IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 22, 2021, accepted August 2, 2021, date of publication August 11, 2021, date of current version August 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104116

Meta-Heuristic Approach for the Development of Alternative Process Plans in a Reconfigurable Production Environment

IQRA SADAF KHAN[1], **USMAN GHAFOOR**[2], **AND TAIBA ZAHID**[3]
[1]Industrial Engineering and Management, Faculty of Technology, University of Oulu, 90570 Oulu, Finland
[2]Department of Mechanical Engineering, Institute of Space Technology, Islamabad 44000, Pakistan
[3]National University of Science and Technology, Islamabad 44000, Pakistan

Corresponding author: Taiba Zahid (taibazahid_35@yahoo.com)

ABSTRACT The need for automated production plans has evolved over the years due to internal and external drivers like developed products, new enhanced processes and machinery. Reconfigurable manufacturing systems focus on such needs at both production and process planning level. The age of Industry 4.0 focused on mass customization requires computer aided planning techniques that are able to cope with custom changes in products and explores intelligent algorithms for efficient scheduling solutions to reduce lead time. This problem has been categorized as NP-Hard in literature and is addressed by providing intelligent heuristics that focus on reducing machining time of the products at hand. However, as 70% of the lead time is consumed in non-value added tasks, it is fundamental to provide modular solutions that can reduce this time and handle part variety. To address the subject, this paper focuses on the generation of automated process plans for a single machine problem while focusing on reducing time lead time. Two evolutionary algorithms (EAs) have been proposed and compared to answer complex problem of process planning. A modified genetic algorithm (GA) has been proposed in addition to cuckoo search (CS) heuristic for this discrete problem. On testing with selected benchmark part ANC101, significant improvement was seen in terms of convergence with proposed EAs. Moreover, a novel Precedence Group Algorithm (PGA) is proposed to generate quality input for heuristics. The algorithm produces a set of initial population which significantly effects the performance of proposed heuristics. For the discrete constrained process planning problem, GA outperforms CS providing 10% more feasible scheduling options and three times lesser run time as compared to CS. The proposed technique is flexible and responsive in order to accommodate part variety, a necessary requirement for reconfigurable systems.

INDEX TERMS Process planning, reconfigurable systems, heuristics, evolutionary algorithms, genetic algorithm, cuckoo search.

I. INTRODUCTION

Process planning is a vital step for computer aided process planning (CAPP). It can be defined as the generation of sequence of operations in order to manufacture the desired product. The evolution of modern manufacturing with industry 4.0 concepts has further emphasized the research in CAPP for handling part variety. The plan should be flexible at product as well as systems level for a dynamic demand. This notion has further created the need for responsiveness on both planning and control level. In order to achieve this desired level of flexibility, six special design characteristics have been defined in literature [1] for a product or process to be termed as '*reconfigurable*'. These characteristics focus on adaptability of a manufacturing system at machine and process level that can be upgraded in the system with product development and technological changes [2]. Numerous researches in the area of reconfigurable machine tools (RMT) are available that focus on optimization of machine features that may provide desired level of flexibility to build a part family without redundancy [3], [4]. However, this study deals with another aspect of reconfiguration aimed to provide '*convertibility*' and '*integrability*' on planning and scheduling level. Reconfigurable process plans (RPP)

The associate editor coordinating the review of this manuscript and approving it for publication was Shunfeng Cheng.

promote concurrent engineering by working on machine selection and process sequencing in parallel at initial stages of manufacturing. Such dynamic process plans that are scalable to handle part variety are of significance as they affect the entire production process [5]. Moreover, to address the reconfiguration issues, processing routes can provide sequencing and processing flexibility [6]. The current paper focuses on the development of alternative plans for a single machine environment. For a job-shop having high variety and low volume products, single machine cells with higher manufacturing capabilities are of significance [7].

With the motivation of accommodating variety and producing high quality products, researchers have focused on the development of planning and scheduling for computer numeric control (CNC) with computer aided process planning [8]. Prabhu *et al.* [9] worked on rotational components with plans in the form of a decision tree. Branches of the tree indicated alternative process plans. However, the study was limited to rotational components and lacked diversity required for prismatic parts. Waiyagan and Bohez [10] proposed process planning system for a part with rotational and prismatic features termed as prisronal. The approach was more focused towards feature recognition for a part. In a similar study [11] feature based modelling was suggested for automated retrieval of process plans. However, these techniques emphasized on feature recognition methods for part manufacturing. Chung and Suh [12] developed branch and bound algorithm for optimal process plan with the aim of minimizing machining time on a CNC turning machine. The machines were considered to have a parallel machining capability in order to reduce machining time. A similar study proposed priority heuristic for CNC with parallel machining in order to reduce lead-time [13]. Such studies with parallel machining limited their use to particular part families.

While sequencing, maintaining precedence constraints in a process plan is an essential requirement. For various operations performed on the machine, generating a feasible process plan is a NP-Hard problem [14]. While generating automated process plans, the feasibility problem has been addressed with two main approaches in literature. The first one is reactive in a sense that it reacts after identification of infeasible points in the plan which violate precedence constraints [15], [16]. The other strategy is proactive as it generates process plans in a step-wise manner in order to avoid infeasible process plan [17], [18]. During each step of such algorithms, topological constraints are considered before adding an operation in the process plan chain. With these techniques, multiple process plans can be generated for a single machine problem where the number of alternatives depends on the number of precedence constraints. However, both these techniques do not pay attention on integration of a feasible plan with optimal schedule. Huang *et al.* [19] pointed out that for effective process planning, the plans should be integrated with production facility. In this new mode of research, product and process selection cannot be bounded to a simple optimization problem. In an effort by Manafi *et al.*, [20] custom rule based algorithm was proposed in order to extract machining features for process plans. In another similar study by Sormez and Khoshnevis [21] knowledge based rules were proposed for alternative process plans while taking into account the limitations of machine tool and the capital available for manufacturing.

In case of alternatives, the aim of production manager is to answer this question: which is the best plan for the given production scenario? Alternative process plans explore various possibilities of sequencing of a part. The problem can be categorized as a variant of vehicle routing problem (VRP) with additional constraints. The presence of these technological, design and topological constraints in a part add to the complexity scale of the problem making it NP-Hard. In the past, exact algorithms like branch and bound, linear and dynamic programming have been proposed to resolve this issue [22]. These enumeration techniques are applicable when the problem of interest is of smaller size. Recently, researchers have been focusing on expert-based intelligent heuristics for the solution of process planning problems [23]. As there is no such optimization algorithm that is best to achieve targeted results for every problem, different techniques are chosen depending upon the problem of interest. Afteni & Frumusanu [24] did a detailed review of optimization techniques that have been applied in non-linear constrained problems. Literature conducted in the field of applied computing suggests that in comparison with exact classical approaches, evolutionary algorithms can provide improved solutions rather than sub-optimal solutions [14]. Numerous evolutionary algorithms have been proposed for the optimization of process plans. Krishna and Rao [25] proposed an ant colony algorithm (ACO) in addition to a simulated annealig (SA) approach but with a different objective [26]. Another variation of SA was suggested by Li *et al.* [15] with a feature recognition method used to extract features and develop alternative route plans for the part. Salehi and Bahreininejad [27] provided a hybrid genetic algorithm (GA) designed to obtain optimal process plan in a job-shop environment. However, these evolutionary algorithms were unable to accommodate complex part features and the non-conformities which may occur by not satisfying the technical constraints. Moreover, the proposed strategies work for a certain set of operations and lacked responsiveness.

