SELECTION OF OPTIMAL SEQUENCE OF MANUFACTURING OPERATIONS IN CAPP

Lakshmi Chaitanya Maddila, Avinash Gudimetla, Sunil Raj Musinada,
Venkata Lakshmi Medisetti
Department of Mechanical Engineering,
Pragati Engineering College, Surampalem, AP, India

ABSTRACT
Computer aided processes planning (CAPP) is an important interface between Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) in the Computer Integrated Manufacturing (CIM) environment. In CAPP systems, operation sequencing is apprehensive with the assortment of machining operations in steps that can produce each form feature of the part by agreeable appropriate scientific constraints specified in the part drawing. The present work involves the application of Differential Evolution (DE), a global search technique for a quick identification of optimal or near optimal operation sequences in a dynamic planning environment. At first, the part is represented as an assembly of form features, with details of geometric specifications, tolerance and surface finish requirements and next, feasibility constraints are considered among various machining operations to find the optimal or near optimal sequences within the minimum possible time and hence this can be used by the process planner to generate alternative feasible sequences for the prevailing operating environment.

KEYWORDS: *Computer Aided Process Planning (CAPP), Constraints, Differential evolution, operation sequence rating index (OSRI)*

I. INTRODUCTION

Manufacturing industries have been significantly influenced by the changes in market demands over the years his document is template. With advances in information and electronic communication technology, industries have also shown a trend of interest towards making use of computers in each phase of manufacturing process. In search of a computerized system that leads to faster product development, higher productivity, greater flexibility, lower cost and better quality products, the concept of Computer integrated system (CIM) has emerged in the manufacturing vocabulary. In this context, the integration of Design and Manufacturing operations is made possible through Computer aided process planning (CAPP). So, CAPP forms an important interface between CAD (Computer aided design) and CAM (Computer aided manufacturing) in CIM environment.

CAPP makes use of a computer system to automate the decision-making tasks of process planning dealing with the translation of a set of design specifications into a set of technologically feasible instructions describing how to manufacture a piece part or assembly. This important function encompasses many different tasks including interpretation of design data, selection and sequencing of processes to manufacture the component, selection of machine tools and cutting tools, determination of cutting parameters, choice of jigs and fixtures and calculation of machining times and costs. Of these tasks, sequencing of the defined operations is a complex one due to the need of taking into account several types of constraints and the size of the resulting solution space. CAPP techniques are developed to overcome some of the problems occurring in manual process planning, such as long turnaround times, inconsistent routings or tooling, non-uniqueness in cost and labor requirements and scarcity of skilled process planners.

A review of about 206 references and an extensive list of CAPP systems developed until 1988 can be found in a survey article by Alting and Zhang [1]. The use of artificial intelligence technique can also

be found in the development of CAPP systems and over 50 knowledge-based systems developed until 1995 have been reviewed by Kiritis [2]. Subsequently, Cay and Chasspis [3] presented a survey of more than 170 references and brought out the research issues in the area of CAPP. Of the tasks involved within process planning, operation selection and sequencing present two major tasks. The selection of an operation is based on the form-feature geometry and its technological requirements and mapping these specifications to the appropriate operation or series of operations. Operation sequencing is one of the most complex tasks, exhibiting combinatorial nature. As the operations sequencing problem involves various interdependent constraints, it is very difficult to formulate and solve the sequencing problem using integer programming and dynamic programming methods alone. Application of graph theory for operation sequencing has been dealt by Sundaram and [4] Prabhu et al. [5]. In the latter work, the elimination of infeasible operation sequences and the use of tree structure for enumerating all the paths for weeding out the infeasible sequences have been reported. Rho et al. [6] have used a procedence matrix approach after analyzing the technological constraints and their optimization of the sequence of operations is based on the criteria of minimum cutting-tool-change and tool-travel times. The possibility of alternative sequences with the available resources is not discussed in their wok. Irani et al. [7] have used Hamiltonian path (HP) analogy for process planning problem based on the precedence graph and operation cost matrix. The need for using heuristic approaches for randomly generating alternative sequences is highlighted in their research work.

