genetic algorithm”, *Journal of Mechanical Science and Technology*, Vol. 28 No. 10, pp. 4213–4220.

- Chaudhary, K. and Chaudhary, H. (2015a), “Optimal dynamic balancing and shape synthesis of links in planar mechanisms”, *Mechanism and Machine Theory*, Elsevier Ltd, Vol. 93, pp. 127–146.
- Chaudhary, K. and Chaudhary, H. (2015b), “Optimal design of planar slider-crank mechanism using teaching-learning-based optimization algorithm”, *Journal of Mechanical Science and Technology*, Vol. 29 No. 12, pp. 5189–5198.
- Chaudhary, K. and Chaudhary, H. (2016), “Optimal dynamic design of planar mechanisms using teaching-learning-based optimization algorithm”, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 230 No. 19, pp. 3442–3456.
- Coello, C.A. (2000), “An updated survey of GA-based multiobjective optimization techniques”, *ACM Computing Surveys*, Vol. 32 No. 2, pp. 109–143.
- Coello, Coello, C. and Montes, E.M. (2001), “Use of dominance-based tournament selection to handle constraints in genetic algorithms”, *Intelligent Engineering Systems through Artificial Neural Networks*, Vol. 11, pp. 177–182.
- Danfelt, E.L., Hewes, S.A. and Chou, T.W. (1977), “Optimization of composite flywheel design”, *International Journal of Mechanical Sciences*, Vol. 19 No. 2, pp. 69–78.
- Darlow, M.S. (1989), *Balancing of High-Speed Machinery*.
- Deb, K. (1997), “GeneAS: A robust optimal design technique for mechanical component design.”, *In Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, Heidelberg., pp. 497–514.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, Vol. 6 No. 2, pp. 182–197.
- Dede, T. (2014), “Application of Teaching-Learning-Based-Optimization algorithm for the discrete optimization of truss structures”, *KSCE Journal of Civil Engineering*, Vol. 18 No. 6, pp. 1759–1767.
- Demeulenaere, B., Verschuur, M., Swevers, J. and De Schutter, J. (2009), “A General and Numerically Efficient Framework to Design Sector-Type and Cylindrical Counterweights for Balancing of Planar Linkages”, *Journal of Mechanical Design*, Vol. 132 No. 1, p. 011002.
- Ebrahimi, N.D. (1988), “Optimum design of flywheels”, *Computers and Structures*, Vol. 29 No. 4, pp. 681–686.

- Eby, D., Averill, R.C., Iii, W.F.P. and Goodman, E.D. (1999), "The Optimization of Flywheels using an Injection Island Genetic Algorithm", *AI EDAM*, Vol. 13 No. 5, pp. 327–340.
- Elliott, J.L. and Tesar, D. (1977), "The Theory of Torque, Shaking Force, and Shaking Moment Balancing of Four Link Mechanisms", *Journal of Engineering for Industry*, Vol. 99 No. 3, p. 715.
- Esat, I. and Bahai, H. (1999), "Theory of complete force and moment balancing of planer linkage mechanisms", *Mechanism and Machine Theory*, Vol. 34 No. 6, pp. 903–922.
- Feng, G. (1990), "Complete shaking force and shaking moment balancing of 26 types of four-, five- and six-bar linkages with prismatic pairs", *Mechanism and Machine Theory*, Vol. 25 No. 2, pp. 183–192.
- Feng, L., Mao, Z.-Z. and Yuan, P. (2012), "An improved multi-objective particle swarm optimization algorithm and its application", *Kongzhi Yu Juece/Control and Decision*, Vol. 27 No. 9.
- Foiles, W.C., Allaire, P.E. and Gunter, E.J. (1998), "Review: Rotor balancing", *Shock and Vibration*, Vol. 5 No. 5-6, pp. 325–336.
- FU, J.F., Fenton, R.G. and Cleghorn, W.L. (1991), "a Mixed Integer-Discrete-Continuous Programming Method and Its Application To Engineering Design Optimization", *Engineering Optimization*, Vol. 17 No. 4, pp. 263–280.
- Garvie, D.W. and Welbank, P.J. (1967), "A rasp-drum laboratory thresher and cleaner", *Journal of Agricultural Engineering Research*, Vol. 12 No. 3, pp. 229–232.
- Geoffrion, A.M. (1974), "Lagrangean relaxation", *Mathematical Programming Studies*, Vol. 2, pp. 82–114.
- Ghaly, A.E. (1985), "A stationary threshing machine: design, construction and performance evaluation", *AMA, Agricultural Mechanization in Asia*, Vol. 16 No. 3, pp. 19–30.
- Ghotbi, E. and Dhingra, A.K. (2012), "A bilevel game theoretic approach to optimum design of flywheels", *Engineering Optimization*, Vol. 44 No. 11, pp. 1337–1350.
- Goodman, T.P. (1964), "A Least-Squares Method for Computing Balance Corrections", *Journal of Engineering for Industry*, Vol. 86 No. 3, p. 273.
- Gorial, B.Y. and O'callaghan, J.R. (1991a), "Separation of particles in a horizontal air stream", *Journal of Agricultural Engineering Research*, Vol. 49, pp. 273–284.