For selection of best among alternatives, various aspects of manufacturing operations may be considered for defining objective function. Numerous studies have been performed for optimization of flow shops sequencing with the objective of minimizing time [15], [28]. The studies differ in their method for selecting next set of solutions (next generation of population) from the previous one. The methodology was combined with an intelligent search technique for clustering in a job shop environment. In a similar effort by Reddy *et al.* [29], they used GA for obtaining near optimal alternative feasible plans for current set-up of production floor in order to make it dynamic and reduce tardiness. Such objectives focused on overall reduction of lead time at the production floor. In case of single machine, the focus has been towards parameteric optimization with target of achieving better machining quality for the product [11], [30]. However, it has been established that non-value added tasks such as setup and tool change utilize 70% of the total time required for job completion [31]. With this premise, an objective function can be chosen that minimizes time by reducing non-value adding tasks. Moreover, as the set-up tasks are reduced, the inherent problem of variability in machinig can also be effectively handled.

The research presented in this paper draws its motivation from the aforementioned shortcomings in literature. The problem of routing and process planning is well-established NP-hard problem in literature. Exact enumeration methods are not considered appropriate to find solutions to such problems. Because of the probabilistic nature of heuristic algorithms, problems such as slow convergence, local optima, variance in end results and longer computation times arise. None of the methods have been proved perfect to solve industrial scale problems. However, among the various proposed heuristics, simulated annealing (SA), GA (genetic algorithm) and PSO (particle swarm optimization) are the most widely discussed techniques in literature [32]. SA is a single point search technique that progresses with a single solution and a single search direction due to which it is criticized for slow convergence. PSO is a population-based technique with multiple search directions but is most widely used for continuous variable problems while routing and process planning is a discrete problem. GA is a well-known population-based technique in the area of discrete constrained problems that are NP-Hard in nature. GA has been used extensively in literature for scheduling and routing problems. However, one of the major demerits of GA is slow convergence and variance in results. The leading factor of variance is its stochastic search nature. In comparison, CS is a relatively new heuristic introduced in 2010 and has been reported to outperform GA for various scheduling problems and benchmarks [33]. However, there was a gap in application of CS for constrained and discrete problem of process planning. This motivated to apply CS algorithm on single machine process planning problems and investigate its performance with most widely adopted GA. Most of the researchers have focused on using GA that produced better results with reasonable computation time as compared to exact and exhaustive search algorithms [27]. However, multiple random decision parameters of GA such as selection methods, crossover and mutation operator and population size make this heuristic more ambiguous. In the current approach, a new version of these decision parameters has been adopted to provide better optimal results and have been compared with a bench mark problem. In addition, cuckoo search (CS) heuristic approach has been developed for process planning in a single machine environment. GA and CS are population-based techniques which require a set of initial solutions to explore the search space. As the quality of solution generated from meta-heuristics significantly depends on initial population [34], the current study provides a new approach for generating alternatives with precedence group algorithm (PGA). It is initiated for the creation of initial population that produces feasible and diversified process plans for initial population. For both algorithms, the quality of initial solutions improved search directions and provided faster convergence to global optima. The PGA can be modified easily for a part family and increases reconfigurability in the system by adding integrability. Performance of CS was found to be lower than GA for process planning problem due to its discrete and constrained nature. The next section provides detail on this algorithm followed by the proposed heuristics for single-machine process planning problem.

II. METHODS

A. PROCESS PLAN GENERATION

Process planning is defined as a group of instructions followed in a step-wise manner to manufacture a product. Shabaka and Elmaraghy [35] have classified various types of process plans in three categories based on their precision level (multi-domain, macro and micro process planning level). Multi-domain is the least detailed form of process planning concerned with initial decisions required before planning such as assembly method and material selection. Macro process planning defines set of instructions regarding sequencing of operations. Unlike dedicated matching systems (DMS) with fixed setups, this process needs frequent revisions due to product design changes in RMS. Micro level planning is the search of best sequence defined by criteria of manufacturers at earlier stages of planning.

In literature, macro level process planning has been adopted for generating alternative plans. The present work proposes PGA for generation of alternatives to find optimal sequence at micro level. The proposed PGA provides an extension of the algorithm proposed by Zahid and Baqai [36]. Process planning problem has two stages which are operation selection and sequencing. This particular problem cannot be correlated with the usual travelling salesman problem (TSP). In TSP, for 'n' number of cities, n! would be the number of total possible routing options. However, in case of process planning, options would be less than $n!$. This happens due to the technical feasibility constraints of the part under consideration. Various types of datum, geometrical and logical constraints need to be kept in mind for developing feasible process plans.

The proposed methodology has been developed as a general applicable methodology with following assumptions:

- The methodology has been proposed for single machine problem.
- Simultaneous machining is not allowed.

There are no limitations on geometry of the part. For PGA, a precedence group matrix (PGM) is developed to group operations in a column as per their constraints. Each group represents a level of precedence between operations and any member from a group can be randomly selected. After all the elements of a group are selected, the algorithm moves

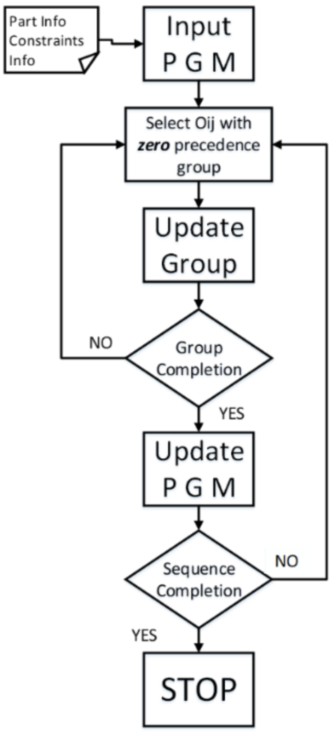

FIGURE 1. Flowchart for precedence group algorithm (PGA).

towards the next group. No precedence is established among the activities of a same group. If an activity is written twice in a group, it is not a *necessary* activity and algorithm may continue to the next group without selecting that activity. However, for a complete process plan, every activity must be included in the sequence, which is assured by a check placed in the flowchart. The flow chart for generation of process plan is show in Figure 1.

A hand spike, an industrial part frequently used in literature [29], [37], shown in figure 2 is used for description of process plan generation method. It has been adapted frequently in literature to suggest and compare optimization algorithms and process plan methodologies. This component is a pre-formed part and it requires 16 operation to be done on a machining centre. The part drawing has been appropriately modified to include constraints like datum and location. Four types of machining features (milling, drilling, boring and reaming) need to be performed for manufacturing the desired part. Description of machining operations is provided in table 1.

1) INPUT DATA

Algorithm starts with information shown in figure 1 while the precedence group matrix PGM is created by analysing technical constraints of the part. The machining steps in table 1 along with part geometry and drawings are used as an input here.

This matrix has been developed through network diagram of the part shown in figure 3. There is no precedence in

TABLE 1. Machining features for example part.

Operation ID	Machining Detail	Tool Selection
A1	Rough plain milling	Face mill cutter
A2	Finish plain milling	Face mill cutter
B1	Milling the boss	End mill cutter
B2	Drilling	Drill
B3	Boring	Boring Bar
B4	Reaming	Reamer
B5	Finish Reaming	Reamer
C1	Drilling	Drill
C2	Milling	End Mill
D1, E1, F1	Drilling	Drill
D2, E2, F2	Tapping	Tapping tool
G	Drilling	Drill

the groups and the number of operations may vary. Each operation in a group, i.e. O_{ij} stands for *ith* operation in group *j* where *n* is the total number of operations which need to be performed. There is no priority among operations in the same group and are chosen randomly. It should be noted that no repetition of an operation is allowed in the same group however, operations may repeat in different groups. Such operations are termed as *flexible* and algorithm can jump to the next group without performing these operations.

2) SELECTION OF ZERO PRECEDENCE OPERATIONS

The machining operations having zero precedence are those present in the first group of PGM. In this example, operation number A1 is going to have zero precedence.