Vancza and Markus [8] have applied a genetic algorithm (GA) in which each string is represented by elements corresponding to feature states that are produced by machining operations. It was reported that the use of crossover and mutation operators resulted in about 10% infeasible sequences as the operators violated some of the precedence constraints. Hip-Hoi and Dutta [9] have used a new coding strategy in GA for sequencing of parallel machining operations, where combinations of interacting work-holding and tool-holding devices are used. Their coding methodology allows generation of only valid operation strings and the strings can be obtained with GA operators like crossover, mutation and reproduction. The drawback with the above methodology is that operation features with multiple parents are not considered, i.e., a feature cannot be produced unless, at least, one of its preceding features is machined (OR relation) or all of its preceding features are machined (AND relation). For large-size problems as encountered in process planning the use of GA with proper generation and pruning schemes can help in arriving at the optimal/near-optimal plans.

In order to handle conveniently the problems of process planning, it is prudent to divide it into two stages. In the present work, the CAPP is divided into preliminary planning, and secondary and detailed planning. In the preliminary stage, sequencing of form features is carried out considering their geometry and their independence. During secondary and the detailed level of planning, relevant manufacturing information such as operations, machine tools, cutting tools, operation sequence and fixturing plan for the part are determined.

The present work presents the details of a method for preliminary planning. As the sequences can be obtained quickly, the proposed approach can be used effectively by a process planner to generate alternative feasible sequences for a given operating environment.

The paper is organized as follows: Section 2 of the manuscript explains about the proposed method for quick identification of optimal or near optimal operation sequences. The methodology carried out carried out using differential evolution is presented in Section 3. The results and discussions on the methodology proposed are discussed in Section 4, while Section 5 presents the concluding remarks and scope for further work.

II. DIFFERENTIAL EVOLUTION

Recently, evolutionary algorithms EAs (encompassing genetic algorithms, evolution strategies, and genetic programming) have received a lot of attention for solving a wide range of non-linear optimization problems. Among EAs, GA has got a lot of popularity in the recent past. Although several GA versions have been developed, they are not efficient when convergence speed is taken into consideration.

DE is an exceptionally simple, fast and robust evolutionary computation method proposed by Storn and Price [10] for solving continuous space optimization problems and is more likely to find the true

optimum of a problem. This method has been applied to several diverse fields: digital filter design [11], neural network learning [12], fuzzy-decision-making problems of fuel ethanol production [13], multi-sensor fusion [14], dynamic optimization of continuous polymer reactor [15], batch fermentation process [16], optimization of heat transfer parameters in trickle-bed reactors [17] etc.
Similar to other EAs, DE works with a population (N) of candidate solutions. N is one of the control parameters of the method, which does not change during the optimization process. Initial population is generated randomly and the algorithm improves the population iteratively. This population is successively improved by the mutation, crossover and selection operators. The flowchart of DE is shown in Figure.1.

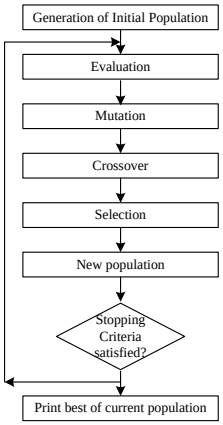

Figure 1: Flow chart of Differential evolution

Currently there are several versions of DE which are classified based on the notation *DE/x/y/z* where *x* specifies the vector to be mutated, *y* is the number of difference vectors considered for mutation of *x*, and *z* stands for the crossover scheme. *x* can be either 'rand' (randomly chosen vector) or 'best' (best vector of current population). For mutation, either single or two vector differences can be used and accordingly *y* becomes either 1 or 2. *z* can be either 'bin' (binomial) or 'exp' (exponential) depending upon the type of crossover scheme used. Price and Storn [18] have suggested ten different working strategies and some guidelines for applying these strategies to a specific problem.
The variant *DE/rand/1/bin* expressed as $X_{r1,\,g}$+ F. ($X_{r2,\,g}$-$X_{r3,\,g}$) strategy has been used in the present work. Main operators are explained below:

2.1. Mutation

All solution vectors (target vectors) of current generation ($X_{i,g}$, i = 1,2,3,….,N) are tested cyclically against newly generated vectors called trial vectors ($u_{i,\,g+1}$) . First, in this phase, DE generates a mutant vector ($v_{i,\,g+1}$), by adding a weighted difference of two population vectors to a third vector and is generated according to the following equation:

$$v_{i,g+1} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) \quad (1)$$

Where F>0 is a scaling factor, which controls the magnitude of the differential variation of $(X_{r2,g} - X_{r3,g})$. The vectors X and $X_{r3,g}$ are to be randomly selected and to be different from the current target vector.