- Gorial, B.Y. and O'callaghan, J.R. (1991b), "Separation of grain from straw in a vertical air stream", *Journal of Agricultural Engineering Research*, Vol. 48, pp. 111–122.
- Gosselin, C.M., Moore, B. and Schicho, J. (2009), "Dynamic balancing of planar mechanisms using toric geometry", *Journal of Symbolic Computation*, Vol. 44 No. 9, pp. 1346–1358.
- Guo, C.-X., Hu, J.-S., Ye, B. and Cao, Y.-J. (2004), "Swarm intelligence for mixed-variable design optimization.", *Journal of Zhejiang University. Science*, Vol. 5 No. 7, pp. 851–60.
- Haines, R.S. (1981), "Minimum RMS shaking moment or driving torque of a force-balanced 4-bar linkage using feasible counterweights", *Mechanism and Machine Theory*, Vol. 16 No. 3, pp. 185–195.
- He, S., Prempan, E. and Wu, Q. (2004), "An improved particle swarm optimizer for mechanical design optimization problems", *Engineering Optimization*, Vol. 36 No. 5, pp. 585–605.
- Huang, J. and Fadel, G.M. (2000), "Heterogeneous flywheel modeling and optimization", *Materials & Design*, Vol. 21 No. 2, pp. 111–125.
- Huang, J., Fadel, G.M., Blouin, V.Y. and Grujicic, M. (2002), "Bi-objective optimization design of functionally gradient materials", Vol. 23, pp. 657–666.
- Huang, N.C. (1983), "Equipomental system of rigidly connected equal particles", *Journal of Guidance, Control, and Dynamics*, Vol. 16 No. 6, pp. 1194–1196.
- Jeet, V. and Kutanoglu, E. (2007), "Lagrangian relaxation guided problem space search heuristics for generalized assignment problems", *European Journal of Operational Research*, Vol. 182 No. 3, pp. 1039–1056.
- Jiang, L. and Wu, C.W. (2017a), "Topology optimization of energy storage flywheel", *Structural and Multidisciplinary Optimization*, Vol. 55 No. 5, pp. 1917–1925.
- Jiang, L. and Wu, C.W. (2017b), "Topology optimization of energy storage flywheel", *Structural and Multidisciplinary Optimization*, Vol. 55 No. 5, available at:<https://doi.org/10.1007/s00158-016-1576-1>.
- Jiang, L., Zhang, W., Ma, G.J. and Wu, C.W. (2016), "Shape optimization of energy storage flywheel rotor", available at:<https://doi.org/10.1007/s00158-016-1516-0>.
- Joshi, H.C. (1981), "Design and Selection of Thresher Parameters and Components", *AMA, Agricultural Mechanization in Asia*, Vol. 12, pp. 61–70.

