Development and Correlation Analysis of Non-Dominated Sorting Buffalo Optimization NSBUF II Using Taguchi's Design Coupled Gray Relational Analysis and ANN

Tamal Ghosh[1]*, Kristian Martinsen[2], Pranab K Dan[3]

[1, 2] The Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, Teknologivegen 22, 2815 Gjøvik, Norway

[3] RMSoEE, Indian Institute of Technology, Kharagpur 721302, India

Email: tamal.ghosh@ntnu.no [1], kristian.martinsen@ntnu.no [2], pkdan@see.iitkgp.ernet.in [3]

*Corresponding Author

Abstract

African Buffalo Optimization (ABO) is a latest bio-inspired optimization technique in the domain of evolutionary optimization, which mimics the migratory behavior of the buffalo foraging for food across the plains and forests. The ABO is, by now, recognized as a single-objective optimization algorithm, comprising the ability to solve both, the continuous and discrete optimization problems. However, a multi-objective version of ABO could be more useful for industrial problems. An aim is made in this article to develop the multi-objective variant of ABO, namely NSBUF II, which incorporates Pareto search for non-dominated solutions in the state space and a local search module for faster convergence. Selection of parameters for the NSBUF II is extremely sensitive to the obtained Pareto fronts. Thus, a Grey Relational Analysis (GRA) coupled with Taguchi's L_{16} orthogonal array is adopted, which efficiently obtains the best set of parameters for the NSBUF II. Initially the proposed NSBUF II is tested using utilization based bi-objective production cell design problem and compared with published Multi-Objective Particle Swarm Optimization (MOPSO), and Non-dominated Sorting Genetic Algorithm (NSGA II) successfully. To analyse the performance of the NSBUF II, Self-Organizing Map (SOM) is applied, which is a powerful tool for visualizing the high-dimensional data in low dimensional maps. Applied SOM visually reveals the hidden correlational structure among the design parameters and the objective space. The performance of the NSBUF II is validated statistically. NSBUF II is further verified with a real-world case obtained based on the Abrasive Water Jet Machining (AWJM) process. Validation test proves the competence of the proposed NSBUF II for real-world problem solving. The contribution of this paper is threefold. First, a novel multi-objective algorithm NSBUF II is developed. Second, a SOM based visual analysis is proposed to visualize the correlation among design parameters and Pareto fronts. Third, the NSBUF II is employed to solve a combinatorial production cell design problem followed by a real-world industrial problem.

Keywords: African Buffalo Optimization; Non-dominated Sorting; Multi-objective Production Cell Design; Self-Organizing Map; Grey Relational Analysis; Abrasive Water Jet Machining

1. Introduction

Since past few decades, most of the domains of scientific research are exploring computer-based programming to develop software tools to perform tedious tasks or solve complex mathematical problems. This trend essentially eliminates the unnecessary human hard work in research while enhancing the productivity [1]. Among all the computer-based programming techniques, optimization algorithms are substantially critical due to the nature of their applications. For that matter, research on optimization algorithms turn out to be one of the preferred areas for scientific explorations, which undoubtedly signifies the development of numerous optimization algorithms. Mostly investigated methods among the optimization algorithms are, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Bat Inspired Algorithm (BA), Bacterial Foraging Optimization (BFO), Cuckoo Search, Firefly Algorithm, Grey Wolf Optimization (GWO) etc. However, these algorithms are not flawless, and exist with limitations such as early or slower convergence, tendency to be trapped in local optima, or having too many parameters to be set up [2]. These algorithms fluctuate in their functioning depending upon different operational approaches. However, they perform some significant trade-offs between global solution search and manipulation in local search. As these techniques grounded on natural phenomena, therefore none of these algorithms is perfectly suitable for every problem domain. Thus, new algorithms are being developed and existing algorithms are being modified or extended to disseminate further investigations [3].

African Buffalo Optimization (ABO) is one such optimization algorithm recently developed [4]. ABO is classified as a swarm-based nature-inspired optimization technique initially developed for the continuous domain. This technique mimics the migratory behavior of the large herd of buffalo foraging for food across the plains and forests. They migrate in search of green fields and travel across vast territory. Buffalo herd is generally led by one wise and old leader who determines where the herd would move. The leader Buffalo is capable enough to predict the monsoon, trace the green meadows, and directs the herd to it. Depending upon the seasonal differences the buffalo herd travel continuously across the plains, mountains, and forests in quest of green land. The ABO algorithm exploits the self-organizing attitude of buffalo entities while searching for the optimality in vast state space. ABO advances with two natural inter-communication strategies among the herd members, (i) the roaring *waaa* (grunts), which indicates the sense of risks or scarcity of greenery in present location and directs the herd to move to the next location. (ii) The roaring *maaa* (mumbles), which confirms promising green meadows and suggesting the herd to explore locally [4]. Researchers successfully practiced ABO in different problem domains, such as Travelling Salesman Problem (TSP) [5], parameter optimization of PID controller [6] etc.

In the present study, the ABO is considered as a research methodology and further extended as a posterior multi-objective optimization algorithm. A posterior multi-objective optimization attains solutions for problems that consider more than one objective. Unlike the single objective optimization problems, a multi-objective optimization problem can have conflicting objectives that yield multiple solutions, which can trade-off among objectives. The primary goal is set to determine the set of optimal trade-off points among the objectives to achieve Pareto fronts [7]. The most prevalent algorithms of this kind are, non-dominated sorting GA (NSGA II) [8] and multi-objective PSO (MOPSO) [9] practiced in different domains of research in past. Recently most of the single objective optimization algorithms are being extended to solve the multi-objective problems, such as, Multi-objective ACO [11], Multi-objective Bat Algorithm (NSBAT II) [7], Multi-objective Cuckoo Search (MOCS) [10] etc. In this study a multi-objective extension of the recently published ABO, namely NSBUF II is proposed. Parameter selection for ABO is substantially critical while reaching the global optima. Therefore, a comprehensive method based on the Grey Relational Analysis (GRA) coupled with Taguchi's orthogonal design is demonstrated to select the optimal set of parameters. A powerful visualization tool called Self-Organizing Map (SOM) is applied further. SOM visually reveals the correlational structures among the design parameters and objectives. Thereafter a latest multi-objective combinatorial problem based on the utilization-based production cell design is solved successfully using the NSBUF II. The effectiveness of the NSBUF II could be shown using a comparative discussion performed using similar techniques like MOPSO and NSGA II. The results are obtained, and Pareto fronts are checked against NSGA II and MOPSO. Finally, the NSBUF II is also evaluated using a real-world multi-response manufacturing process optimization problem using Abrasive Water Jet Machining (AWJM). The obtained Pareto solutions are validated correctly. Since this article has considered several tools and techniques while developing and analysing the NSBUF II algorithm and its performance, therefore various research papers are reviewed and presented sub-section wise in the next section. The rest of the paper is organized in the following manner; section 2 portrays the background work, problem model, and NSBUF II algorithm. Section 3 discusses the execution of NSBUF II on the data for production cell design, the SOM visualization for correlation, and the real-world case of multi-response AWJM process optimization. Finally, Section 4 concludes the study and proposes the future works.

2. African Buffalo Optimization (ABO)

ABO is a latest optimization algorithm in the cluster of bio-inspired algorithms, which is proposed in 2015 [4]. Few applications are already performed in the area of combinatorial optimization [5, 6]. ABO is developed for the single objective problems and multi-objective model is not yet developed. The aim of this paper is to seal that research gap. ABO is inspired from the migratory behavior of buffalo. Buffalo forage for food across the plains and forests of vast African land. They migrate in search of green fields and travel across thousands of miles in a group called herd containing few hundreds of the animals. The herd leader is experienced enough to sense the seasonal changes and directs the herd to move constantly across the plains, mountains, and forests in quest of greenery. The movement of the herd is portrayed in Figure 1. The ABO algorithm simulates the self-organizing approach of buffalo entities while searching for the optimality in the vast state space. Let the ABO advances with the i^{th} buffalo and its two natural inter-communication methods among the herd members ($i=1,2,3,\ldots,n$), (i) the roaring *waaa* (grunts), denoted by w_i, which helps in exploring the solutions in global search. (ii) The roaring *maaa* (mumbles), denoted by m_i, which helps in exploiting in the local search space. The values of m and w for $(i+1)^{th}$ buffalo are updated using the following equations,

$$m_{i+1} = m_i + lp_1 \times (bg - w_i) + lp_2 \times (bp_i - w_i) \quad (1)$$

$$w_{i+1} = \frac{(w_i + m_i)}{\lambda} \quad (2)$$

Where lp_1 and lp_2 are learning parameters, bg and bp_i are the best position of the herd and the i^{th} buffalo's best-known position respectively, λ is a prefixed random number.

Figure 1. Migration of buffalo herd following the herd leader
(https://pixels.com/featured/aerial-view-of-a-herd-of-african-beverly-joubert.html)

The artificial herd of buffalo is generated with initial herd $x_i = \{x_1, x_2, \ldots, x_n\}_{i\in[1,n]}$ representing the n coordinates for continuous domain or n vectors for discrete domain. Eq. (1) can be decomposed in three sub expressions. (1) *Memory expression:* It updates the former position of the buffalo from m_i to m_{i+1}. (2) *Explorative expression:* The simulated buffalo are good at mutual communications, which can fetch the best position of the herd in every iteration using lp_1, bg and w_i, (3) *Exploitative expression*: buffalo can update their own best positions while comparing with their present positions. Further Eq. (2) defines how the herd would move to a new location depending on Eq. (1). The pseudocode of ABO is provided in algorithm 1.

Algorithm 1: ABO

Step 1. Define the fitness function $f(x)$, $X=(x_1, x_2, \ldots, x_d)^T$
Step 2. Initialize the herd of buffalos with position x_i, $(i = 1, 2, \ldots, n)$
Step 3. Define the learning parameters $lp1, lp2, m, w, \lambda$
Step 4. Initialize bg and bp
Step 5. While (t< maximum number of iteration) do
Step 6. for each x_i in the herd do
Step 7. update the fitness values of buffalos using Eq. (1), (2)
Step 8. if $f(x_i)<bp_i$ then
Step 9. $bp_i = f(x_i)$
Step 10. end
Step 11. end
Step 12. if $bp_i < bg$ then
Step 13. $bg = bp_i$
Step 14. end
Step 15. end
Step 16. return the global best buffalo bg

2.1. Taguchi's Orthogonal Design and Grey Relational Analysis

Taguchi method is an essential tool for optimization of process parameters or experimental design variables, which keeps the process under control by managing variations while improving quality [12]. This approach is being used in selection of engineering and manufacturing process parameters heavily since past few decades. This approach also reduces the number of experimental runs substantially with the help of orthogonal design. For that matter a quality loss function could be employed, which controls the digression between the experimental and desired values of variables. This loss function is then converted into a signal-to-noise (S/N) ratio. Taguchi's design is suitable for single response or single objective design. For multi-objective design approach, the GRA has been developed which can exploit Taguchi's design [13] and estimates the degree of the correlation between experimental runs using grey relational grade (GRG) [14]. Steps of GRA are,

Step1: The data are normalized to reduce the inconsistency, which transforms the data values to be restricted in the range {0, 1}. When the performance objective is to be minimized smaller-the-better (Eq. (3) rule is applied, else larger-the-better (Eq. (4)) rule is applied,

$$y_i^*(x) = \frac{y_i^0(x)_{max} - y_i^0(x)}{y_i^0(x)_{max} - y_i^0(x)_{min}} \quad (3)$$

$$y_i^*(x) = \frac{y_i^0(x) - y_i^0(x)_{min}}{y_i^0(x)_{max} - y_i^0(x)_{min}} \quad (4)$$

where, $i \in [1,m]$ and $x \in [1,N]$, m is the number of experimental runs and N is the number of response objectives. $y_i^0(x)_{max}$ and $y_i^0(x)_{min}$ are the largest and smallest values of $y_i^0(x)$, normalized data and $y_i^*(x)$ is the original data.