2.2.Crossover

The mutant vector and the target vector are then subjected to crossover to generate the trial vector $(u_{i,g+1})$ based on the following equation:

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1} & if\ m_j \le c_r \\ x_{ji,g} & otherwise \end{cases} \quad (2)$$

Where j=1, 2…D. Here *D* represents the number of dimensions of a vector. $C_r \in [0, 1]$ is the crossover constant which represents the probability of trial vector that inherits parameter values from the mutant vector. Crossover is introduced in the algorithm to control the amount of diversity of mutant vectors.

2.3.Selection

The trial vector produced by the crossover operator is compared with the target vector to determine the member for the next generation. If a trial vector produces a smaller function value it is copied to next generation otherwise target vector is passed into next generation.

$$If\ f(u_{i,g+1}) \leq f(x_{i,g}), set\ x_{i,g+1} = u_{i,g+1}; otherwise\ x_{i,g+1} = x_{i,g} \quad (3)$$

The above procedure continues until a stopping criterion is met. This criterion can be either to have the current best objective function value smaller than a specified value or the number of generations is equal to a predetermined maximum value.

2.4.Choice of Control Parameters

Some guidelines are available to choose the control parameters *N*, *F*, and C_r [15]. Normally, *N* is to be between 5 to 10 times the numbers of parameters in solution vector. As for *F*, it is to be in the range of 0.4 to 1.0. *F*=0.8 is a good initial choice and it can be increased if the population converges prematurely. On the other hand, a good value for C_r is 0.5; however to speed up convergence a greater value can be attempted.

2.5.Boundary Constraints

In boundary-constrained problems, the parameter values need to be with in the allowable ranges. Equation (1) may produce the parameter values that may lie outside the allowable ranges. Therefore, in case of boundary constraint violation, the following method (5) can be used to adjust the parameter values.

$$u_{ji,g+1} = \begin{cases} \left(x_{ji,g} + x(_j^{(L)}\right)/2, & if\ u_{ji,g+1} < x_j^{(L)} \\ \left(x_{ji,g} + x(_j^{(U)}\right)/2, & if\ u_{ji,g+1} < x_j^{(U)} \\ u_{ji,g+1} & otherwise \end{cases} \quad (4)$$

Although DE is used widely for continuous problems, yet it can also be used for solving NP-hard combinatorial problems. To solve the present combinatorial problem, the smallest position value (SPV) rule suggested by Tasgetiren et al. [19, 20] is adopted. This is used in the present work to convert a continuous position vector into a permutation problem.

III. METHODOLOGY

The part drawing shown in Figure. 2 is taken as an example. Dimensional and geometrical tolerances along with other technical specifications are shown on the part drawing. The features involved in manufacturing the part are listed in Table.1. Each feature is coded with a numerical letter.