- Kamenskii, V.A. (1968), "On the question of the balancing of plane linkages", *Journal of Mechanisms*, Vol. 3 No. 4, pp. 303–322.
- Kharab, A. and Guenther, R.B. (2012), *An Introduction to Numerical Methods: A MATLAB Approach*, CRC Press.
- Kitayama, S., Arakawa, M. and Yamazaki, K. (2006), "Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization", *Structural and Multidisciplinary Optimization*, Vol. 32 No. 3, pp. 191–202.
- Knospe, C.R., Hope, R.W., Tamer, S.M. and Fedigan, S.J. (1996), "Robustness of adaptive unbalance control of rotors with magnetic bearings", *JVC/Journal of Vibration and Control*, Vol. 2 No. 1, pp. 33–52.
- Kochev, I.S. (1987), "General method for full force balancing of spatial and planar linkages by internal mass redistribution", *Mechanism and Machine Theory*, Vol. 22 No. 4, pp. 333–341.
- Kochev, I.S. (2000), "General theory of complete shaking moment balancing of planar linkages: A critical review", *Mechanism and Machine Theory*, Vol. 35 No. 11, pp. 1501–1514.
- Kress, G.R. (2000), "Shape optimization of a flywheel", Vol. 1, pp. 74–81.
- Kripka, M. (2004), "Discrete optimization of trusses by simulated annealing", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 26 No. 2, pp. 170–173.
- Kyu Ha, S., Kim, D.J. and Sung, T.H. (2001), "Optimum design of multi-ring composite flywheel rotor using a modified generalized plane strain assumption", *International Journal of Mechanical Sciences*, Vol. 43 No. 4, pp. 993–1007.
- Lampinen, J. and Zelinka, I. (1999), "Mixed Integer Discrete Continuous Optimization by Differential Evolution, Part 2: a practical example", *Proceedings of MENDEL '99, 5th International Mendel Conference on Soft Computing*, pp. 77–81.
- Lee, T.W. and Cheng, C. (1984), "Optimum Balancing of Combined Shaking Force, Shaking Moment, and Torque Fluctuations in High-Speed Linkages", *Journal of Mechanisms Transmissions and Automation in Design*, Vol. 106 No. 2, p. 242.
- Leyffer, S. (2001), "Integrating SQP and Branch-and-Bound for Mixed Integer Nonlinear Programming", Vol. 18, pp. 295–309.

- Li, L.J., Huang, Z.B. and Liu, F. (2009), “A heuristic particle swarm optimization method for truss structures with discrete variables”, *Computers and Structures*, Elsevier Ltd, Vol. 87 No. 7-8, pp. 435–443.
- Li, Z. (1998), “Sensitivity and robustness of mechanism balancing”, *Mechanism and Machine Theory*, Vol. 33 No. 7, pp. 1045–1054.
- Loh, H.T. and Papalambros, P.Y. (1991), “A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems”, *Journal of Mechanical Design*, Vol. 113 No. 3, p. 325.
- Lowen, G.G. and Berkof, R.S. (1971), “Determination of Force-Balanced Four-Bar Linkages With Optimum Shaking Moment Characteristics”, *Journal of Engineering for Industry*, Vol. 93 No. 1, p. 39.
- Lowen, G.G., Tepper, F.R. and Berkof, R.S. (1974), “The quantitative influence of complete force balancing on the forces and moments of certain families of four-bar linkages”, *Mechanism and Machine Theory*, Vol. 9 No. 3-4, pp. 299–323.
- Madan Lal, K. (2012), “An Improved Multi Crop Thresher”, Indian patent No. 253863.
- Mahdi, M. (2010), “An Optimal Two-Dimensional Geometry of Flywheel for Kinetic Energy Storage”, *International Journal of Thermal and Environmental Engineering*, Vol. 3 No. 2, pp. 67–72.
- Marler, R.T. and Arora, J.S. (2004), “Survey of multi-objective optimization methods for engineering”, *Structural and Multidisciplinary Optimization*, Vol. 26 No. 6, pp. 369–395.
- Marler, R.T. and Arora, J.S. (2010), “The weighted sum method for multi-objective optimization: New insights”, *Structural and Multidisciplinary Optimization*, Vol. 41 No. 6, pp. 853–862.
- Messenger, T. and Pyrz, M. (2013), “Discrete optimization of rigid rotor balancing”, *Journal of Mechanical Science and Technology*, Vol. 27 No. 8, pp. 2231–2236.
- Metwalli, S.M., Shawki, G.S.A. and Sharobeam, M.H. (1983), “Optimum Design of Variable-Material Flywheels”, *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105 No. 82, pp. 249–253.
- Mortenson, M.E. (2006), *Geometric Modeling*, McGraw Hill Education Private Limited, New Delhi, India.
- Mundo, D., Gatti, G. and Dooner, D.B. (2009), “Optimized five-bar linkages with non-circular gears for exact path generation”, *Mechanism and Machine Theory*, Vol. 44 No. 4, pp. 751–760.