Step2: Compute grey relational coefficient (GRC) using Eq. (5),

$$\varepsilon_i(x) = \frac{\delta_{min} - \varepsilon \times \delta_{max}}{\delta_i^0(x) - \varepsilon \times \delta_{max}} \quad (5)$$

Where $\delta_i^0(x) = y_i^0(x) - y_i^*(x)$, $\delta_i^0(x)$ is the deviation coefficient, $y_i^0(x)$ is the normalized data and $y_i^*(x)$ is the original data.

Step3: Calculate grey relational grade (GRG) using Eq. (6),

$$\gamma_i = \frac{1}{N} \times \sum_{k=1}^{N} \varepsilon_i(x) \quad (6)$$

GRG depicts the overall quality index and the degree of correlation between the normalized data and the original data. The values of GRG determine the ranking of experimental runs and obtain optimal set of variables.

Step4: Calculate the Analysis of Variance (ANOVA) to find out the sensitivity of the variables to the design process at 95% confidence level and obtain the response table. This includes ranks based on delta statistic, which compares the relative magnitude of the effects. The delta statistic shows the difference between the largest and the smallest average for each variable. It finally indicates the most sensitive variables to the design process.
In this paper, this approach is adopted to find the optimal set of design parameters for the proposed NSBUF II algorithm.

2.2. Utilization based production cell design

This article considers an important problem related to designing of production systems, known as utilization-based cell design, which has been practiced since 90s [15]. Machine utilization percent is considered as a production factor in this problem. The Machine-Part Utilization Matrix (MPUM) is defined as $U = [u_{ij}]_{q \times p}$ where $\forall i \in \{1..q\}, \forall j \in \{1..p\}$. The machine utilization percent of part j on machine i is defined as a value in the range $\{0,1\}$, if the part goes to that machine otherwise zero. The cumulative utilization of the part j is less than 1. It needs appropriate clustering technique to group the parts into families and machines into cells, which further converts the MPUM into block-diagonal structure where non-zero elements of the matrix are appeared diagonally in blocks. An operation lying outside the diagonal blocks indicates a bottleneck machine (Exceptional Element), which increases the inter-cell material flow. The objective of this problem is to rearrange the MPUM to control the total utilization percentage induced by bottleneck machines and total number of empty voids inside the cellular structure. An example MPUM and block diagonal solution matrix are presented in Table 1(a) and 1(b) respectively.

Table 1. MPUM of size 5×7 (a) and the block diagonal solution matrix (b)

(a)

	p1	p2	p3	p4	p5	p6	p7
m1	0	0.23	0	0.19	0.33	0.11	0
m2	0.52	0	0.23	0	0	0	0
m3	0.21	0	0.12	0	0	0.26	0.17
m4	0	0.25	0	0.52	0	0.12	0
m5	0.13	0	0	0	0.51	0	0.23

(b)

	p2	p4	p6	p1	p3	p5	p7
m1	0.23	0.19	0.11	0	0	0.33	0
m4	0.25	0.52	0.12	0	0	0	0
m2	0	0	0	0.52	0.23	0	0
m3	0	0	0.26	0.21	0.12	0	0.17
m5	0	0	0	0.13	0	0.51	0.23

The production cell design is classified as a NP-hard problem, which is combinatorial in nature [16]. Therefore, much attention has been offered while developing suitable methodologies to obtain optimal solutions for the stated problem. In recent past, several review articles appeared based on the solution methodologies [17]-[19]. These methodologies can be primarily categorized as mathematical programming-based approaches, bio-inspired techniques such as neural networks and meta-heuristics algorithms [17], [20]. These are exclusively genetic algorithms (GA) [21]-[25], tabu search [26]-[27], simulated annealing [28]-[29], ant colony optimization (ACO)

[30]-[31], particle swarm optimization (PSO) [32]-[33], bee's algorithm [34], water flow-like algorithm [35], firefly-inspired algorithm [36], bacteria foraging algorithms [37], bat algorithms [38]-[39] etc. Few noticeable facts obtained from past literature,

- Researchers preferred to consider the binary data instead of workload data while designing cells [15].
- Optimization of the cell load variations and number of bottleneck machines are used as objectives while designing production cells and weighted sum method is used to solve these objectives, which eventually reduces the problem into a scalarized single-objective problem [20].
- The opted methodologies are not multi-objective in true sense due to the above facts.
- Fourthly, not many performance metrics are available in past literature except the recently published one [15], which can truly evaluate the solutions obtained.

The utilization-based production cell design is opted in this research with two objectives, which are solved using proposed NSBUF II algorithm. The objectives are defined mathematically in Eq. (7) and Eq. (8),

Minimization of total machine utilization percentage induced by exceptional elements (TEU) is expressed as,

$$minimize\ f1 = 0.5 \times \sum_{k=1}^{c}\sum_{j=1}^{p}\sum_{i=1}^{q}\left(x_{ik} - y_{jk}\right)^2 u_{ij} \tag{7}$$

Minimization of total number of voids in cellular blocks is expressed as,

$$minimize\ f2\ = \sum_{k=1}^{c}\sum_{j=1}^{p}\sum_{i=1}^{q}(1 - a_{ij})\, x_{ik}\, y_{jk} \tag{8}$$

$$u_{ij} = \begin{cases} x, & if\ part\ j\ is\ processed\ in\ machine\ i, \\ 0, & Otherwise \end{cases} \quad (x \in \mathbb{R}\,|x > 0) \quad \forall i \in \{1..q\}, \forall j \in \{1..p\} \tag{9}$$

$$a_{ij} = \begin{cases} 1, if\ part\ j\ is\ processed\ in\ machine\ i, \\ 0, \end{cases} \quad Otherwise \quad \forall i \in \{1..q\}, \forall j \in \{1..p\} \tag{10}$$

$$x_{ik} = 1\ if\ machine\ i\ is\ in\ cell\ k, else\ 0 \qquad \forall i, k \tag{11}$$

$$y_{jk} = 1\ if\ machine\ j\ is\ in\ cell\ k, else\ 0 \qquad \forall j, k \tag{12}$$

$$\sum_{k=1}^{c} x_{ik} = 1 \qquad \forall i \tag{13}$$

$$\sum_{i=1}^{q} x_{ik} \geq 1 \qquad \forall k \tag{14}$$

$$\sum_{k=1}^{c} y_{jk} = 1 \qquad \forall j \tag{15}$$

$$\sum_{j=1}^{p} y_{jk} \geq 1 \qquad \forall k \tag{16}$$

Eq. (9) depicts the MPUM matrix, Eq. (10) depicts the machine-part incidence matrix and Eq. (11) - (12) are the decision variables. Eq. (13) - (16) are the assignment constraints, which ensure that each machine/part is assigned to only one cell and each cell holds at least one machine/part.

2.3. Self-Organizing Map (SOM)

The Self-Organizing Map (SOM) is classified as unsupervised artificial neural network (ANN) algorithm, which can present the high-dimensional (n-D) data against a two-dimensional (2D) map [40]. This presentation saves the topological information of the original data in such a manner that the new data with close similarity could further be mapped to relatively nearby locations to the previous data on the 2D map. This graphical presentation enables the users to visualize patterns or clusters (data similarity). Thereafter the correlations among the data points could be revealed. SOM is also known as Kohonen's SOM (KSOM) or self-Organizing feature map

(SOFM) [41]-[42]. SOM has been used in several domains of research ranging from data clustering, modelling of machining process, multi-objective performance prediction due to its strong visualization capabilities [43]-[46]. However, SOM has never been studied in correlational analysis of multi-objective responses and experimental design parameters yet.

Steps of SOM algorithm are,

Competitive Step: In every iteration SOM trains the network using an input data point x. It is a competitive learning process, which implies that a winning neuron c with weight vector w_c is selected on the map, which is closest to the input data point x based on,

$$||x - w_c|| = min_i(||x - w_i||) \tag{17}$$

Cooperative Step: The weight vector w_c is updated to match the data point x. Thereafter the weights of the neurons in close neighborhood of c are also updated, which visually implies that the neurons in proximity moved towards c which represents the data point x. The update expression is demonstrated as,

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)] \tag{18}$$

Where t is the count of iterations, $x(t)$ is the input data point x during t. Here, the $h_{ci}(t)$ is the *neighborhood function* in the vicinity of the winning neuron c. It is decreasing type Gaussian function ranged from c to neighborhood boundary. It is defined as,

$$h_{ci}(t) = \alpha(t) \times e^{\left(-\frac{|r_c - r_i||}{2\sigma(t)^2}\right)} \tag{19}$$

Where $\alpha(t)$ is defined as the learning rate, which states the territory of impact for any input data on the map. The adjustment of the winning neuron is proportional with the vastly spaced neighborhood throughout the entire training phase. SOM algorithm is not truly designed to perform clustering, but it is a visual tool, which reduces the dimensionality of the data to present it in 2-D and reveals hidden pattern in data.

This visualization aspect of SOM is utilized in this research to identify the correlation among the design variables and multi-objective responses of NSBUF II. The complete flow of this research work is portrayed in Figure 2.