In case of multiple zero precedence operations, a random operation would be selected from the list. In this particular case, operation A1 (op1) is selected from group I and added in the sequence array.

3) UPDATE ZERO PRECEDENCE OPERATIONS

To continue with sequence array, the group is updated in PGM deleting the already selected operation (i.e. op1). This step ensures that there is no repetition that may result in an infeasible process plan having an operation performed twice. Moreover, with this approach, input to the optimization algorithm is feasible thus reducing the computation time in later stages since optimizer searches for the optimal solution without worrying about the feasibility test. The operation selected here is deleted from the PGM matrix in order to avoid repetition of the same operation in sequence array.

4) GROUP COMPLETION

The algorithm verifies if all the *necessary* operations of the group have been added in the sequence array. It means that the flexible tasks may be performed in the next group and the group can be removed at this stage. If yes, the algorithm updates PGM by deleting the current group and the next

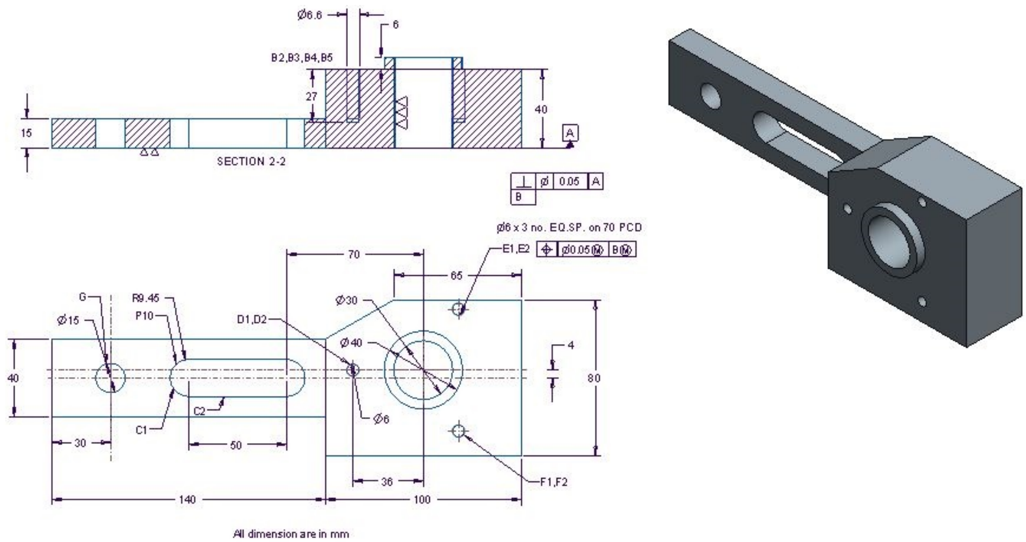

FIGURE 2. Part Features for hand spike.

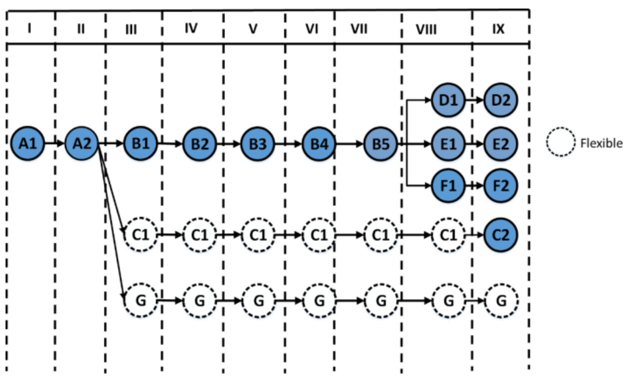

FIGURE 3. Network diagram for example part.

group becomes zero precedence group. If not, step 2 and 3 are repeated.

B. SEQUENCE COMPLETION

After updating PGM, sequence is checked for completion at the current stage. The total number of elements in a sequence array should be equal to the total number of operations. For instance, a complete sequence for the part may result in the following sequence array:

A1 → A2 → B1 → B2 → B3 → B4 → B5 → C1 → G → D1 → E1 → F1 → D2 → F2→ E2 → C2

C. OPTIMIZATION OF PROCESS PLANS

The proposed methodology generates given number of alternative process plans in according with the chosen population size. Instead of single solution, meta-heuristic algorithms start with a set of solutions where the quality of output is significantly dependent on these initial solutions [38]. This kind of problem cannot be constrained to a finite solution space with simple explicit constraints which makes it difficult to solve by exact optimization techniques. The present study suggests customized evolutionary approach for the generation of optimal process plans. In later part of this section, GA and CS methodologies for optimized process plan will be discussed in detail. Figure 4 describes these two individual population-based heuristics. The initial population input is generated with the use of PGA. The flowchart describes the main highlighted difference between these two algorithms. The creation of new population and hence, the new set of solutions is different for both heuristics. While GA uses crossover and mutation, CS makes the use of levy flight phenomenon.

1) FITNESS FUNCTION

For an optimal process plan, an objective function needs to be assigned for the purpose of comparison. To account for the non-value added time, minimal tool and set up changes are set as the desired criteria in the current work. Weightage method has been used for assigning weights to both criteria. Tool change matrix is composed of binary values in a matrix form. In case of a tool change from one to the next operation in sequence array (i.e. $TC_{i\rightarrow i+1}$, *where i is the operation index*), a value equivalent to 1 is assigned in the matrix and otherwise it is 0. Number of columns and rows represent total number of operations. Hence, all the diagonal entries show relation between same operations and will always be zero. Total number of tool changes in a sequence can be represented as;

$$\boldsymbol{TC_k} = \sum\nolimits_{\boldsymbol{i\rightarrow i+1}}^{\boldsymbol{n-1\rightarrow n}} TC_{i\rightarrow i+1} \tag{1}$$

where $\boldsymbol{TC_k}$ = number of tool changes in sequence k

n = total number of operations in a sequence

Same methodology will be used for setup change matrix i.e. $SC_{i\rightarrow i+1}$.

In the same manner, value of 1 is assigned when a setup change is required for manufacturing and 0 otherwise.

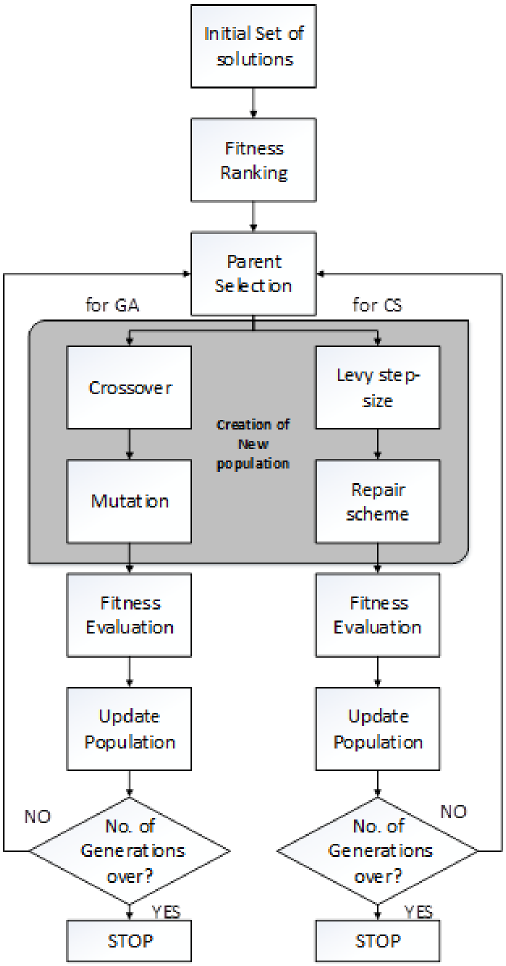

FIGURE 4. **Flowchart for GA and CS algorithm.**

It should be noted that setup change is mainly linked here with tool approach direction (TAD) considering a machine with single axis for machining (z-axis). However, for a multi-axis machine, the matrix can be constructed in the same manner.

$$SC_k = \sum_{i\rightarrow i+1}^{n-1\rightarrow n} SC_{i\rightarrow i+1} \tag{2}$$

Fitness criteria has been considered as;

$$(minimize)Z = W_1 * TC_k + W_2 * SC_k \tag{3}$$

As an example, TC_k and SC_k for the part in figure 2 is calculated here in table 2. The first row represents a process plan generated with PGA. The tool requirements for the operations have already been displayed in table 1.