- Nema, S., Goulermas, J., Sparrow, G. and Cook, P. (2008), “A hybrid particle swarm branch-and-bound (HPB) optimizer for mixed discrete nonlinear programming”, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, Vol. 38 No. 6, pp. 1411–1424.
- Olaoye, J.O., Oni, K.C. and Mary M Olaoye. (2010), “Computer applications for selecting operating parameters of stationary grain crop thresher”, *International Journal of Agricultural and Biological Engineering*, Vol. 3 No. 3, pp. 8 – 18.
- Parsopoulos, K.E. and Vrahatis, M.N. (2002), “Particle swarm optimization method in multiobjective problems”, *2002 ACM Symposium on Applied Computing (SAC 2002)*, pp. 603–607.
- Pathak, V.K., Singh, A.K., Singh, R. and Chaudhary, H. (2017), “A modified algorithm of Particle Swarm Optimization for form error evaluation”, *Technisches Messen*, Vol. 84 No. 4, available at: <https://doi.org/10.1515/teme-2016-0040>.
- Pedrolli, L., Zanfei, A., Ancellotti, S. and Fontanari, V. (2016), “Shape optimization of a metallic flywheel using an evolutive system method : Design of an asymmetrical shape for mechanical interface”, Vol. 0 No. 0, pp. 1–14.
- Pennestri, E. and Qi, N.M. (1991), “Optimum balancing of four-bar linkages”, *Mechanism and Machine Theory*, Vol. 26 No. 3, pp. 337–348.
- Prashad, M. and Sharma, K. (1985), “Bhagirath Mal Mahaveer Parshad agricultural machinery manufacturer”, Jaipur, India.
- R. S. Berkof and G. G. Lowen. (1969), “A new Method for Complete!! Force Balancing Simple Linkages”, *ASME Journal of Engineering for Industry*, Vol. 91 No. 1, pp. 21–26.
- Rajeev, S. and Krishnamoorthy, S.C. (1992), “Discrete optimization of structures using genetic algorithms”, Vol. 118 No. 5, pp. 1233–1250.
- Rao, R. V. and Savsani, V.J. (2012), *Mechanical Design Optimization Using Advanced Optimization Techniques*, Springer Science & Business Media.
- Rao, R. V., Savsani, V.J. and Vakharia, D.P. (2011), “Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems”, *CAD Computer Aided Design*, Elsevier Ltd, Vol. 43 No. 3, pp. 303–315.
- Rao, R.V. (2015), *Teaching Learning Based Optimization Algorithm And Its Engineering Applications*, Springer.