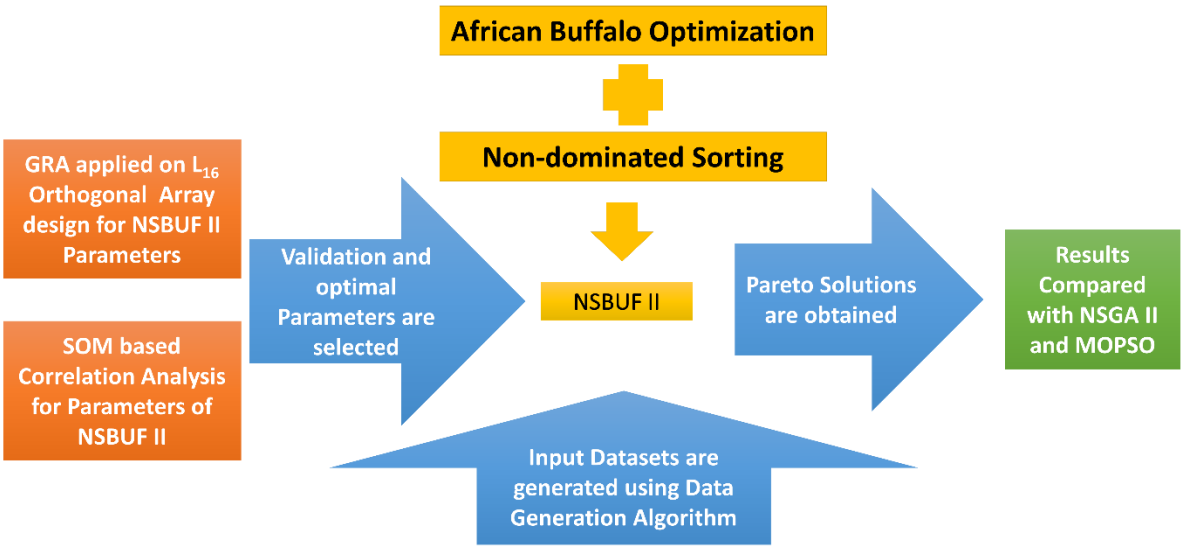

Figure 2. Research flow diagram of NSBUF II

2.4. NSBUF II

The proposed NSUF II starts with a herd of N_p buffalo (solutions). These solutions or simulated 'buffalo' are encoded on an integer vector of pre-defined length depending on the number of machines and parts in the plant. Each solution has a position x_i where i is the row index in the herd matrix. Every row in the matrix is a representation of a buffalo with its position. If the number of machines and parts are m and n respectively, then one buffalo is represented by $m+n$ dimensional vector.

2.4.1 Initial herd generation

Initial herd is generated using a specifically designed random vector generator function, which generates every solution vector with a length of $m+n$ and the elements of the vector are in the interval [1, c] with at least one occurrence [47]. An example solution vector of 5 machines, 7 parts and 2 cells test data has the length of 12 bits (5+7) with each bit representing a cell number (either 1 or 2). Encoding of a solution is presented in Table 2.

Table 2. Example solution vector of NSBUF II for the 5×7 problem

M1	M2	M3	M4	M5	P1	P2	P3	P4	P5	P6	P7
1	2	2	1	2	2	1	2	1	2	1	2

This implies, cell 1 contains machines 1, 4 and parts 2, 4, 6 and cell 2 contains machine 2, 3, 5 and parts 1, 3, 5, 7 respectively. Using this method, the whole initial herd matrix of size $N_p \times (m+n)$ is generated.

2.4.2 Setting up the parameters

The choice of parameters in optimization algorithms has a large impact on the process of optimization. Selection of optimum parameters is a critical task for the researchers, which can attain global optimal solutions [48]. In this study, the GRA coupled with Taguchi's orthogonal design is adopted. The computational analysis is portrayed in section# 3.

2.4.3 Fitness functions

The objective functions principally evaluate the fitness of a solution by computing numerical scores. Eq. (7) and Eq. (8) are used as the fitness functions for NSBUF II. Once the fitness values for the whole herd are handy, position of each buffalo is checked with others in the herd for the non-dominance. The solution is marked to be strictly non-dominated if it is superior for all the objectives considered. Then the non-dominated solution is moved to an empty pool. This procedure is repeated for every buffalo in the herd. At the end, a pool of non-dominated solutions is generated. All these solutions marked as non-dominated, represent the Pareto solutions for one iteration.

2.4.4 Modified fitness and position update strategy for NSBUF II

In this step, fitness and position update strategies for the NSBUF II are defined. NSBUF II requires special strategy to be adopted for discrete combinatorial optimization problems. The current position of a buffalo, x_i is generally updated using Eq. (1) and Eq. (2) for ABO. These are modified and stated in Eq. (20) and Eq. (21). For the first iteration, bg and bp_i are selected randomly from the non-dominated pool obtained in last step. w_i, bg and bp_i can be expressed as assignment matrices y_{wi}, y_{bg}, y_{bpi} of size $(m+n) \times c$ where c is the number of cells and m, n are the number of machines and parts respectively.

$$m_{i+1} = m_i + lp1 \times [y_{bg} - y_{wi}] + lp_2 \times [y_{bpi} - y_{wi}] \tag{20}$$

$$y_{w(i+1)} = \frac{(y_{wi} \times m_{i+1})}{2 \times \lambda} \tag{21}$$

Eq. (21) generates an intermediate assignment matrix $y_{w(i+1)} = [a_{k\times j}]_{(m+n)\times c}\ \{\forall k \in 1 \ldots m+n;\ \forall j \in 1 \ldots c\}$ with real values. In order to obtain the equivalent binary assignment matrix $y'_{w(i+1)} = [a'_{k\times j}]_{(m+n)\times c}\ \{\forall k \in 1 \ldots m+n;\ \forall j \in 1 \ldots c\}$, some assignment rules are applied,

$$a'_{k\times j} = \begin{cases} 1, & a_{k\times j} == \max(a_{k\times j}) \quad and\ a_{i\times j} \neq 0 \quad \forall j \in 1 \ldots c \\ 1, & (a_{k\times j} < 0) \text{ and } \max(a_{k\times j}) == 0 \quad \forall j \in 1 \ldots c \\ 1, & (a_{k\times j} == 0\ and\ \min no.of\ non-zero\ in\ j^{th} cell \quad \forall j \in 1 \ldots c \\ 0, & Orherwise \end{cases} \tag{22}$$

2.4.5 Swap based local search

In order to improve the speed of convergence for NSBUF II, a small local search module is adopted, which might explore the unexplored area of solution search space. This part of the algorithm is an extension, where it is assumed that the leader of the herd can execute a random walk while exploring the green field. This phenomenon doesn't happen always but could happen sometime, which might produce improved results. A probability P_{local} is defined for that purpose. A random number $RN1$ is generated so that, if $RN1 < P_{local}$, a non-dominated solution vector is selected from the pool and two-point random swap operation is performed on the solution vector. This procedure

would try to diversify the population with trivial modifications, which eventually helps in finding optimal solutions. The procedure is depicted in algorithm 2.

Algorithm 2: Swap Based Local Search

Step 1. *While* $RN1 < P_{local}$
Step 2. *Do*
Step 3. *Select* vector *v(1 to m+n)* from non-dominated pool
Step 4. *Define vm = v(1 to m)* and *vn=v(1 to n)*
Step 5. *Generate* random integers, r1 ∈ [1, m], r2 ∈ [1, m], r3 ∈ [1, n] and r4 ∈ [1, n]
Step 6. *tempmc = vm(r1)*
Step 7. *vm(r1) = vm(r2)*
Step 8. *vm(r2) = tempmc*
Step 9. *temppt = vn(r3)*
Step 10. *vn(r3) = vn(r4)*
Step 11. *vn(r4) = tempt*
Step 12. *v*=concatenate (*vm, vn*)
Step 13. *Return* v

2.4.6 Termination condition

The execution of NSBUF II is controlled by some stopping condition. The execution is eventually terminated if the count attains the pre-defined number of iterations. The NSBUF II algorithm is depicted in algorithm 3.

Algorithm 3: NSBUF II

Input: Utilization-based Machine-Part incident matrix
Output: Machine Cells and Part families with low TEU and Void scores
Step 1. *Initialize parameters:*
iterations = 700, herd_size = 200, lp1 = 0.8; lp2 = 0.7, λ = 0.5, iter=0, index=0, m, n, c, P_{local}
Step 2. *While* (index < herd_size)
Step 3. *Generate* v = herd (index,1 to m+n)
Step 4. *Set* herd (index, m+n+1) = f1(v); herd (index, m+n+2) = f2(v)
Step 5. *Set* index = index+1
Step 6. End
Step 7. *Create* empty list Non_dom []
Step 8. *Set* p=0, q=0
Step 9. *While* (p, q < herd_size)
Step 10. *If* (herd (p, m+n+1) < herd (q, m+n+1) && herd (p, m+n+2) < herd (q, m+n+2))
Step 11. *Set* Non-dom (index, 1 to m+n) = herd (p, 1 to m+n)
Step 12. *Set* Non-dom (index, m+n+1) = f1(v), Non-dom (index, m+n+2) = f2(v)
Step 13. *Set* p=p+1; q=q+1
Step 14. End
Step 15. *While* (iter < iterations)
Step 16. *While* (index < herd_size)
Step 17. Obtain new solution using Eq. (20)-(22)
Step 18. Create an empty pool [] and add this new solution in pool
Step 19. *Set* p=0, q=0
Step 20. *While* (p, q < herd_size)
Step 21. *If* (herd (p, m+n+1) < herd (q, m+n+1) && herd (p, m+n+2) < herd (q, m+n+2))
Step 22. *Set* Non-dom (index, 1 to m+n) = herd (p, 1 to m+n)
Step 23. *Set* Non-dom (index, m+n+1) = f1(v), Non-dom (index, m+n+2) = f2(v)
Step 24. *Set* p=p+1; q=q+1
Step 25. *Set* index = index+1
Step 26. End
Step 27. End
Step 24. Perform algorithm 2 if random number <P_{local} probability and obtain new solution
Step 25. *Execute* Step 16-22
Step 26. *Set* iter = iter+1
Step 27. end
Step 28. Terminate with optimal solution set

2.4.7. Computational Complexity of NSBUF II

Let P number of objective functions are to be considered, and the herd size is n for NSBUF II, the non-dominated sorting plays an important role while evaluating the computational complexity. For the non-dominance of a herd member within n number of herd members, $P \times n$ number of comparisons are computed. From the above theory the worst-case scenario is determined as $O(Pn^2)$. For further information, the complexities of NSGA-II, and MOPSO are also $O(Pn^2)$. Therefore, NSBUF II is comparable with these two popular algorithms.

3. Results and Discussions

The performance of the proposed NSBUF II is demonstrated on bi-objective optimization problems, which obtains optimal production cells. For that matter, utilization-based data are required. These data are not readily available in literature due to the novelty of the problem considered. Realistic test data are generated using a systematic technique demonstrated in algorithm 4.

Algorithm 4: Data Generation

Step 1. Generate random ratio matrix of size (q×p),
Step 2. if ($0 < q \leq 10$)
Step 3. Restrict density of zeroes in the range of 40-50% in generated matrix
Step 4. else if ($10 < q \leq 20$)
Step 5. Restrict density of zeroes in the range of 60-70% in generated matrix
Step 6. else
Step 7. Restrict density of zeroes in the range of 80-90% in generated matrix
Step 8. end
Step 9. end
Step 10. end
Step 11. Restrict each row sum ≤ 1

30 test data of small to medium sizes ranging from 4×7 to 35×48 are obtained using the above algorithm. At first, the sensitivity of various parameters for NSBUF II is investigated using the Taguchi's orthogonal design coupled GRA. Determining the optimal set of parameters is crucial in this regard.