The fitness function is used to select best solutions from the search space fort the next steps of proposed meta-heuristics. Various methods for example roulette selection, tournament selection, elitism and rank selection etc., have been proposed for this purpose [39]. Elitist method [40] has been used for selection purposes in the present work for GA. This method selects best ranked individuals as parents while discarding the other ones. For CS, a fraction of poor solutions are removed from the next iteration while keeping population size constant. Solutions are ranked based on their performance as per the objective function and are included in the next iteration based on probability of acceptance i.e. P_a which is kept between 0.1-0.5 [41].

2) CREATION OF NEW POPULATION

After selecting parents for creating next group of population, evolutionary algorithms explore different strategies for exploring the global search space for optimal solution. In the subsections, we propose two different heuristics (i.e. GA and CS) for creation of new set of solutions.

3) GENETIC ALGORITHMS

Crossover and mutation are performed for combining fitter parents for the creation of new population. Mutation operator is used to introduce diversification in order to explore search space efficiently. Crossover is used to change the programming of chromosomes between generations. Crossover is performed in many variations such as single, multiple or uniform crossover. These traditional strategies cannot be adopted for this specific sequencing problem. Consider an example below in figure 5 having a string of ten operations with single point crossover. It can be observed that after crossover, the new solutions (represented as children) are infeasible as some operations are repeated (e.g. op9 in child 1) while some operations are not performed (e.g. op8 in child 1).

To deal with this issue positional crossover is used in the proposed methodology. By selecting two crossover points, a temporary solution is generated for children as depicted in figure 6. Temporary solution is then completed by inverting the order of operations. For instance, to create Child 1, the temporary segmented part is completed by placing same operations as existed in parent 1. The difference lies in their sequence which is adopted from the second parent. This ensures that the new solutions are feasible and maintain precedence constraints as well. The same procedure is repeated for generating C2.

The next step is to perform mutation which provides diversification in the population in order to explore search space more efficiently. Mutation is considered necessary in order to avoid a locally optimal solution. Two mutation sites are randomly chosen to swap operations. Repetition will not occur in this case but the solution may still become infeasible due to the violation of precedence constraints. Based on a study by Carlson [42], the present study assigns penalty (via penalty matrix) in case of this violation which is added in the fitness function. This, in turn, makes the individual solution less likely to continue in the next population.

$$\text{Fitness Function} = W_1 * TC_k + W_2 * SC_k + P_k \tag{4}$$

4) CUCKOO SEARCH

Several studies on engineering problems have shown that among all metaheuristics, CS algorithms tend to provide best results [43]. In this algorithm, the behaviour of cuckoo

						Crossover point				
Parent 1	2	4	1	3	9	8	7	6	5	10
Parent 2	1	3	4	8	2	6	9	10	5	7
Child 1	2	4	1	3	9	6	9	10	5	7
Child 2	1	3	4	8	2	8	7	6	5	10

FIGURE 5. Single-point crossover with infeasible plans.

Parent 1	2	4	1	3	9	8	7	6	5	10
Parent 2	1	3	4	8	2	6	9	10	5	7
Temp 1	2	4	1	X	X	X	X	6	5	10
Temp 2	1	3	4	X	X	X	X	10	5	7
Child 1	2	4	1	3	8	9	7	6	5	10
Child 2	1	3	4	2	9	8	6	10	5	7

FIGURE 6. Proposed crossover technique.

TABLE 2. Fitness for process plan.

Operation	A1	A2	B1	B2	B3	B4	B5	C1	E1	D1	F1	D2	E2	F2	G	C2
$TC_{i\rightarrow i+1}$	0	0	1	1	1	1	0	1	0	0	0	1	0	0	1	1
$SC_{i\rightarrow i+1}$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$Tool\ change\ for\ sequence\ k = TC_k = 8$, $Setup\ change\ for\ sequence\ k = SC_k = 1$																

breeding is used for the development of a novel meta-heuristic [44]. The concept of Levy flights is used to generate next set of solutions in order to ensure effective search for optimal solution. It is a type of random walk and follows Markov's chain phenomena [45]. It implies that the future value or state of a variable depends upon its present value and not on its past. The new individual is calculated using equation 5. It can be seen that the second term on right hand side of the equation follows power law [46] with heavy tailed probability distribution. This categorizes infinite variance and infinite mean and is used as a step-function for obtaining a new solution. The step function enables CS algorithm to search solution space with a random walk and reach a global optimum. The levy flight concept enables CS to find new solutions with a structure random walk [41]. It has been proved in literature that structured random walks are better in finding a global solution than random walks proposed in other heuristics such as PSO and ant colony optimization [33].

$$y_{t+1} = y_t + \alpha^* t^{-\beta} \tag{5}$$

This heuristic has been applied with standard [47] and modified levy flight [48] to solve large scale problems. However, most of the applications deal with continuous variables. Despite showing better performance for multiple engineering problem, application of CS for process planning is not well known. A few studies [49], [50] discuss application of CS for discrete engineering problems including job-shop scheduling. We propose CS algorithm with modified step size technique in order to generate new solutions (i.e.y_{t+1}) and compare it with GA discussed in the above section. The solution in this case represents a string of operations. For instance, for a problem with 14 operations, a solution is represented with a string of 28 digits as shown in table 3. In order to accommodate this long sequence, we multiply step size with a scaling factor α which equals 10e5. α is the scale adjusting factor that is set according to the encoded problem. This factor can set to be of increased order in case of a longer string with greater number of operations. For a specific encoded problem size, this factor is treated as a constant. In order to avoid convergence with local optima, we have selected value of β to be 2. For $\beta > 3$, the distribution is with limited variance which drives the algorithm to converge locally [51].

As it can be observed from table 3, with levy flight, the new solution can be infeasible in multiple ways. For instance, op9 is repeated twice in the plan while op5 is neglected. To avoid this infeasibility, a repair strategy is proposed.

5) REPAIR SCHEME

A repair scheme solves the problem of infeasible process plans. After the generation of new process plan (i.e.

TABLE 3. Process plan from CS.

y_t	01	02	13	03	07	09	08	04	11	10	05	14	12	06
Step size $(\alpha * t^{-\beta})$ for instance at t = 5											04	00	00	00
y_{t+1}	01	02	13	03	07	09	08	04	11	10	09	14	12	06

TABLE 4. Repaired process plan.