- Rao, R.V. (2016), “Jaya : A simple and new optimization algorithm for solving constrained and unconstrained optimization problems”, Vol. 7, pp. 19–34.
- Rao, R.V., Rai, D.P. and Balic, J. (2017), “A multi-objective algorithm for optimization of modern machining processes”, *Engineering Applications of Artificial Intelligence*, Elsevier Ltd, Vol. 61, pp. 103–125.
- Rao, R.V., Rai, D.P. and Balic, J. (2018), “Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm”, *Journal of Intelligent Manufacturing*, Vol. 29 No. 8, pp. 1715–1737.
- Rao, R.V., Rai, D.P. and Balic, J. (2019), “Multi-objective optimization of abrasive waterjet machining process using Jaya algorithm and PROMETHEE Method”, *Journal of Intelligent Manufacturing*, Vol. 30 No. 5, pp. 2101–2127.
- Rao, R.V. and Saroj, A. (2016), “Multi-objective design optimization of heat exchangers using elitist-Jaya algorithm”, *Energy Systems*, Springer Berlin Heidelberg, available at:<https://doi.org/10.1007/s12667-016-0221-9>.
- Rao, R.V.R. (2019), *Jaya: An Advanced Optimization Algorithm and Its Engineering Applications*, Springer International Publishing AG, part of Springer Nature, available at:<https://doi.org/doi.org/10.1007/978-3-319-78922-4>.
- Rao, S.S. and Xiong, Y. (2005), “A Hybrid Genetic Algorithm for Mixed-Discrete Design Optimization”, *Journal of Mechanical Design*, Vol. 127 No. 6, p. 1100.
- Rao, R. V., More, K.C., Coelho, L.S. and Mariani, V.C. (2017), “Multi-objective optimization of the Stirling heat engine through self-adaptive Jaya algorithm”, *Journal of Renewable and Sustainable Energy*, Vol. 9 No. 3, p. 033703.
- Routh, E.J. (1905), *A Treatise on The Dynamics of A System of Rigid Bodies, Elementary Part I*, Dover Publication Inc., New York.
- Sandgren, E. (1990), “Nonlinear Integer and Discrete Programming in Mechanical Design Optimization”, *Journal of Mechanical Design*, Vol. 112 No. 2, p. 223.
- Sandgren, E. and Ragsdell, K.M. (2016), “Optimal Flywheel Design With a General Thickness Form Representation”, Vol. 105 No. September 1983, pp. 425–433.
- Sherwood, A.A. and Hockey, B.A. (1969), “The optimisation of mass distribution in mechanisms using dynamically similar systems”, *Journal of Mechanisms*, Vol. 4 No. 3, pp. 243–260.

- Shin, D.K., Gurdal, Z. and Griffin, O.H. (1990), “A Penalty approach for nonlinear optimization with discrete design variables”, *Engineering Optimization*, Vol. 16 No. 1, pp. 29–42.
- Shiyu Zhou and Jianjun Shi. (2001), “Active Balancing and Vibration Control of Rotating Machinery: A Survey”, *The Shock and Vibration Digest*, Vol. 33 No. 5, pp. 361–371.
- Simonyan, K.J. and Yiljep, Y.D. (2008), “Investigating Grain Separation and Cleaning Efficiency Distribution of a Conventional Stationary Rasp-bar Sorghum Thresher”, *Agricultural Engineering International: CIGR Journal*, Vol. X, pp. 1–13.
- Singh, P. and Chaudhary, H. (2019), “Optimum two-plane balancing of rigid rotor using discrete optimization algorithm”, *World Journal of Engineering*, Vol. 16 No. 1, pp. 138–146.
- Singh, R., Chaudhary, H. and Singh, A.K. (2017), “Defect-free optimal synthesis of crank-rocker linkage using nature-inspired optimization algorithms”, *Mechanism and Machine Theory*, Vol. 116, pp. 105–122.
- Singh, S. (1978), *Faujdar Engineering Works*, Bharatpur, Rajasthan, India.
- Sonmez, M. (2011), “Discrete optimum design of truss structures using artificial bee colony algorithm”, *Structural and Multidisciplinary Optimization*, Vol. 43 No. 1, pp. 85–97.
- Stout, B.A. and Cheze, B. (1999), *CIGR Handbook of Agricultural Engineering Volume III*, Vol. III, ASAE, St. Joseph, USA.
- Sun, C., Zeng, J. and Pan, J.S. (2011), “A modified particle swarm optimization with feasibility-based rules for mixed-variable optimization problems”, *International Journal of Innovative Computing, Information and Control*, Vol. 7 No. 6, pp. 3081–3096.
- Takahashi, K., Kitade, S. and Morita, H. (2001), “Development of high speed composite flywheel rotors for energy storage systems”, *Advanced Composite Materials: The Official Journal of the Japan Society of Composite Materials*, Vol. 11 No. 1, pp. 41–50.
- Tan, J. and Harrison, H.P. (1987), “Screening Efficiency for Planar and Spatial Drive Mechanisms”, *TRANSACTIONS of the ASAE*, Vol. 30 No. 5, pp. 1242–1246.
- Test Code IS:6284. (1985), “Test Code for power thresher for cereals”, *Indian Standard*.