3.1. Parameter selection for NSBUF II

The viable values for the parameters of NSBUF II was suggested in ref. [4] (i.e., lp1 = 0.6, lp2 = 0.4, herd_size = 40, iterations = 50), however it is also stated that the values of parameters are dependent on the problems considered. To conduct the Taguchi's orthogonal design, four levels of the parameters are selected for L_{16} design (Table 3). This recommends 16 experimental runs, which are presented in Table 4. 18×35 test data is taken to obtain the responses of NSBUF II. To obtain responses, NSBUF II is executed for 20 times for each experimental set and average values of responses are recorded. Using Eq. (3)-(6) the GRA is performed on both the original responses. The GRC and GRG values are obtained for the trial runs of NSBUF II. The results are depicted in Table 4. The GRG response table (Table 5) depicts the mean of each response characteristic for each level of the parameters. The table depicts delta statistic while comparing the relative importance of outcomes. It portrays the difference between the largest and the smallest mean of the parameters. Ranks are allotted based on obtained Delta values. Using the level mean in the response table optimal set of levels of the parameters could be selected for optimal performance of NSBUF II. According to Table 5 number of iterations has the greatest importance, lp1 is the next most significant parameter, followed by lp2 and herd_size. The main effects plot of Figure 3 shows that the optimal set of parameters are lp1 = 0.8, lp2 = 0.5, herd_size = 200, iterations =700 (λ is prefixed to 0.5 as stated) respectively.

Table 3. Parameters of NSBUF II and their levels

Levels	lp1	lp2	herd_size	Iterations
1	0.2	0.5	200	900
2	0.4	0.7	400	700
3	0.6	0.2	600	500
4	0.8	0.9	800	300

Table 4. Taguchi's Design coupled GRA for the NSBUF II parameters

Taguchi's L16 Design					Original Responses		Grey Relational Analysis						
Trials	lp1	lp2	herd _size	Iterations	Voids	TEU	Normalized responses		Deviation Sequence		Grey Relational Coefficient		GRG
							Voids	TEU	Voids	TEU	Voids	TEU	
1	0.2	0.5	200	900	102.000	10.448	0.602	0.736	0.398	0.264	0.557	0.654	0.606
2	0.2	0.7	400	700	145.000	9.296	0.084	1.000	0.916	0.000	0.353	1.000	0.677

3	0.2	0.2	600	500	79.000	11.374	0.880	0.523	0.120	0.477	0.806	0.512	0.659
4	0.2	0.9	800	300	98.000	10.192	0.651	0.794	0.349	0.206	0.589	0.709	0.649
5	0.4	0.5	400	500	94.000	10.326	0.699	0.764	0.301	0.236	0.624	0.679	0.652
6	0.4	0.7	200	300	89.000	10.463	0.759	0.732	0.241	0.268	0.675	0.651	0.663
7	0.4	0.2	800	900	95.000	10.686	0.687	0.681	0.313	0.319	0.615	0.611	0.613
8	0.4	0.9	600	700	102.000	9.994	0.602	0.840	0.398	0.160	0.557	0.757	0.657
9	0.6	0.5	600	300	95.000	10.209	0.687	0.790	0.313	0.210	0.615	0.705	0.660
10	0.6	0.7	800	500	152.000	9.919	0.000	0.857	1.000	0.143	0.333	0.778	0.555
11	0.6	0.2	200	700	86.000	10.170	0.795	0.800	0.205	0.200	0.709	0.714	0.712
12	0.6	0.9	400	900	69.000	13.655	1.000	0.000	0.000	1.000	1.000	0.333	0.667
13	0.8	0.5	800	700	90.000	9.740	0.747	0.898	0.253	0.102	0.664	0.831	0.747
14	0.8	0.7	600	900	103.000	9.920	0.590	0.857	0.410	0.143	0.550	0.777	0.663
15	0.8	0.2	400	300	88.000	10.641	0.771	0.691	0.229	0.309	0.686	0.618	0.652
16	0.8	0.9	200	500	91.000	10.167	0.735	0.800	0.265	0.200	0.654	0.714	0.684

Table 5. Response Table for Means of GRG

Level	lp1	lp2	Herd Size	Iterations
1	0.6474	0.6589	0.6661	0.6559
2	0.6461	0.6661	0.6617	0.6375
3	0.6484	0.6396	0.6598	0.6982
4	0.6867	0.6641	0.641	0.6371
Delta	0.0406	0.0264	0.025	0.0611
Rank	2	3	4	1

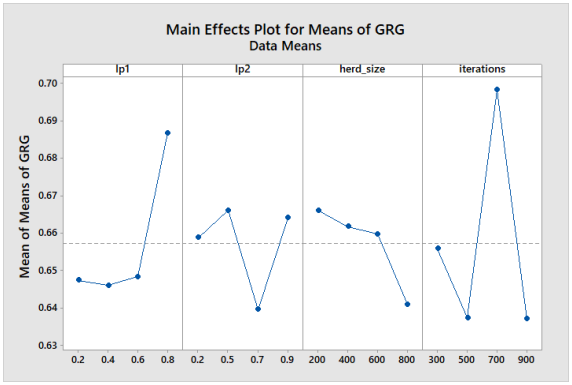

Figure 3. GRG Main effect graph for the parameters of NSBUF II

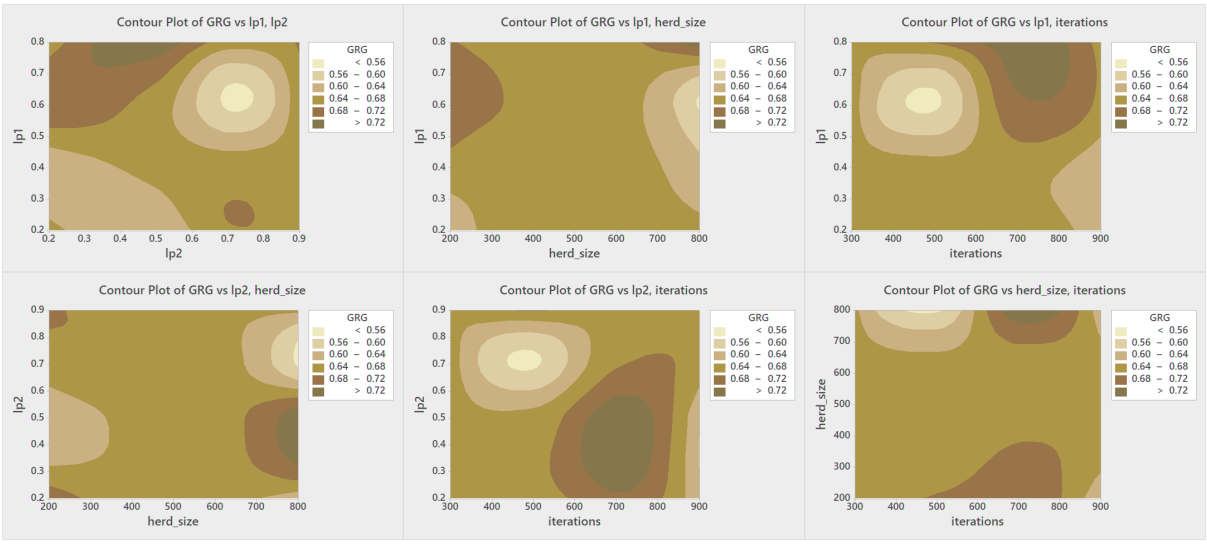

Figure 4. Response surfaces of GRG vs. combinations of each two parameters of NSBUF II

Figure 4 shows the response surfaces for each two of the parameters of NSBUF II vs the obtained GRG values. For the lp1-lp2 combination, a better GRG score could be obtained when the lp1 varies around 0.7-0.8 and lp2 lies in the range of 0.3-0.5. For the lp1-iteration combination, higher GRG values are attained within the range of lp1 values (0.6-0.8) and the iteration values (700-800). When the lp2 varies in the range of 0.2-0.6 and the iteration

remains in the 600-800 range, the GRG score improves. It could be seen that herd_size is less sensitive to the NSBUF II, which also validates the ranks portrayed in Table 5.

3.2. Correlational Analysis by SOM

In this research, SOM is utilized as a correlational tool with an aim of detecting the underlying dependencies among the design parameters of NSBUF II and the responses or objectives considered. Voids and TEU are the objectives of the problem being solved. There are two goals behind the SOM analysis, first, to validate the GRA findings and second, to visually present the objective spaces and correlate with the parameters of NSBUF II. SOM could be more useful for 4 or more dimensional problems where visual presentation of the Pareto front is a challenge. This SOM analysis is done using matlab SOM toolbox [49]. The training of the SOM network is a crucial step and that is done with data presented in Table 4. In past literature, the performance of SOM is measured using, quantization error (QE), topographic error (TE). Both QE and TE could evaluate the performance of the SOM using measurement of characteristic of the continuousness in mapping and topological preservation. These metrics efficiently measure the preservation of topology in grid map of SOM. Minimum or zero values of QE and TE are expected [46]. Therefore, the SOM network is trained for several times to find out better QE, TE and combined error (CE) values with different map sizes. The results are shown in Table 6. For the present training data, an 18×18 map is opted which produces zero QE, TE and CE.

Table 6. Optimal map size selection for SOM

Display MAP Size	MAP Units	QE	TE	CE
7×5	6×6	1.138	0	1.775
12×7	9×9	0.387	0	0.827
14×10	12×12	0.072	0.188	0.240
19×12	15×15	0.003	0	0.001
22×15	18×18	0	0	0

The SOM visual results are depicted in Figure 5, which shows seven sub-maps. The first one (pink) is the Unified distance matrix (U-matrix), which helps in visualizing possible similarities in the data. The next four (orange) sub-maps represent four design parameters of NSBUF II and rest (cyan) are for two response variables.