Repaired	1	2	13	3	7	5	8	4	11	10	9	14	12	6

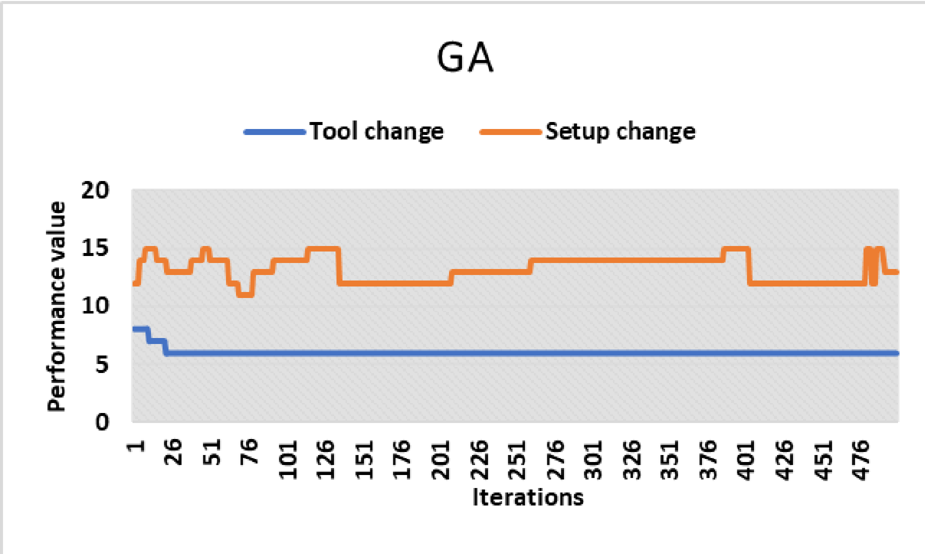

FIGURE 7. Tool change results (Proposed GA) for ANC101.

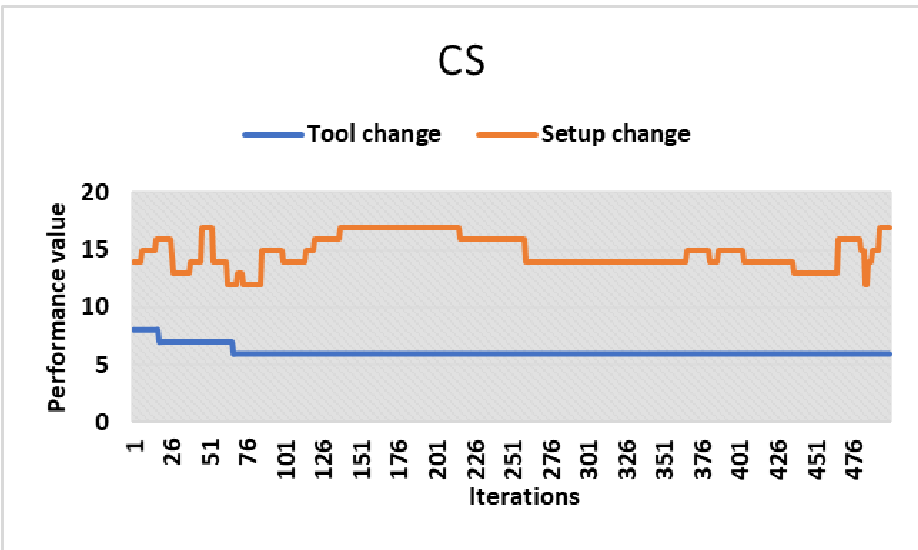

FIGURE 8. Tool change results (Proposed CS) for ANC101.

y_{t+1}), it is checked for feasibility. If an element in the string is greater than the number of operations, then previous element is retained in the sequence. However, if an element is repeated, it is swapped with the parent solution (i.e. y_t). For the above case, the new solution will swap op5 and op9 for a feasible child as depicted in table 4. The fitness of solution is calculated by using equation 4.

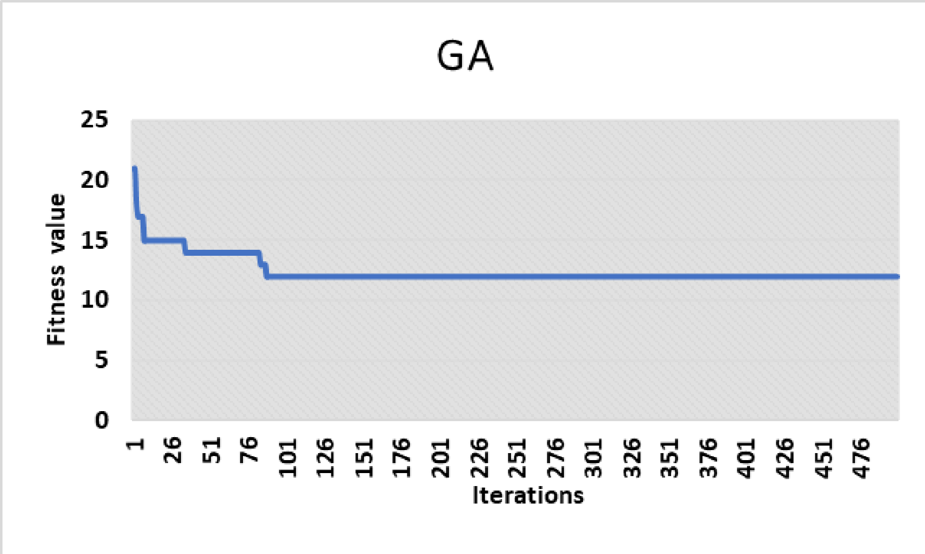

FIGURE 9. Fitness function obtained by GA for ANC 101.

6) STOPPING CRITERIA

Limit criteria cannot be used in our problem due to the large search space and absence of an optimal benchmark. Moreover, literature argues the use of stalling limit as stopping criterion while using elitism as parent selection. The reason behind this is the erratic behaviour of algorithms observed by the use of this combination [34]. Hence, number of generations/iterations is used as a stopping criterion in this paper.

III. EXPERIMENT AND RESULTS

For the examination of proposed algorithms, we chose existing benchmark part (ANC101) that has been used in literature for such CAPP problems [11]. With 14 machining features, a total of 28 operations need to be performed for part manufacturing. The inputs required for optimization of the part (e.g. Network diagram with PGM, setup, tool change and penalty matrix) has been provided in (Supplementary materials). Matlab 2016b was used for simulation of both heuristics and their comparison. Crossover and mutation rate are stochastic parameters for GA. For the present case, crossover and mutation rate was selected to be 0.9 and 0.1 respectively which has been recommended in literature [3], [52]. While comparing the process plans obtained with GA, it was observed that process plans obtained by Kumar & Deb [53] were unable to maintain precedence constraints which resulted in infeasible plans. The proposed technique with GA algorithm considered objective functions separately. It was able to obtain an optimal process plan with six tool changes where as the number of setup changes varied from 11-15. The results were obtained in 147th iteration while using elitist model for selection. Below are the results displayed in figure 7 and 8 by using GA showing tool changes during iterations. It can be seen that the optimal process plans (table 5 and 6) had 6 number of tool changes which was obtained after only 4th generation. Moreover, the generated process plan does not violate any constraint. While comparing the proposed GA and CS, we used combined objective function (equation 4) giving equal weightage to setup and tool change. It can be seen in figure 9 that GA tends to provide better results.