- Timoshenko, S.P. and Goodier, J.N. (1970), *Theory of Elasticity*, 3rd edn., McGraw-Hill.
- Tricamo, S.J. and Lowen, G.G. (1983a), “A Novel Method for Prescribing the Maximum Shaking Force of a Four-Bar Linkage with Flexibility in Counterweight Design”, *Journal of Mechanisms Transmissions and Automation in Design*, Vol. 105 No. 3, p. 511.
- Tricamo, S.J. and Lowen, G.G. (1983b), “Simultaneous Optimization of Dynamic Reactions of a Four-Bar Linkage With Prescribed Maximum Shaking Force”, *Journal of Mechanisms Transmissions and Automation in Design*, Vol. 105 No. 3, p. 520.
- Tzeng, J., Emerson, R. and Moy, P. (2006), “SCIENCE AND Composite flywheels for energy storage”, Vol. 66, pp. 2520–2527.
- V.A.Shchepetil'nikov. (1968), “The determination of the mass centers of mechanisms in connection with the problem of mechanism balancing”, *Journal of Mechanisms*, Vol. 3, pp. 367–389.
- Varshney, A.C. (2004), *Data Book for Agricultural Machinery Design*, Central Institute of Agricultural Engineering, Bhopal, India.
- Van de Vegte, J. (1981), “Balancing of flexible rotors during operation.”, *Journal of Mechanical Engineering Science*, Vol. 23, pp. 257–261.
- Venkata Rao, R., Saroj, A. and Bhattacharyya, S. (2018), “Design Optimization of Heat Pipes Using Elitism-Based”, *Journal of Thermophysics and Heat Transfer*, Vol. 32 No. 3, pp. 1–11.
- Venkata Rao, R. and Waghmare, G.G. (2016), “A new optimization algorithm for solving complex constrained design optimization problems”, *Engineering Optimization*, Vol. 0273 No. April, pp. 1–24.
- Walker, M.J. and Oldham, K. (1978), “A general theory of force balancing using counterweights”, *Mechanism and Machine Theory*, Vol. 13 No. 2, pp. 175–185.
- Wenglarz, R.A., Fogarasy, A.A. and Maunder, L. (1969), “Simplified Dynamic Models”, *Engineering*, Vol. 208, pp. 194–195.
- Wu, S.-J. and Chow, P.-T. (1995), “Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems Via Meta-Genetic Parameter Optimization”, *Engineering Optimization*, Vol. 24 No. 2, pp. 137–159.

- Ye, Z. and Smith, M.R. (1994), "Complete balancing of planar linkages by an equivalence method", *Mechanism and Machine Theory*, Vol. 29 No. 5, pp. 701–712.
- Zeid, I. (1991), *CAD/CAM Theory and Practice*, McGraw-Hill Higher Education.
- Zhang, C. and Wang, H.P. (BEN). (1993), "Mixed-Discrete Nonlinear Optimization With Simulated Annealing", *Engineering Optimization*, Vol. 21 No. 4, pp. 277–291.

Appendix A

MATLAB Code for Non-Dominated Sorting Jaya Algorithm (NSJAYA)

NSJAYA is a posteriori approach based optimization algorithm and optimize the objective functions simultaneously. The MATLAB code of the NSJAYA algorithm is presented here.

```
clc; close all; clear all;
max_run=10;
for run=1:max_run
xl_temp= []; ub=[];lb=[];
pop=[]; % size of population
%objective function is considered 0 for minimization and 1 for maximization%
min= [0 0];
UB=ub; %upper limit on design variables
LB=lb; %lower limit on design variables
dim=numel (LB); % number of design variables
n=numel (min); %number of objective Function
%Algorithm start here%
%% Initialize population
for i=1:dim
    cxl(:,i)=LB(i)+(UB(i)-LB(i))*rand(pop,1);
end
xl_new=quasi(LB,UB,xl);
xl=[xl;xl_new];
xl(:,dim+1:dim+n)=objective(xl);
xl_temp =xl;
xl=[]; b=[];
%Find out Non_dominated solutions%
xl=nd(n,dim,temp_xl,min);
xl=xl(1:pop,:);
xl_temp =[];
for iter=1:max_iteration;
    a=[];
    b=[];
```

```

a= xl(:,dim+n+1)==1;
b=find(xl(a,dim+n+2)==100);
count=numel(b);
% Updation of solutions%
xl_temp =updation(xl(:,1:dim),count);
%handling boundary violations%
xl_temp =boundary (LB, UB, xl_temp);
xl_temp (:,dim+1:dim+n) =objective (xl_temp);
comb_xl=cat (1, xl (:, 1:dim+n), xl_temp);
xl_temp = []; xl= [];
xl=nd(n,dim,comb_xl,min);
xl=xl(1:pop,:);
comb_xl=[];
xl_temp =quasi (LB,UB,xl(:,1:dim));
xl_temp (:,dim+1:dim+n)=objective(xl_temp);
comb_xl=cat(1,xl(:,1:dim+n), xl_temp);
xl_temp =[];
xl=[];
xl= nd(n,dim,comb_xl,min);
xl=xl(1:pop,:);
comb_xl=[];
xl=cat(2,xl,constraint(xl(:,1:dim)));
end
end
%Plotting of Pareto front %
figure(1);
plot(xl(:,dim+1),xl(:,dim+2),'*');
xlabel('objective_1');
ylabel('objective_2');