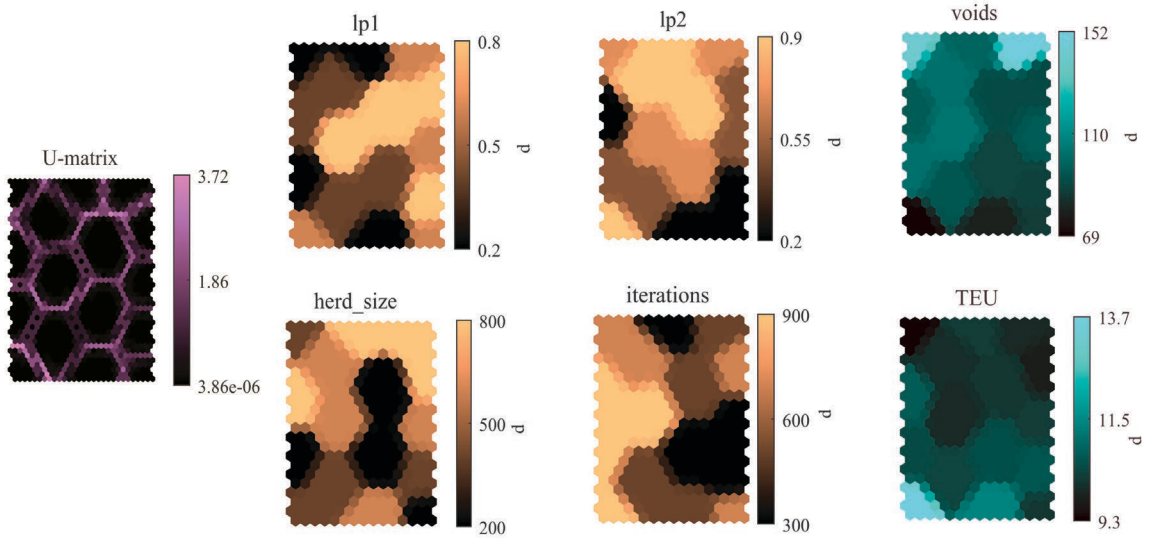

Figure 5. SOM Visualization for the design parameters and responses of NSBUF II

These sub-maps could be interpreted by comparing the patterns and the distributions of colour units. The trends for the responses could be predicted from these sub-maps with the changes in design variables of NSBUF II. Further the most sensitive variables could be pointed out which affect the responses. Following are the findings,

- The colour codes vary from lighter (higher values) to darker (lower values) shades and it can be seen that lp2 and herd_size follow the similar pattern along with the voids, hence these are proportionally sensitive to voids. Voids are minimum (dark shade) with the lower values of lp2 and herd_size.
- For 2nd response TEU, lp1 and iterations are most important since these parameters preserve lower values when TEU is minimum (dark shed).
- It is also noticed that, for combined objectives, iterations and lp1 both are significant factors whereas and herd_size have less correlation when both the responses are considered, which justifies the results obtained using GRA.

- While comparing the responses with each other's, they depict the opposite behavior. Voids are minimum when TEU is maximum, which follows an inverse correlation. That proves the true nature of multi-objective problems and the conflicting behavior of objectives while searching for Pareto optimality.
- In the U-matrix sub-map, it could be observed that very few areas with light sheds and the dark shaded areas are uniformly spread throughout the sub-map. Thus, another relevant finding is that, the Pareto optimal values are uniformly distributed throughout the state space with no peak or valley.

From the above findings, it can be concluded that the SOM has some excellent abilities that correlates the process design variables of NSBUF II and the output responses, validates the nature of multi-objective optimization, confirms the data uniformity, and facilitates similar findings like GRA without much statistical analysis.

3.3. Computational Results

To validate the proposed NSBUF II, 30 test data are used as stated earlier. The NSBUF II algorithm is coded with MATLAB libraries on Intel 8650U @1.90 GHz computer. The results are compared with the results obtained using two popular multi-objective algorithms namely MOPSO [9] and NSGA II [8]. The MOPSO and NSGA II flowcharts are depicted in Figure 6.

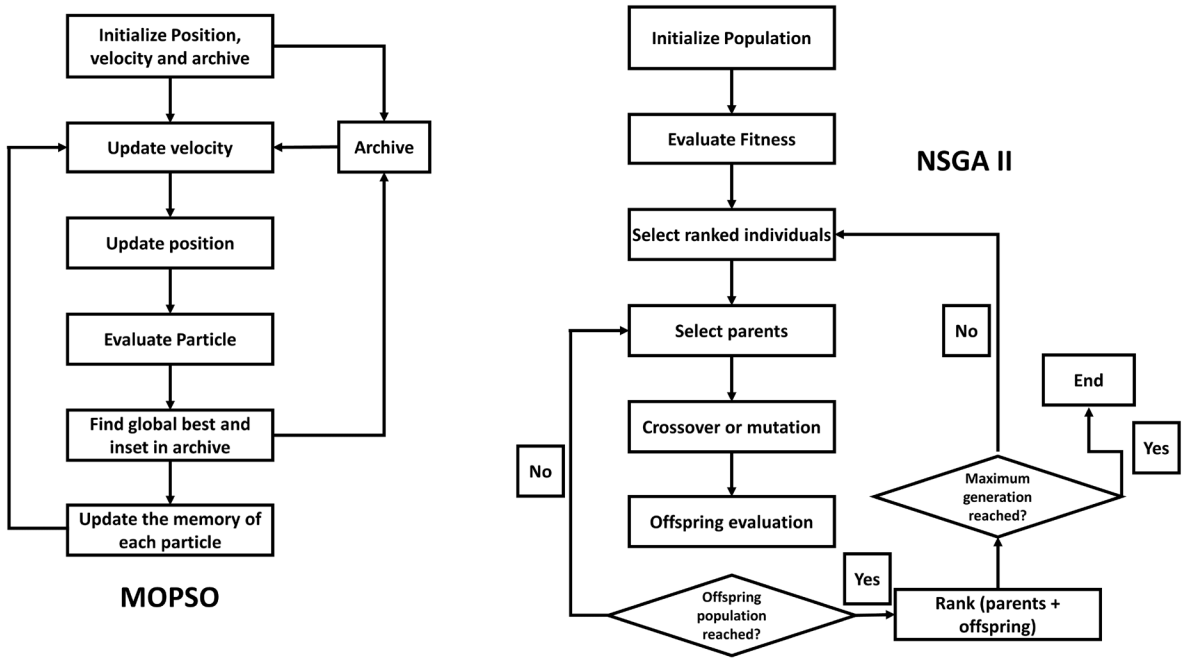

Figure 6. MOPSO and NSGA II flowcharts

The NSBUF II is shown to attain promising solutions with optimal objective scores and outperforms the published MOPSO and NSGA II. Due to the nature of NP-Hardness of the problem, obtaining solutions is not an easy task. The number of variables and constraints increases with the size of the data. Hence the algorithmic complexity increases exponentially with the increased number of machines and parts. Therefore, a good design is indeed important while dealing with larger data. The evaluation criteria of NSBUF II is based on TEU, and the total number of voids. For all the three algorithms, the presented solutions are picked from obtained Pareto frontiers. Table 7 presents the comparison among NSBUF II, NSGA II, and MOPSO algorithms and reveals that the NSBUF II algorithm is an improved method, which not only reduces the exceptional elements and voids but also minimizes the TEU. Computational time is not considered in this research since it is not a focus area here. The optimal solution for the largest dataset (35×48) is obtained within seven minutes. Smaller datasets consumed trivial CPU time. A powerful Pareto frontier could be observed for NSBUF II algorithm while compared with MOPSO and NSGA II in Figure 7. The NSBUF II Pareto frontier consists of 17 near optimal solutions while comparing with the 4 Pareto optimal solutions are attained by MOPSO and 14 Pareto solutions are obtained by NSGA II.

Table 7. Performance comparison: NSBUF II vs. MOPSO and NSGA II

			NSBUF II			MOPSO			NSGA II		
No.	Size	No. of Cells	TEU	Voids	D Values	TEU	Voids	D Values	TEU	Voids	D Values
1	4×7	2	0.509	3	0.453383	0.676	4	0.308249	0.509	3	0.41576
2	5×10	2	0.564	14	0.508007	0.564	14	0.270494	0.564	14	0.466619
3	6×8	2	0.750	15	0.672766	0.654	17	0.315196	0.652	16	0.539261

4	7×10	2	0.661	17	0.595888	0.661	17	0.318198	0.661	17	0.547409
5	7×11	2	0.999	12	0.891731	0.846	21	0.40574	0.862	16	0.708943
6	8×15	2	0.501	39	0.495668	1.762	23	0.813335	1.03	33	0.859725
7	8×22	2	0.653	55	0.656471	0.653	55	0.450628	0.653	55	0.611605
8	9×9	2	0.402	29	0.392335	1.137	12	0.522067	0.721	22	0.60058
9	9×15	2	0.869	31	0.792412	1.428	23	0.664374	1.69	10	1.380429
10	10×10	3	0.744	22	0.673411	1.115	19	0.52016	0.914	21	0.754736
11	10×10	3	1.111	23	0.997135	1.930	18	0.884231	1.111	23	0.915418
12	10×25	3	1.768	50	1.598013	1.768	50	0.860698	1.768	50	1.468561
13	10×25	3	1.023	93	1.046603	1.716	62	0.868593	1.023	93	0.976982
14	12×12	3	1.098	55	1.023884	2.747	23	1.256615	1.39	42	1.157396
15	12×24	3	1.570	128	1.567562	3.189	49	1.480635	1.462	167	1.501227
16	12×29	3	1.438	125	1.455778	2.683	93	1.347554	1.438	125	1.357355
17	14×30	3	1.519	183	1.690809	2.950	116	1.519646	1.519	183	1.591696
18	14×35	3	1.314	234	1.747331	1.314	234	1.561351	1.314	234	1.667151
19	15×15	3	1.314	81	1.252981	2.427	49	1.143754	2.311	57	1.911655
20	16×32	3	1.859	246	2.144903	5.028	87	2.347464	3.918	171	3.331068
21	17×27	3	1.706	229	1.979879	1.753	239	1.675073	1.706	229	1.870598
22	18×35	3	1.753	324	2.380232	6.422	122	3.014324	3.013	209	2.710565
23	18×35	3	1.808	322	2.404355	1.649	352	2.295864	2.19	291	2.390513
24	18×35	3	1.750	335	2.424926	1.750	335	2.21315	1.750	335	2.319321
25	20×20	4	2.978	166	2.806219	3.860	132	1.93402	2.978	166	2.59378
26	20×35	4	4.258	270	4.075417	7.007	146	3.309626	4.258	270	3.774501
27	22×35	4	1.605	291	2.155724	1.428	468	2.957318	1.576	343	2.270488
28	24×40	4	2.497	289	2.740546	2.983	593	3.899162	2.752	484	3.466294
29	30×48	5	5.319	481	5.434774	8.638	381	4.575184	5.319	481	5.072539
30	35×48	5	2.878	835	5.293391	7.500	437	4.345018	3.33	801	5.145595

Since the Pareto front for every dataset contains many good solutions, it is difficult to pick the most promising one. For that matter the *knee point* concept is utilized in this work [50]. A *Knee point* F^k is defined as the solution point on the Pareto front having the shortest euclidean distance from the *utopia point* F^u. F^u is defined as a solution point $\mu^* \in \Omega$ such that $f_k(\mu^*) \geq f_k(\mu)$ for $\forall \mu \in \Omega$ and $k \in \{1, 2, ..., P\}$ where P is the number of objectives. The concept of *knee point* is depicted in Figure 8. The knee point solutions are picked for every data set and demonstrated as the best solutions. The obtained solutions are depicted in Table 7 for NSBUF II, NSGA II, and MOPSO algorithms. The euclidean distances D (Figure 8) are also computed and displayed in Table 7. These D values are used further to evaluate the performances of the algorithms statistically. For that matter (0,0) is selected as the utopia point, which is an ideal solution point where the ideal cells are formed with no bottleneck machines and compact cellular structures (without empty places/voids inside cells). The statistical analysis is presented next.