IV. DISCUSSION

The GA algorithm was tested on a benchmark problem and performed well. The optimal value was achieved with faster convergence. This was possible due to the fact that as compared to previously proposed GA, the algorithm starts with feasible initial population and does not undergoes feasibility tests but focuses on searching optimal solution in search space. CS was compared with GA on process planning problem and although, CS outperforms GA for continuous problems, it did not perform well in case of discrete constrained process planning problem. Figure 7 displays the results for GA tested on ANC101 with desired objective minimum number of tool changes. Variation in setup changes was also recorded during iterations and was seen to fluctuate between 11-15. It can be seen in table 6 that best optimization run for GA gave optimal result at 4th iteration. However, due to stochastic nature of the algorithm, the runs were performed 100 times on average and it was seen that 70% of the time, optimal values were achieved in first 15 iterations and reached to 100% for 30 iterations. It should be noted that these results were obtained by keeping the population size of 50. Figure 8 displays the results for CS algorithm with the same objective of minimizing number of tool changes. As compared to GA, the convergence to optimal value for CS was slower with the best run giving optimal value at 67th iteration. However, on average, 100% of the time optimal results were achieved with less than 100 iterations. The total number of

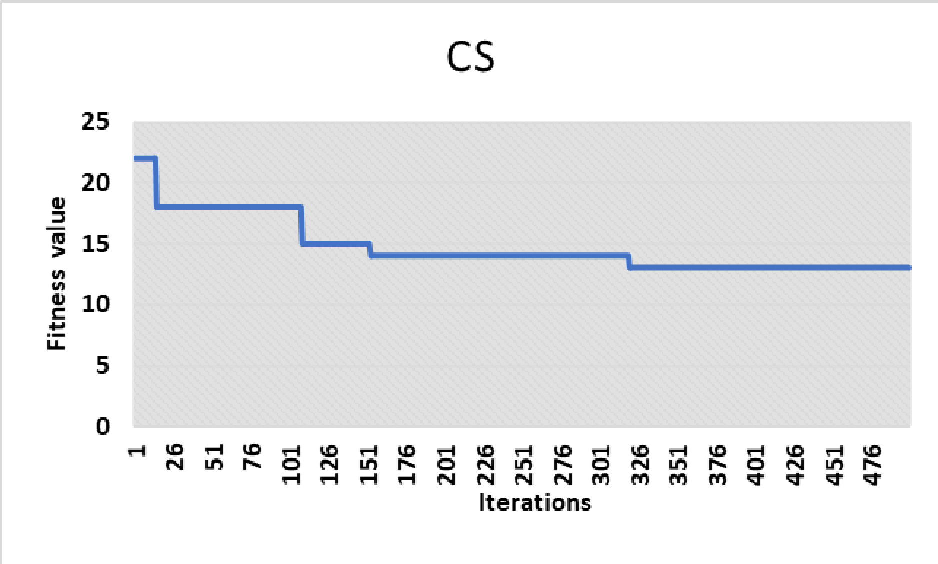

FIGURE 10. Fitness function obtained by CS for ANC101.

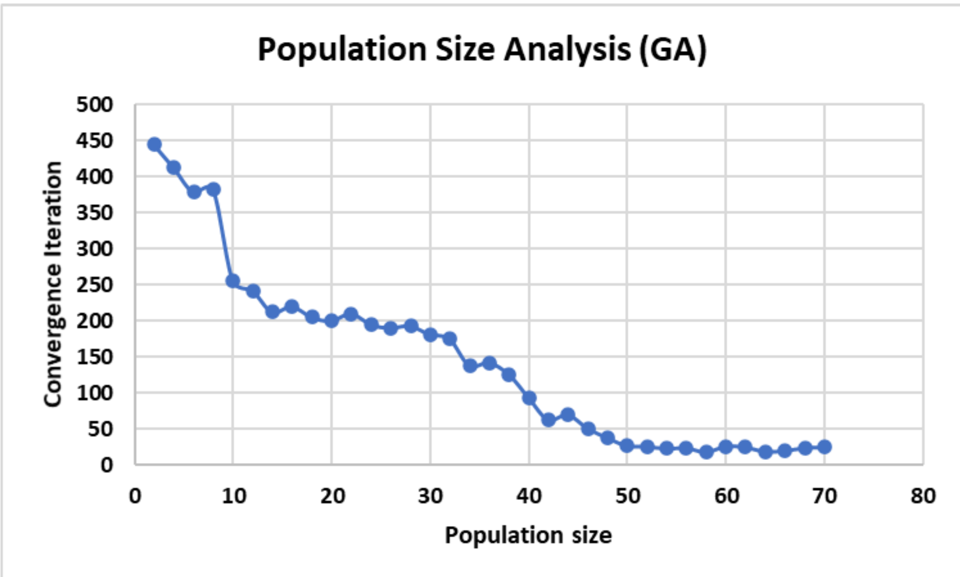

FIGURE 11. Population Size Vs Convergence Iteration in GA for ANC101.

TABLE 5. Optimal process plan for ANC101.

Operations	11	1	19	5	3	20	2	6	18	7	4	13	12	8(*cont..)
$TC_{i\rightarrow i+1}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$SC_{i\rightarrow i+1}$	0	0	0	1	0	1	0	0	1	1	0	0	0	1
Op (*cont.)	25	26	10	9	21	14	15	16	17	27	23	22	24	28
$TC_{i\rightarrow i+1}$	1	1	1	0	0	1	0	0	0	0	0	1	0	1
$SC_{i\rightarrow i+1}$	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$Tool\ change\ for\ sequence\ k = TC_k = 6$, $Setup\ change\ for\ sequence\ k = SC_k = 6$														

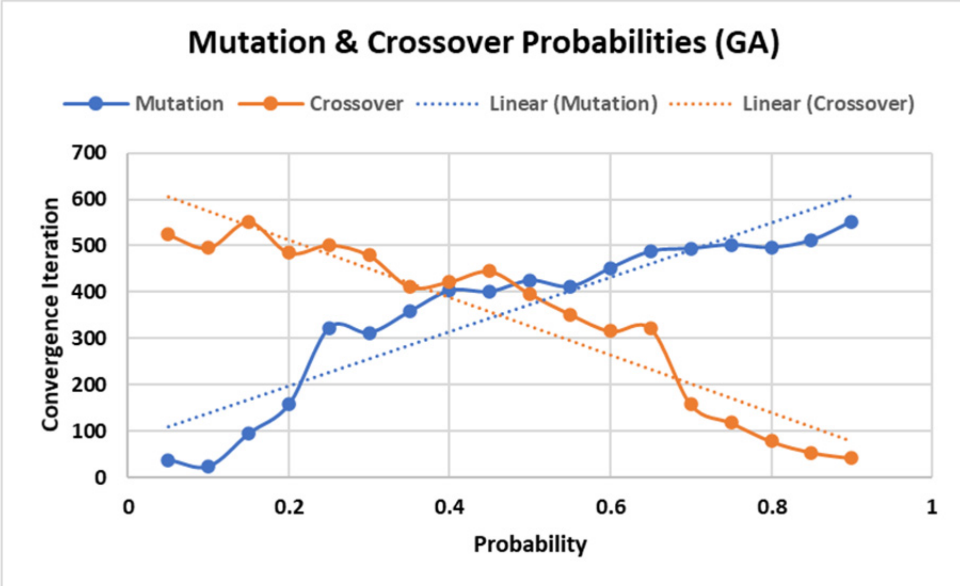

FIGURE 12. Impact of crossover and mutation probabilities in GA for ANC101.

TABLE 6. Comparison of results for ANC101.