```

MATLAB code for modified Jaya algorithm

A modified Jaya algorithm handles all types of variables. The MATLAB code of the modified Jaya algorithm is illustrated through the discrete two-plane balancing of the threshing drum.

```

clc;clear all;close all
tic % Start Stopwatch
%%Discrete set for design variables%%
mij=[0.0 .010 .020 .050 .100 .200 .300];
phij=[20:20:360];
%Upper and lower Bound on design variables%
LB=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 20 20 20 20 20 20];
UB=[0.300 0.300 0.300 0.300 0.300 0.300 360 360 360 360 360 360];
%% number of variables and population size%%
variables =12;
population_size =300;
maxrun=20;
%%initialize the population size %%
for run=1:maxrun
for i=1: variables
for j=1: population_size
x0(i,j)=LB(j)+rand()*(UB(j)-LB(j));
end
end
%%discrete operator%%
for n=1: population_size
for l=1: size(x0,2)-6
for s=1:size(mij,2)-1
if mij(s)<x0(n,l)&& x0(n,l)<mij(s+1)
if x0(n,l)<(mij(s)+mij(s+1))/2
x0(n,l)=mij(s);
else
x(n,l)=mij(s+1);

```

```

end
end
if x0(n,l)==0
x0(n,l)=mij(s+1);
end
end
end
for o=7:12
for p=1:size(phi_j,2)-1
if phi_j(p)<x0(n,o)&& x0(n,o)<phi_j(p+1)
if x0(n,o)<(phi_j(p)+phi_j(p+1))/2
x0(n,o)=phi_j(p);
else
x0(n,o)=phi_j(p+1);
end
end
end
end
end
end
%% set maximum number of iteration%%
maxite=3000;
for i=1:n
f0(i,:)=objective_fun(x0(i,:));
end
f3=f0(:,3); [fbest,indexb]=min(f3); [fworst,indexw]=max(f3);
% initial Best and initial worst
best=x0(indexb,:); worst=x0(indexw,:);
% % Jaya algorithm-----start
for k=1:maxite
for i=1: variables
for j=1: population_size
% update the design variabe
x(i,j)=(x0(i,j)+rand()*(best(1,j)-abs(x0(i,j))))-rand()*(worst(1,j)-abs(x0(i,j))));
if x(i,j)<LB(j)

```

```

x(i,j)=LB(j);
elseif x(i,j)>UB(j)
x(i,j)=UB(j);
end
end
end
%% discrete operator %%
for n=1: population_size
for l=1: size(x0,2)-6
for s=1:size(mij,2)-1
if X(s)<x(n,l)&& x(n,l)<X(s+1)
if x(n,l)<(X(s)+X(s+1))/2
x(n,l)=X(s);
else
x(n,l)=X(s+1);
end
end
if x(n,l)==0
x(n,l)=X(s+1);
end
end
end
for o=7:12
for p=1:size(phij,2)-1
if x1(p)<x(n,o)&& x(n,o)<x1(p+1)
if x(n,o)<(x1(p)+x1(p+1))/2
x(n,o)=x1(p);
else
x(n,o)=x1(p+1);
end
end
end
end
end
end