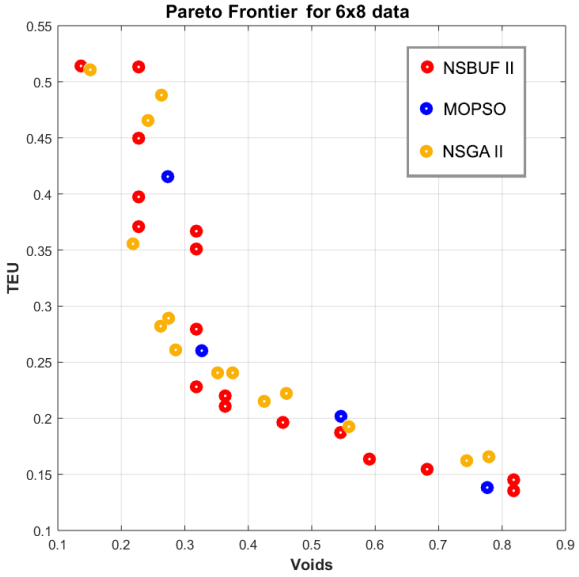

Figure 7. Pareto frontier obtained for example problem of size 6×8

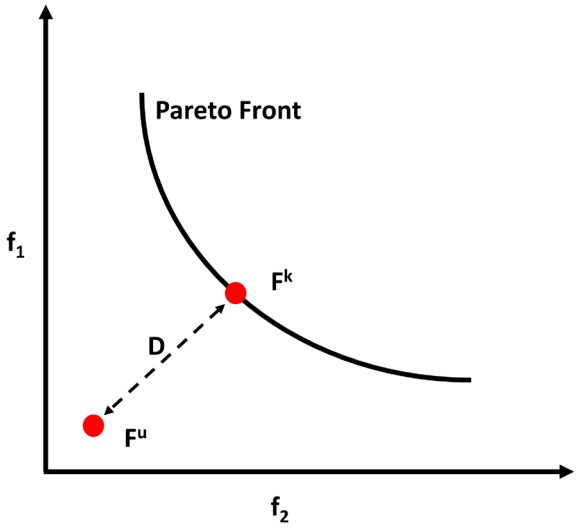

Figure 8. Knee Point concept for Pareto front along with utopia point

In first step, the normality test is carried out using Kolmogorov-Smirnov normality test. The plots are depicted in Figure 9. The null hypothesis H0 is accepted if the data are normal. Therefore, the rejection of the null hypothesis happens when the p-value ≤ α, which is the significance level of 0.05. The obtained p-values are 0.058, 0.054, 0.15 > 0.05 for the NSBUF II, MOPSO, and NSGA II respectively. Therefore, the null hypothesis is accepted, which concludes with 95% confidence level that the obtained results are normally distributed. In next step the equality of variances is tested among the *D* values obtained from Table 7 assuming that the data are normal. If the test statistic < critical value ($F < F_{critical}$) accept the null hypothesis; in other words, if the p-value > α, accept the null hypothesis.

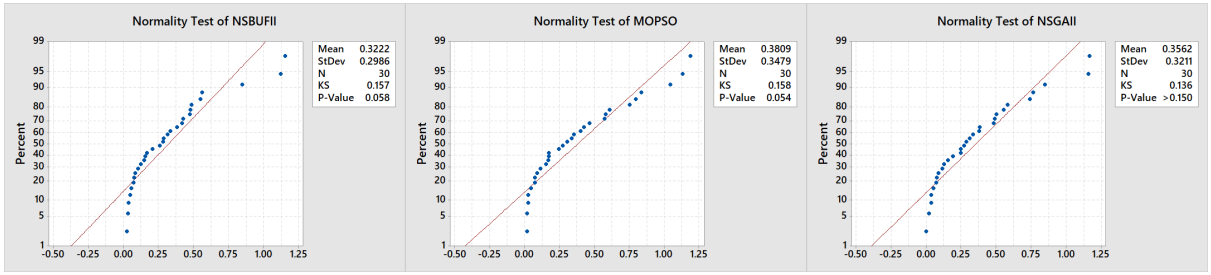

Figure 9. Normality Test for all the three metaheuristics

Table 8. 95% Bonferroni Confidence Intervals for Standard Deviations

Sample	Size	Standard Deviations	Confidence Interval
NSBUFII	30	0.298564	(0.182787, 0.529964)
MOPSO	30	0.347857	(0.247409, 0.531502)
NSGAII	30	0.321129	(0.215281, 0.520561)

Individual confidence level = 98.3333%

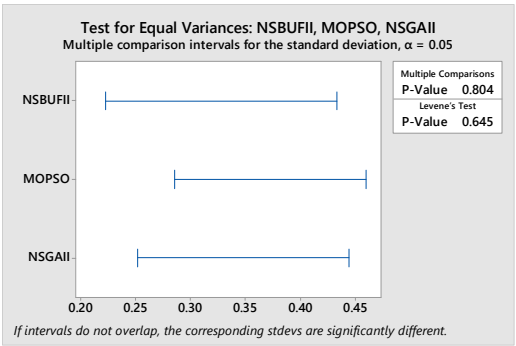

Figure 10. Equality of Variance Test Result

Table 8 and Figure 10 depict the equality of variances test result, where p-values are larger than α and on the plot with the multiple comparison intervals, all the comparison intervals overlap. Therefore, the null hypothesis is accepted, and the variances are equal. In other words, the performance differences of the algorithms are not statistically significant.

In final step, the one-way ANOVA is performed assuming the equal variances. The null hypothesis is accepted if the p-value > α. In this test the comparisons among NSBUF II-NSGA II and NSBUF II-MOPSO are tested. For that matter, the Dunnett's Test is employed [51]. The result is portrayed in Table 9-11. The interval plot is presented in Figure 11. From the p-values, interval plots, and grouping results it can be concluded that the null hypothesis is accepted, and the means are same. Therefore, the obtained results are consistent. Thus, the NSBUF II performs equally good or better than the NSGA II and MOPSO. Therefore, the statistical analysis proves that the proposed NSBUF II is a well performing optimization algorithm.

Table 9. One-way ANOVA result

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	2	0.05210	0.02605	0.25	0.780
Error	87	9.08480	0.10442		
Total	89	9.13691			

Table 10. Dunnett Simultaneous Tests for Level Mean - Control Mean

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
MOPSO - NSBUFII	0.0587	0.0834	(-0.1289, 0.2463)	0.70	**0.705**
NSGAII - NSBUFII	0.0340	0.0834	(-0.1537, 0.2216)	0.41	**0.887**

Individual confidence level = 97.29%

Table 11. Grouping Information Using the Dunnett Method and 95% Confidence

Factor	N	Mean	Grouping
NSBUFII (control)	30	0.3222	**A**
MOPSO	30	0.3809	**A**
NSGAII	30	0.3562	**A**

Means not labeled with the letter A are significantly different from the control level mean.

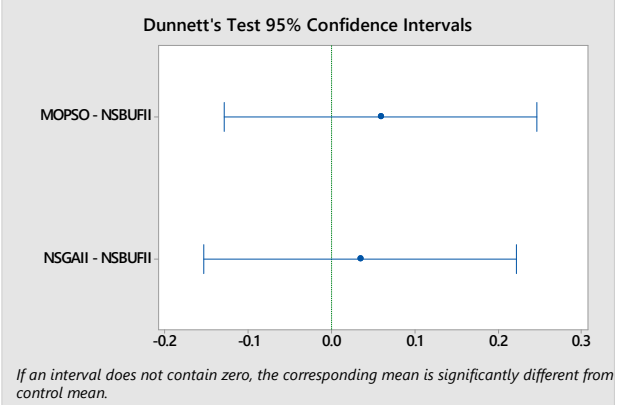

Figure 11. Interval plots of NSBUF II-MOPSO and NSBUF II-NSGA II

3.4. Multi-response AWJM process optimization using NSBUF II

A real-life case is presented based on the Abrasive Water Jet Machining (AWJM), which is identified as a superior method for the glass cutting. The AWJM could be utilized in various forms of glass works such as artistic table-top inlays, detailed stained glass designs, mirrors, glass ornaments, and replacement windows for antique cars etc. [52]. Process parameters of the AWJM process, which primarily affect the quality of cutting, are, water pressure (WP), abrasive flow rate (AFR), traverse speed (TS) and stand of distance (SOD). Important quality characteristics in AWJM are Material Removal Rate (MRR), Surface Roughness (Ra), Top kerf width (TKW), and Bottom Kerf Width (BKW) [53]. AWJM combines the principles of abrasive jet and water jet machining. The AWJM is a non-conventional machining process, where the material is removed by impact erosion of high pressure and high velocity of water and entrained high velocity of grit abrasives on a work piece. The AWJM process (Figure 10) is based on the principle of rapid erosion by high-speed abrasive waterjet combined with rapid cooling by the water

jet and it is a powerful tool for processing of various materials. After vigorous testing within various options, the garnets are selected, which is a substance commonly used on the sandpaper [54]. This technology can achieve faster machining speeds and leave a fine surface quality, which is free of thermal distortion [55]. The ultra-high pressure (UHP) coupled AWJM process could cut any hard material such as ceramics, glass and composites. The commercial application of AWJM is found in sizing the concrete slabs, designing on floor tiles and stones [56].

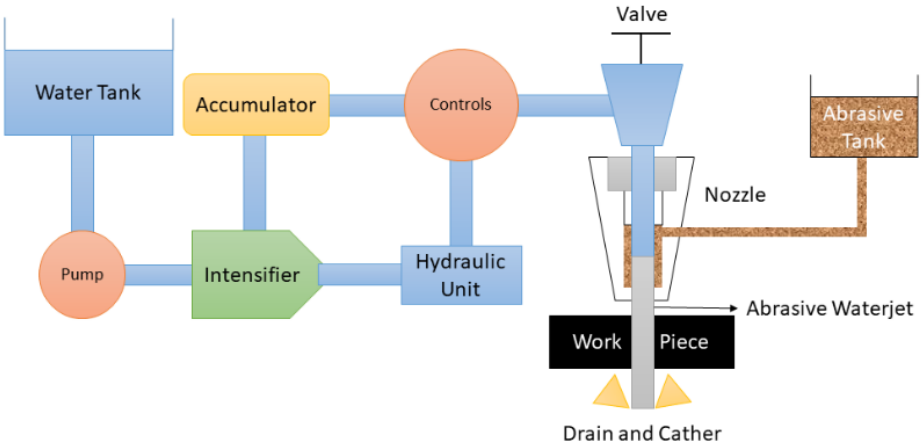

Figure 12. AWJM Process Flowchart

In this research, the glass (soda lime or soda-lime-silica) samples are used and these are installed in wooden blocks for the AWJM cutting. A 60 HP pump was used to generate the required water pressure. The machining process was numerically controlled by 802D SL Sinumeric, 80 BS garnets sand were used as abrasive. In these experiments, WP, AFR, TS and SOD were varied between 1500 to 3500 bar, 2 gm/min to 8 gm/min, 2mm/min to 10 mm/min and 2 mm to 10 mm respectively. The nozzle with tungsten carbide focusing tube of internal diameter of 0.76 mm was used for continuous operations (each) to minimize the effect of incremental nozzle diameter by SiC particle abrasives. Size of the cutting table is 1 m^2, the abrasive feeder used is Abraline III and RO Unit is Tharmax 500LPH. The sequence of machining operation was programmed using Meta CAM3D. The Surface roughness (Ra) was measured by a non-contact profiler (Contour GT-I). The experimental data are presented in Table 12-13 using Box-Behnken Design.