Methodology	Optimum value for tool change	Optimum value for setup change	No of iterations
GA proposed by Kumar & Deb [44]	6	6	147
Proposed GA	6	6	4
Proposed CS	6	7	67

setup changes varied from 12-17. This is possible due to the fact that in order to apply CS methodology on process planning problems, we have to produce new solutions/sequences via levy flights. But when levy flight concept is applied on a process plan to create new population, it violates precedence constraints which leads to increased penalties on the generated sequence. Such sequences are then eliminated for next simulation run due to poor fitness value. This proves our initial point that one algorithm might not perform best for every problem under consideration and decision has to be taken for the appropriate choice of optimization algorithm. Figure 9 and 10 show results against the total fitness function defined in equation 4 with equal weightage given to both setup and tool change criteria. Similar results were observed with CS converging slower as compared to GA. On detail analysis of the results, it was seen that CS population results in a large number of penalties that compromises their fitness to be selected as parents for the next iteration. Experimental evaluations were also performed to investigate the effect of stochastic parameters of population size, mutation and crossover rate in GA on the performance. Figure 11 shows the effect of population size (mutation rate 0.1, crossover rate 0.9) on convergence performance of GA heuristic. The performance was measured by plotting the iteration on which optimal value was achieved for each sample of population size. It can be seen that lower population size seems to provide slower convergence while a size of 50-60 provides the best results. Figure 12 shows the impact of mutation and crossover rate variation on the convergence performance. A higher mutation rate results in slow convergence and higher computation time requirements due to violation of constraints. The optimum value for mutation ranges from 0.05-0.1 while a higher value results in slower convergence. It is because mutation results in increased number of penalties for the generated process plan. Such process plans are then eliminated from the population by the elitist selection method. On the other hand, lower crossover rate results in limiting the search direction providing repetition of solutions in next generations/iterations. For CS, the stochastic parameter is probability of acceptance i.e. P_a [0.1-0.5] with a step size of 2 as recommended in literature [41]. A higher value of Pa exhibits that there are more chances of selecting a poor solution for the next generation. A value of 0.5 converged rapidly to provide a sub-optimal fitness value of 15 for ANC-101 while a value of 0.25 and 0.3 provided fitness value of 13. A lower value of Pa ranging from 0.1-0.20 did not provided any improved solutions but converged slowly towards optimum solution. The investigation shows that although GA has more stochastic parameters as compared to CS, it is capable to provide better results in case of discrete problem of process planning. CS should be explored more in terms of step size to generate feasible solutions.

V. CONCLUSION

The present work lies in the domain of optimization in industrial systems. It aims at generating optimal process plans for a part/part family for a single machine. Along with that, search areas in optimization are also discussed in order to yield optimal process plan from the alternatives. The proposed GA outperforms other algorithms in literature and provides improved performance in terms of fitness function as well as computation time. Although CS algorithm is argued to provide good results for continuous problems, its performance in case of discrete problems with complex constraints is somewhat limited. The step-size used to generate new sequences should be further explored. The proposed PGM matrix provides a novel way to obtain initial solution set. The matrix can be easily modified to add new features in a product making it robust. Furthermore, variety of solutions generated by PGM provide a good quality of initial population which improves performance of proposed evolutionary strategies. Percentage of the optimal solution provided by GA was more than 80. However, in case of CS, this percentage was reduced to 70. For ANC101 part, the best solution was always obtained in less than 50 iterations. Furthermore, run time required by CS was 3 times higher than of GA.

Certain future work areas can provide a wider range of practical applications of the study. Parallel sequencing can be considered for cases in which minimum machine time is required and precision is not an issue. Along with the specified criteria, parameter optimization can be performed resulting in suggesting optimal feed, speed and depth of cut for the part. This will require detailed literature review on tooling and cutting force directions. The problem should be extended to multi-modal resource constrained scheduling problem with multiple machines and jobs. CS algorithm will be explored in future work to investigate its strength in manufacturing industry.

AUTHOR CONTRIBUTION

ISK conducted simulations and experiments. UG verified the data analyses and contributed in making figures in the revision process. TZ has suggested the theoretical aspects of the study, corrected the manuscript, and supervised all the process from the beginning. All the authors have approved the final manuscript.

REFERENCES

[1] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *J. Manuf. Syst.*, vol. 29, no. 4, pp. 130–141, Oct. 2010, doi: 10.1016/j.jmsy.2011.01.001.

[2] A. R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui, "Reconfigurable manufacturing systems from an optimisation perspective: A focused review of literature," *Int. J. Prod. Res.*, vol. 1, pp. 1–19, Oct. 2020, doi: 10.1080/00207543.2020.1813913.

[3] M. Liu, L. An, J. Zhang, F. Chu, and C. Chu, "Energy-oriented bi-objective optimisation for a multi-module reconfigurable manufacturing system," *Int. J. Prod. Res.*, vol. 57, no. 19, pp. 5974–5995, Oct. 2019, doi: 10.1080/00207543.2018.1556413.

[4] N. Brahimi, A. Dolgui, E. Gurevsky, and A. R. Yelles-Chaouche, "A literature review of optimization problems for reconfigurable manufacturing systems," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 433–438, 2019, doi: 10.1016/j.ifacol.2019.11.097.

[5] H. Besharati-Foumani, M. Lohtander, and J. Varis, "Intelligent process planning for smart manufacturing systems: A state-of-the-art review," *Procedia Manuf.*, vol. 38, pp. 156–162, Jan. 2019.

[6] Q. Xia, A. Etienne, J. Denten, and A. Siadat, "Reconfigurable machining process planning for part variety in new manufacturing paradigms: Definitions, models and framework," *Comput. Ind. Eng.*, vol. 115, pp. 206–219, Jan. 2018.

[7] Y. Su, X. Chu, Z. Zhang, and D. Chen, "Process planning optimization on turning machine tool using a hybrid genetic algorithm with local search approach," *Adv. Mech. Eng.*, vol. 7, no. 4, Apr. 2015, Art. no. 1687814015581241.

[8] M. Al-Wswasi, A. Ivanov, and H. Makatsoris, "A survey on smart automated computer-aided process planning (ACAPP) techniques," *Int. J. Adv. Manuf. Technol.*, vol. 97, nos. 1–4, pp. 809–832, Jul. 2018.

[9] P. Prabhu, S. Elhence, and H. Wang, "An operation generator network for computer aided process planning," *J. Manuf. Syst.*, vol. 9, no. 4, pp. 283–291, 2003.

[10] K. Waiyagan and E. L. J. Bohez, "Intelligent feature based process planning for five-axis mill-turn parts," *Comput. Ind.*, vol. 60, no. 5, pp. 296–316, Jun. 2009.

[11] V. Krishna, P. Shankar, and N. V. S. Surendra, "Feature based modeling and automated process plan generation for turning components," *Adv. Prod. Eng. Manage.*, vol. 6, no. 3, pp. 153–162, 2011.

[12] D.-H. Chung and S.-H. Suh, "ISO 14649-based nonlinear process planning implementation for complex machining," *Comput.-Aided Des.*, vol. 40, no. 5, pp. 521–536, May 2008.

[13] Y. S. Kim, Y. Kim, F. Pariente, and E. Wang, "Geometric reasoning for mill-turn machining process planning," *Comput. Ind. Eng.*, vol. 33, nos. 3–4, pp. 501–504, Dec. 1997.

[14] V. Oduguwa, A. Tiwari, and R. Roy, "Evolutionary computing in manufacturing industry: An overview of recent applications," *Appl. Soft Comput.*, vol. 5, no. 3, pp. 281–299, Mar. 2005.

[15] W. D. Li, S. K. Ong, and A. Y. C. Nee, "Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts," *Int. J. Prod. Res.*, vol. 40, no. 8, pp. 1899–1922, 2002.

[16] Y. Yun, H. Chung, and C. Moon, "Hybrid genetic algorithm approach for precedence-constrained sequencing problem," *Comput. Ind. Eng.*, vol. 65, no. 1, pp. 137–147, May 2013.

[17] Y. Yun and C. Moon, "Genetic algorithm approach for precedence-constrained sequencing problems," *J. Intell. Manuf.*, vol. 22, no, 3, pp. 379–388, 2011.

[18] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *Eur. J. Oper. Res.*, vol. 140, no. 3, pp. 606–617, Aug. 2002.

[19] S. H. Huang, J. Mei, H. C. Zhang, and S. R. Ray, "An integrated process planning project," Dept. Ind. Eng., Texas Tech. Univ., Austin, TX, USA, Tech. Rep., 1994.

[20] D. Manafi, M. Nategh, and H. Parvaz, "Extracting the manufacturing information of machining features for computer-aided process planning systems," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 231, pp. 2072–2083, Oct. 2017.