```

```

for i=1:n
f1(i,:)=objective_fun(x(i,:));
end
f2=f1(:,3);
for i=1:n
if f2(i,1)<f3(i,1);
f3(i,1)=f2(i,1);
x0(i,:)=x(i,:);
end
end
fmen=mean(f3); fmen(k,run)=fmen;
% finding out the best and worse
[fbest1,indexB]=min(f3); [fworst1,indexW]=max(f3);
% updating gbest and best fitness
if fbest1<fbest
fbest=fbest1;
best=x0(indexB,:);
end
if fworst1<fworst
worst=x0(indexW,:);
end
end
sol=objective_fun(best); f_value=sol(:,3); fff(run)=f_value; rbest(run,:)=best;
end
[best_function,best_run]=min(fff);
best_variables=rbest(best_run,:);
design=objective_fun(best_variables);

```

Papers published based on this work

Journal Publications

1. Prem Singh and Himanshu Chaudhary, "Optimal shape synthesis of a metallic flywheel using non-dominated sorting Jaya algorithm", *Soft Computing*, Springer, (2019), <https://doi.org/10.1007/s00500-019-04302-x>. **(SCI)**
2. Prem Singh and Himanshu Chaudhary," A Modified Jaya Algorithm for Mixed Variable Optimization Problems", DOI: 10.1515/jisys-2018-0273, *Journal of intelligent system*, 2018. **(SCOPUS/ESCI)**
3. Prem Singh and Himanshu Chaudhary," Optimal design of the flywheel using nature inspired optimization algorithms", *Open Agriculture*, 3(1), 490-499. **(SCOPUS/ESCI)**
4. Prem Singh, Himanshu Chaudhary, "Optimum two-plane balancing of rigid rotor using discrete optimization algorithm" *World Journal of Engineering* <https://doi.org/10.1108/WJE-05-2018-0167>, 2019. **(SCOPUS/ESCI)**
5. Prem Singh and Himanshu Chaudhary, "Dynamic balancing of cleaning unit used in agricultural thresher using Jaya algorithm", *World Journal of Engineering*, Emerald, 2019 **(SCOPUS/ESCI)**

Submitted

1. Prem Singh and Himanshu Chaudhary, "Optimum discrete balancing of the threshing drum using a Jaya algorithm", *Mechanics Based Design of Structures and Machines* (Under Review).
2. Prem Singh and Himanshu Chaudhary, "Dynamic balancing of the cleaning unit used in agricultural thresher using a non-dominated sorting Jaya algorithm", *Engineering Computations* (Under Review).

Conference Publications

1. Prem Singh and Himanshu Chaudhary, "A Jaya Algorithm for Discrete Optimization Problems." In *Engineering Vibration, Communication and Information Processing*, Springer, Singapore, pp. 197-203, 2018, DOI: 10.1007/978-981-13-1642-5_18.
2. Prem Singh and Himanshu Chaudhary, " Shape optimization of the flywheel using the cubic B spline curve, *Advances in Engineering Design*, Springer, Singapore, 805-813, 2019.

Brief Bio-data of the author

Name : **Prem Singh**
Date of birth : July 20, 1984
Address for correspondence : Plot-17, Flat-G1, Shri Shyam Apartment,
Keshar Vihar Colony, Jagatpura, Jaipur

Education:

Degree	Discipline	Institute/Board/ University	Percentage/ CGPA	Year	Division
Secondary	General	RBSE	63.50	1999	First
Senior Secondary	PCM	RBSE	68.31	2001	First
B.E.	Mechanical Engineering	Govt. Engineering College, Bikaner (University of Rajasthan, Jaipur)	63.10	2006	First
M.Tech.	Mechanical Engineering (Design)	S.V. National Institute of Technology, Surat	8.19	2009	--
Ph.D.	Mechanical Engineering	Malaviya National Institute of Technology, Jaipur	9.0	2019	--

Teaching Experience:

Organization	Designation	Service Period
NC College of Engineering, Israna, Panipat, Haryana	Lecturer	September ,2009- July ,2012
JECRC University, Jaipur	Assistant Professor	July, 2012- July, 2019
SKIT, Jaipur	Associate Professor	July, 2019-Onwards