Table 12. AWJM parameters with levels

Factor	Name	Low Level (-1)	Medium Level (0)	High Level (+1)
A	WP (Bar)	1500	3000	4500
B	AFR (Gm/Min)	2	5	8
C	TS (Mm/Min)	2	6	10
D	SOD (Mm)	2	6	10

In order to optimize the AWJM process using NSBUF II, the multiple regression models are obtained for the process responses, MRR, Ra, TKW, and BKW. The details of the analysis are presented in Table 14. From Table 14 it could be seen that the p-values are not in the acceptable range. The R^2 values do not point out good regression fit. Therefore, the optimal results cannot be predicted using this regression model only. To obtain the optimal results, the NSBUF II is employed for the AWJM process optimization. The regression equations are presented in Eq. (23)-(26). These equations are utilized as objective functions to the NSBUF II algorithm.

$$MRR = 0.6608 + 0.000007 \times WP + 0.00078 \times AFR - 0.01204 \times TS + 0.00477 \times SOD \quad (23)$$

$$Ra = 0.0841 - 0.000010 \times WP + 0.01203 \times AFR + 0.00660 \times TS - 0.00513 \times SOD \quad (24)$$

$$TKW = 0.9122 - 0.000011 \times WP + 0.00147 \times AFR - 0.00027 \times TS + 0.00788 \times SOD \quad (25)$$

$$BKW = 0.8566 + 0.000008 \times WP - 0.00025 \times AFR + 0.00004 \times TS - 0.00325 \times SOD \quad (26)$$

The MRR is a maximization type objective and rest are minimization type. The NSBUF II parameters are kept constant as lp1 = 0.8, lp2 = 0.5, herd_size = 200, iterations =700 (λ is prefixed to 0.5) respectively. Due to the nature of the multi-objective optimization, the convergence curve could not be achieved, and Pareto solutions are obtained, which depict multiple optimal solutions with trade-offs among the objectives. The Pareto solutions are portrayed in Figure 13.

Table 13. Experimental design space of AWJM using Box-Behnken designs

	WP	AFR	TS	SOD	MRR	Ra	TKW	BKW
1	4500	5	10	6	0.606	0.034	0.906	0.881
2	3000	2	6	2	0.644	0.069	0.903	0.830
3	3000	8	6	10	0.667	0.052	0.953	0.782
4	1500	2	6	6	0.551	0.085	0.864	0.803
5	3000	2	10	6	0.528	0.042	1.037	0.897
6	3000	2	2	6	0.753	0.024	0.947	0.863
7	1500	5	2	6	0.773	0.058	0.908	0.834
8	4500	8	6	6	0.629	0.130	0.939	0.867
9	3000	5	10	2	0.867	0.280	0.888	0.872
10	1500	8	6	6	0.719	0.334	1.002	0.816
11	3000	5	6	6	0.521	0.067	0.891	0.894
12	4500	5	2	6	0.759	0.004	0.980	0.886
13	4500	5	6	2	0.573	0.380	0.874	0.830
14	1500	5	6	2	0.578	0.098	0.961	0.832
15	3000	8	6	2	0.596	0.031	0.946	0.840
16	3000	5	6	6	0.521	0.067	0.891	0.894
17	3000	8	10	6	0.605	0.211	0.881	0.887
18	3000	5	6	6	0.521	0.067	0.891	0.894
19	3000	8	2	6	0.660	0.138	0.949	0.938
20	1500	5	6	10	0.689	0.286	1.063	0.876
21	3000	5	2	2	0.657	0.220	0.920	0.875
22	4500	5	6	10	0.733	0.063	0.956	0.766
23	3000	2	6	10	0.609	0.087	0.992	0.836
24	1500	5	10	6	0.624	0.090	0.929	0.844
25	3000	5	2	10	0.826	0.120	0.928	0.823
26	3000	5	6	6	0.521	0.067	0.891	0.894
27	3000	5	6	6	0.521	0.067	0.891	0.894
28	4500	2	6	6	0.763	0.156	0.874	0.910
29	3000	5	10	10	0.620	0.224	0.978	0.840

Table 14. Analysis of Variance (ANOVA) results show the p-values and R^2 values of the model

	Source	MRR	Ra	TKW	BKW
P-Values	WP	0.711	0.601	0.244	0.355
	AFR	0.936	0.225	0.752	0.950
	TS	0.106	0.371	0.938	0.989
	SOD	0.512	0.486	0.032	0.287
R^2 Values		12.46%	11.66%	21.94%	7.97%

Total 98 solutions are obtained, and the statistical details of the solutions are portrayed in Table 15. The most promising solution is picked for the validation test. This solution is marked with bold dotted line in Figure 13. Ten experiments were conducted with the obtained parameter settings (Table 16). The average MRR, Ra, TKW, and BKW values are computed and depicted in Table 16. The validation runs are compared with the NSBUF II output. Experimental results are very close to the model outputs with highest error of 12.61%, which is acceptable according to the machine operator. Therefore, the validation test indicates that the NSBUF II result could produce best process responses for the AWJM cutting.

Table 15. Statistical details of the obtained Pareto solutions

	MRR	Ra	TKW	BKW
Min	0.582	0.376	0.891	0.84
Max	0.71	0.187	0.969	0.882
Mean	0.657	0.107	0.932	0.859
Standard Deviation	0.032	0.028	0.0205	0.01

Table 16. Validation test results

#	AWJM Parameters	Predicted Responses	Experimental Responses	Deviations
1	WP=3759.8, AFR=2.1, TS=2.2, SOD=9.5	MRR=0.7073, Ra=0.0376, TKW=0.9483, BKW=0.8553	MRR=0.765, Ra=0.0291, TKW=0.984, BKW=0.895	MRR=8.66%, Ra=12.61%, TKW=11.8%, BKW=6.36%

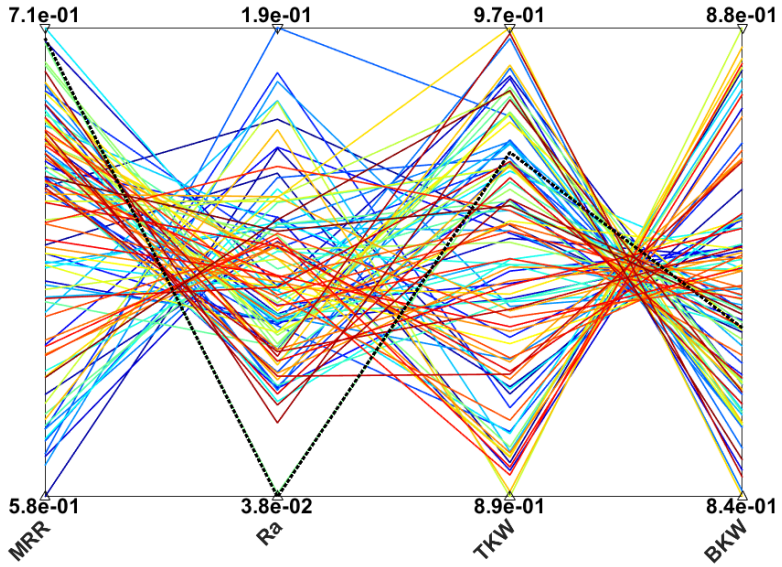

Figure 13. Pareto solutions for AWJM process using coordinate plot

4. Conclusions

This article proposes a multi-objective variant of the African Buffalo Optimization (ABO) algorithm, namely NSBUF II, which incorporates Pareto search for non-dominant solutions in the state space and a local search module for faster convergence. The selection of parameters for the ABO is done using Grey Relational Analysis (GRA), which efficiently obtains the optimal set of the parameters for the NSBUF II. This is further validated using a Self-Organizing Map (SOM) based approach, which is a powerful tool for visualization of the high-dimensional data in 2D plane. Applied SOM also visually reveals the complex correlational structure among the design variables and responses. The performance of the proposed NSBUF II is verified on the utilization based bi-objective optimization problems, which can obtain production cells successfully. The proposed NSBUF II is successfully compared with two state-of-the-art algorithms namely NSGA II, and MOPSO and shown to obtain impressive results. The contributions of this research are as follows,

- The non-dominated sorting module of NSBUF II effectively converts the ABO into a true multi-objective optimization algorithm.
- The GRA effectively obtains optimal set of parameters for the NSBUF II and SOM correctly shows the correlations among the parameters of NSBUF II and the objectives of the utilization-based cell formation problem.
- The NSBUF II is shown to perform well along with the state-of-the-art techniques namely NSGA II and MOPSO. The performance of NSBUF II is confirmed using some statistical approach called Dunnett's Test.
- The performance of the NSBUF II is further verified with the real world AWJM process optimization problem. It is shown to attain the optimal set of process variables and performance characteristics, which are validated using laboratory experiments.

The future extension of this work is to utilize the NSBUF II for the manufacturing process optimization to control the process variabilities in robust design by incorporating more conflicting objectives. Implementation of the many-objective version of NSBUF II considering larger industrial data is another future direction to be explored.

Acknowledgement

This work is supported by the SFI Manufacturing (Project No. 237900) and funded by the Norwegian Research Council.