[21] D. N. Sormaiz and B. Khoshnevi, "Generation of alternative process plans in integrated manufacturing systems," *J. Intell. Manuf.*, vol. 14, pp. 509–526, Dec. 2003.

[22] Y. Crama, "Combinatorial optimization models for production scheduling in automated manufacturing systems," *Eur. J. Oper. Res.*, vol. 99, no. 1, pp. 136–153, May 1997.

[23] S. P. L. Kumar, "Knowledge based expert system in manufacturing planning: State-of-the-art review," *Int. J. Prod. Res.*, vol. 57, nos. 15–16, pp. 4766–4790, 2016.

[24] C. Afteni and G. Frumusanu, "A review on optimization of manufacturing process performance," *Int. J. Model. Optim.*, vol. 7, no. 3, pp. 139–144, 2017.
[25] A. G. Krishna and K. M. Rao, "Optimisation of operations sequence in CAPP using an ant colony algorithm," *Int. J. Technol.*, vol. 29, pp. 159–164, May 2006.
[26] G. H. Ma, Y. F. Zhang, and A. Y. C. Nee, "A simulated annealing based optimization algorithm for process planning," *Int. J. Prod. Res.*, vol. 38, no. 12, pp. 2671–2687, 2000.
[27] M. Salehi and A. Bahreininejad, "Optimization process planning using hybrid genetic algorithm and intelligent search for job shop machining," *J. Intell. Manuf.*, vol. 22, no. 4, pp. 643–652, Aug. 2011.
[28] X. Pang, H. Xue, M. L. Tseng, M. K. Lim, and K. Liu, "Hybrid flow shop scheduling problems using improved fireworks algorithm for permutation," *Appl. Sci.*, vol. 10, pp. 1–16, Jan. 2020.
[29] S. V. B Reddy, M. S. Shunmugam, and T. T. Narendran, "Operation sequencing in CAPP using genetic algorithm," *Int. J. Prod. Res.*, vol. 37, pp. 1063–1074, Mar. 1999.
[30] R. Stryczek, "A metaheuristic for fast machining compensation error," *J. Intell. Manuf.*, vol. 27, no. 6, pp. 1209–1220, 2016.
[31] M. E. Merchant, "The inexorable push for automated production," *Prod. Eng.*, vol. 4, pp. 45–46, Jan. 1977.
[32] M. Song and D. Chen, "A comparison of three heuristic optimization algorithms for solving the multi-objective land allocation (MOLA) problem," *Ann. GIS*, vol. 24, no. 1, pp. 19–31, Jan. 2018, doi: 10.1080/19475683.2018.1424736.
[33] T. K. Ghosh, S. Das, S. Barman, and R. Goswami, "A comparison between genetic algorithm and cuckoo search algorithm to minimize the makespan for grid job scheduling," *Adv. Intell. Syst. Comput.*, vol. 509, pp. 141–147, Nov. 2017, doi: 10.1007/978-981-10-2525-9_14.
[34] S. Dauzère-Pérès, W. Roux, and J. B. Lasserre, "Multi-resource shop scheduling with resource flexibility," *Eur. J. Oper. Res.*, vol. 107, no. 2, pp. 289–305, 1998.
[35] A. I Shabaka and H. A. Elmaraghy, "Generation of machine igurations based on product features," *Int. J. Comput. Integr. Manuf.*, vol. 20, no. 4, pp. 355–369, 2007.
[36] T. Zahid and A. A. Baqai, "Multi-criteria optimization of process plans for Reconfigirable manufacturing systems: An evolutionary approach," in *Proc. Adv. Manuf.*, Nov. 2013, pp. 1–8.
[37] A. Azab and H. A. ElMaraghy, "Mathematical modeling for reconfigurable process planning," *CIRP Ann.*, vol. 56, no. 1, pp. 467–472, 2007.
[38] D. Laha and J. N. D. Gupta, "An improved cuckoo search algorithm for scheduling jobs on identical parallel machines," *Comput. Ind. Eng.*, vol. 126, pp. 348–360, Dec. 2018, doi: 10.1016/j.cie.2018.09.016.
[39] H. M. Pandey, "Performance evaluation of selection methods of genetic algorithm and network security concerns," *Procedia Comput. Sci.*, vol. 78, pp. 13–18, Jan. 2016.
[40] D. Thierens and D. Goldberg, "Elitist recombination: An integrated selection recombination GA," in *Proc. 1st IEEE Conf. Evol. Comput.*, Jun. 1994, pp. 508–512.
[41] S. E. Carlson, "A general method for handling constraints in GA," Univ. Virgina, Charlottesville, VA, USA, Tech. Rep., 1999, pp. 663–667.
[42] A. B. Mohamad, A. M. Zain, N. E. N. Bazin, and A. Udin, "Cuckoo search algorithm for optimization problems—A literature review and its applications," *Appl. Mech. Mater.*, vol. 421, pp. 502–506, May 2013.
[43] X. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput.*, vol. 1, Oct. 2009, pp. 210–214.
[44] T. Tran, T. T. Nguyen, and H. L. Nguyen, "Global optimization using Lévy flight," in *Proc. ICT*, 2004, pp. 1–12.
[45] X. S. Yang, *Nature Inspired Metaheuristic Algorithms*. U.K.: Luniver Press, 2008.
[46] A. Noghrehabadi, M. Ghalambaz, M. Ghalambaz, and A. Vosough, "A hybrid power series–Cuckoo search optimization algorithm to electrostatic deflection of micro fixed-fixed actuators," *Int. J. Multidisciplinary Sci. Eng.*, vol. 2, no. 4, pp. 22–26, 2011.
[47] S. Walton, O. Hassan, K. Morgan, and M. R. Brown, "Modified cuckoo search: A new gradient free optimisation algorithm," *Chaos, Solitons Fractals*, vol. 44, no. 9, pp. 710–718, Sep. 2011.
[48] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *Int. J. Adv. Manuf. Technol.*, vol. 64, nos. 5–8, pp. 951–959, Feb. 2013.
[49] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. Math. Model. Number Optim.*, vol. 1, no, 4, pp. 330–343, 2010.
[50] M. Sipper, W. Fu, K. Ahuja, and J. H. Moore, "Investigating the parameter space of evolutionary algorithms," *BioData Mining*, vol. 11, no. 1, pp. 1–14, Dec. 2018, doi: 10.1186/s13040-018-0164-x.
[51] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemorary Phys.*, vol. 46, no. 5, pp. 323–351, 2004.
[52] C. Kumar and S. Deb, "Generation of optimal sequence of machining operations in setup planning by genetic algorithms," *J. Adv. Manuf. Syst.*, vol. 11, pp. 67–80, Jul. 2012.

IQRA SADAF KHAN received the B.S. degree (Hons.) in business administration from COMSATS University, Pakistan, in 2011, the M.S. degree in management science from RIPHAH International Islamabad, Pakistan, in 2013, and the master's degree in education and globalization from the University of Oulu, Oulu, Finland, in 2017, where she is currently pursuing the Ph.D. degree in industrial engineering and management with the Graduate School. Her research interests include industry 4.0, sustainable development, and innovation ecosystems.

USMAN GHAFOOR received the B.E. degree in mechatronics engineering from Air University Islamabad, in 2011, the M.S. degree in industrial and manufacturing engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2016, and the Ph.D. degree in mechanical engineering from Pusan National University, Busan, Republic of Korea, in 2021. His research interests include robotics, lean manufacturing, and brain–machine interface. He served as a Reviewer for IEEE Access.

TAIBA ZAHID received the Ph.D. degree from the Chairs of Logistics, Department of Mechanical Engineering, TU Dresden, Germany, in 2017. Since then, she has been supervising various research projects. Her research interests include operations research, supply chain management, smart logistics, and manufacturing systems design in industry 4.0.