References

1. Mirjalili, S., Jangir, P., & Saremi, S. (2016). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Applied Intelligence, 46(1), 79-95. doi:10.1007/s10489-016-0825-8
2. Kar, A.K. (2016). Bio inspired computing – A review of algorithms and scope of applications. Expert Systems with Applications, 59(C), 20-32. doi:10.1016/j.eswa.2016.04.018
3. David, H.W., & William, G.M. (1997). No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67-82. doi: 10.1109/4235.585893

4. Odili, J.B., Kahar, M.N.M., & Anwar, S. (2015). African Buffalo Optimization: A Swarm-Intelligence Technique. Procedia Computer Science, 76, 443-448. doi:10.1016/j.procs.2015.12.291
5. Odili, J.B., & Mohmad Kahar, M.N. (2016). Solving the Traveling Salesman's Problem Using the African Buffalo Optimization. Computational Intelligence and Neuroscience, 2016, 1510256. doi:10.1155/2016/1510256
6. Odili, J.B., Mohmad Kahar, M.N., & Noraziah, A. (2017). Parameters-tuning of PID controller for automatic voltage regulators using the African buffalo optimization. PLoS One, 12(4). e0175901. doi: 10.1371/journal.pone.0175901
7. Prakash, S., Trivedi, V., & Ramteke, M. (2016). An elitist non-dominated sorting bat algorithm NSBAT-II for multi-objective optimization of phthalic anhydride reactor. International Journal of System Assurance Engineering and Management, 7(3), 299-315. doi:10.1007/s13198-016-0467-6
8. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, Berlin, Heidelberg, 849-858. doi:10.1007/3-540-45356-3_83
9. Coello, C.A.C, & Lechuga, M.S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. Proceedings of the 2002 congress on evolutionary computation, 2002. CEC'02, 1051–1056. doi:10.1109/CEC.2002.1004388
10. Balasubbareddy, M., Sivanagaraju, S., & Suresh, C.V. (2015). Multi-objective optimization in the presence of practical constraints using non-dominated sorting hybrid cuckoo search algorithm. Engineering Science and Technology, an International Journal, 18(4), 603-615. doi:10.1016/j.jestch.2015.04.005
11. Bagherinejad, J., & Dehghani, M. (2016). A Non-dominated Sorting Ant Colony Optimization Algorithm Approach to the Bi-objective Multi-vehicle Allocation of Customers to Distribution Centers. Journal of Optimization in Industrial Engineering, 9(19), 61-74. doi:10.22094/joie.2016.230
12. Taguchi, G. (1990). Introduction to Quality Engineering; Asian Productivity Organization: Tokyo.
13. Deng, J. (1989). Introduction to grey system. The Journal of Grey System, 1(1), 1–24.
14. Lin, C.L. (2004). Use of the Taguchi Method and Grey Relational Analysis to Optimize Turning Operations with Multiple Performance Characteristics, Materials and Manufacturing Processes, 19(2), 209-220. doi: 10.1081/AMP-120029852.
15. Ghosh, T., Doloi, D., & Dan, P.K. (2016). Utilization-based grouping efficiency and multi-criteria decision approach in designing of manufacturing cells. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, doi: 10.1177/0954405416629583.
16. Dimopoulos, C., & Zalzala, A.M.S. (2000). Recent developments in evolutionary computations for manufacturing optimization: problems, solutions, and comparisons. IEEE Transactions on Evolutionary Computation, 4(2), 93-113. doi: 10.1109/4235.850651
17. Papaioannou, G., & Wilson, J.M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research. European Journal of Operational Research, 206(3), 509-521. doi:10.1016/j.ejor.2009.10.020
18. Arora, P.K., Haleem, A., & Singh, M.K.J.S. (2013). Recent development of cellular manufacturing systems. 38(3), 421-428. doi:10.1007/s12046-013-0139-z
19. Chattopadhyay, M., Sengupta, S., Ghosh, T., Dan, P.K., & Mazumdar, S. (2013). Neuro-genetic impact on cell formation methods of cellular manufacturing system design: a quantitative review and analysis. Computers & Industrial Engineering, 64 (1) (2013), 256-272. doi:10.1016/j.cie.2012.09.016
20. Ghosh, T., Sengupta, S., Chattopadhyay, M., and Dan. P.K. (2011). Meta-heuristics in cellular manufacturing: A state-of-the-art review. International Journal of Industrial Engineering Computation, 2(1), 87–122. doi: 10.5267/j.ijiec.2010.04.005
21. Gupta, Y., Gupta, M., Kumar, A., and Sundaram, C. (1996). A genetic algorithm-based approach to cell composition and layout design problems. International Journal of Production Research, 34(2), 447–482. doi: 10.1080/00207549608904913
22. Zolfaghari, S., and Liang, M. (2003) A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. Computers & Industrial Engineering, 45(4):713-731. doi: 10.1016/j.cie.2003.09.003
23. Pillai, M.V., and Subbarao, K. (2008). A robust cellular manufacturing system design for dynamic part population using a genetic algorithm. International Journal of Production Research, 46(18), 5191–5210. doi: 10.1080/00207540701332658
24. Arkat, J., Hosseini, L., and Farahani, M.H. (2011). Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm. Expert Systems with Applications, 38(8), 9597–9602. doi: 10.1016/j.eswa.2011.01.161
25. Khaksar-Haghani, F., Kia, R., Mahdavi, I., and Kazemi, M. (2013). A genetic algorithm for solving a multi-floor layout design model of a cellular manufacturing system with alternative process routings and flexible configuration. International Journal of Advanced Manufacturing Technology, 66(5), 845–865. doi: 10.1007/s00170-012-4370-2
26. Logendran, R., and Ramakrishna, P. (1995). Manufacturing cell formation in the presence of lot splitting and multiple units of same machine. International Journal of Production Research, 33(3), 675–693. doi: 10.1080/00207549508930173
27. Adenso-Diaz, B., Lozano, S., Racero, J., and Guerrero, F. (2001). Machine cell formation in generalized group technology. Computers & Industrial Engineering, 41(2), 227–240. doi: 10.1016/S0360-8352(01)00056-0
28. Chen, C.L., Cotruvo, N.A., and Baek, W. (1995). A simulated annealing solution to the cell formation problem. International Journal of Production Research, 33(9):2601–2614. doi: 10.1080/00207549508904834
29. Zolfaghari, S., and Liang, M. (2002). Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems. International Journal of Production Research, 40(9), 2141–2158. doi: 10.1080/00207540210131851

30. Solimanpur, M., Saeedi, S., and Mahdavi, I. (2010). Solving cell formation problem in cellular manufacturing using ant-colony-based optimization. International Journal of Advanced Manufacturing Technology, 50(9), 1135-1144. doi: 10.1007/s00170-010-2587-5
31. Spiliopoulos, K., and Sofianopoulou, S. (2008). An efficient ant colony optimisation system for the manufacturing cells formation system. International Journal of Advanced Manufacturing Technology, 36(5), 589–597. doi: 10.1007/s00170-006-0862-2
32. Anvari, M., Mehrabad, M.S., and Barzinpour, F. (2010). Machine–part cell formation using a hybrid particle swarm optimization. International Journal of Advanced Manufacturing Technology, 47(5):745-754. doi: 10.1007/s00170-009-2202-9
33. Durán, O., Rodriguez, N., and Consalter, L.A. (2008). A PSO-based clustering algorithm for manufacturing cell design. Workshop on knowledge discovery and data mining, 72-75. doi: 10.1109/WKDD.2008.1
34. Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., and Zaidi, M. (2006). The bees algorithm – A novel tool for complex optimization problems. Proceedings of IPROMS Conference 2006, 454–459. doi: 10.1016/B978-008045157-2/50081-X
35. Wu, T.H., Chung, S.H., and Chang, C.C. (2010). A water flow-like algorithm for manufacturing cell formation problems. European Journal of Operational Research, 205(2), 346–360. doi: 10.1016/j.ejor.2010.01.020
36. Sayadi, M.K., Hafezalkotob, A., and Naini, S.G.J. (2013). Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation. Journal of Manufacturing Systems, 32(1), 78–84. doi: 10.1016/j.jmsy.2012.06.004
37. Nouri, H., and Hong, T.S. (2013). Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations. Journal of Manufacturing Systems, 32(1), 20–31. doi: 10.1016/j.jmsy.2012.07.014
38. Soto, R., Crawford, B., Alarcón, A., Zec, C., Vega, E., Reyes, V., and Araya, I. (2016). Solving Manufacturing Cell Design Problems by Using a Bat Algorithm Approach. International Conference on Swarm Intelligence, ICSI 2016, 184-191. doi: 10.1007/978-3-319-41000-5_18
39. Olivares, R., Soto, R., and Crawford, B. (2018). Performance evaluation of the parameterless bat algorithm to solve the manufacturing cell design problem. 13th Iberian Conference on Information Systems and Technologies (CISTI), IEEE. doi: 10.23919/CISTI.2018.8399379
40. Kohonen, T. (2001). Self-Organizing Maps, 3a Ed., Spinger.
41. Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69. doi: 10.1007/BF00337288
42. Ojia, M. Kaski, S., & Kohonen, T. (2002). Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. Neural Computing Surveys 3, 1-156.
43. Buche, D., Milano, M., & Koumoutsakos, P. (2002). Self-Organizing Maps for Multi-Objective Optimization. GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, organ Kaufmann Publishers, 152-155.
44. Parashar, S., Fateh, N., Pediroda, V., & Poloni, C. (2008). Self-Organizing Maps (SOM) for Design Selection in Multi-Objective Optimization using modeFRONTIER. SAE World Congress & Exhibition. doi: https://doi.org/10.4271/2008-01-0874
45. Saraee, M., Kobbacy, K. A. H., Vahid Moosavi, S., & Rezapour, S. (2011). Application of Self Organizing Map (SOM) to model a machining process. Journal of Manufacturing Technology Management, 22(6), 818-830. doi:10.1108/17410381111149666
46. Chattopadhyay, M., Dan, P. K., Mazumdar, S. (2014). Comparison of visualization of optimal clustering using self-organizing map and growing hierarchical self-organizing map in cellular manufacturing system. Applied Soft Computing, 22, 528-543. Doi: 10.1016/j.asoc.2014.04.027
47. Boulif, M., & Atif, K. (2006). A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. Computers & Operations Research, 33(8), 2219-2245. doi: 10.1016/j.cor.2005.02.005
48. Taherkhani, M., and Safabakhsh, R. (2016). A novel stability-based adaptive inertia weight for particle swarm optimization. Applied Soft Computing, 38, 281–295. doi: 10.1016/j.asoc.2015.10.004
49. Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (1999). Self-organizing map in Matlab: the SOM Toolbox, Proceedings of the Matlab DSP Conference, Espoo, Finland, 35-40.
50. Gu, X., Sun, G., Li, G., Mao, L., and Li, Q. (2013). A Comparative study on multiobjective reliable and robust optimization for crashworthiness design of vehicle structure. Structural and Multidisciplinary Optimization, 48(3), 669-684. doi: 10.1007/s00158-013-0921-x
51. Dunnett, C. W. (1955). A Multiple Comparison Procedure for Comparing Several Treatments with a Control. Journal of the American Statistical Association, 50(272), 1096-1121. doi: 10.2307/2281208
52. Armağan M., and Arici. A. A. (2017). Cutting performance of glass-vinyl ester composite by abrasive water jet. Materials and Manufacturing Processes, 32(15), 1715-1722. doi: 10.1080/10426914.2016.1269919
53. Momber, A. W., and Kovacevic. R. (2012). Principles of abrasive water jet machining. Springer-Verlag, London.
54. Selvan, M. C. P., and Raju, N. M. S. (2012). Abrasive Waterjet Cutting Surfaces of Ceramics – An Experimental Investigation. International Journal of Scientific and Engineering Research, 1, 52-59.
55. Huang, C. Z., Wang, J., Feng, Y. X., and Zhu, H. T. (2006). Recent development of abrasive water jet machining technology. Key Engineering Materials, 315–316, 396–400. doi: 10.4028/www.scientific.net/KEM.315-316.396
56. Huang, C. Z., Hou, R. G., Wang, J., and Feng, Y. X. (2006). The effect of high pressure abrasive water jet cutting parameters on cutting performance of granite. Key Engineering Materials, 304–305, 560–564. doi: https://doi.org/10.4028/www.scientific.net/KEM.304-305.560