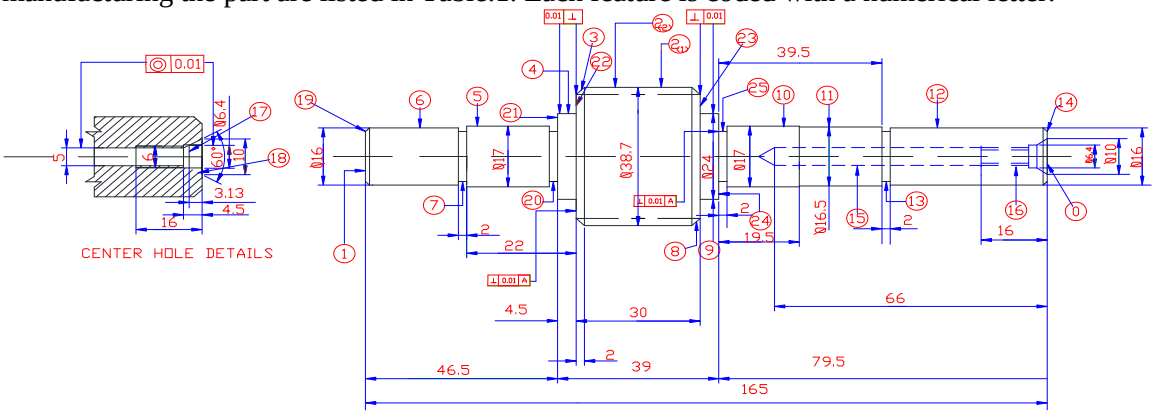

Figure.2 Case Study

Table 2 exhibits the constraints obtained for the part. The objective is to determine the sequence of machining operations that corresponds to the minimum set-up changeovers and minimum tool changeovers.

Table 1: Details of Features	
Feature Number	Feature Description
0	End Face
1	End Face
$2_{(1)}$	External Cylinder
$2_{(2)}$	External Cylinder
3	Chamfer
4	External Cylinder
5	External Cylinder
6	External Cylinder
7	External groove
8	Chamfer
9	External Cylinder
10	External Cylinder
11	External Cylinder
12	External Cylinder
13	External groove
14	Chamfer
15	Internal Hole
16	Internal Tapping
17	Counter boring
18	Counter sinking
19	Chamfer
20	External groove
21	End Face
22	End Face
23	End Face
24	End Face
25	External groove

Table 3: Solution representation of target vector		
Dimension	$X_{ij,g}$	Operation $P_{ij,g}$
0	0.773	5
1	0.702	23
$2_{(1)}$	0.278	17
$2_{(2)}$	0.911	6
3	0.962	20
4	0.863	15
5	0.058	$2_{(1)}$
6	0.184	11
7	0.792	8
8	0.325	22
9	0.533	21
10	0.619	12
11	0.296	25
12	0.402	18
13	0.946	19
14	0.854	9
15	0.254	24
16	0.667	10
17	0.124	16
18	0.462	1
19	0.486	0
20	0.213	7
21	0.392	14
22	0.336	4
23	0.072	$2_{(2)}$
24	0.547	13
25	0.426	3

Table 2: Constraints	
Locating Constraints	$1\rightarrow2_{(1)}$, $1\rightarrow3$, $1\rightarrow4$, $1\rightarrow5$, $1\rightarrow6$, $1\rightarrow7$, $1\rightarrow19$, $1\rightarrow20$, $1\rightarrow21$, $1\rightarrow22$, $0\rightarrow2_{(2)}$, $0\rightarrow8$, $0\rightarrow9$, $0\rightarrow10$, $0\rightarrow11$, $0\rightarrow12$, $0\rightarrow13$, $0\rightarrow14$, $0\rightarrow15$, $0\rightarrow16$, $0\rightarrow17$, $0\rightarrow18$, $0\rightarrow23$, $0\rightarrow24$, $0\rightarrow25$
Accessibility Constraints	$2_{(1)}\rightarrow3$, $4\rightarrow22$, $2_{(1)}\rightarrow22$,$2_{(1)}\rightarrow4$,$5\rightarrow20$,$4\rightarrow5$, $6\rightarrow7$, $4\rightarrow6$, $5\rightarrow6$, $2_{(2)}\rightarrow8$, $2_{(2)}\rightarrow9$, $6\rightarrow19$, $2_{(2)}\rightarrow23$, $23\rightarrow8$, $9\rightarrow24$, $9\rightarrow25$, $9\rightarrow23$, $10\rightarrow25$, $9\rightarrow10$, $10\rightarrow11$, $9\rightarrow13$, $11\rightarrow12$, $11\rightarrow13$, $12\rightarrow13$, $12\rightarrow14$, $15\rightarrow16$, $15\rightarrow17$, $15\rightarrow18$, $4\rightarrow21$, $25\rightarrow24$, $20\rightarrow21$, $17\rightarrow18$
Geometric tolerance constraints	$9\rightarrow23$, $15\rightarrow6$, $4\rightarrow22$
Non-destructive Constraints	$17\rightarrow16$

3.1.Representation

DE works with a population (N) of candidate solutions. N is one of the control parameters of the method, which does not change during optimization process. Initial population is generated and algorithm improves the population iteratively. This population is successively improved by the mutation, crossover and selection operators.Table.3 illustrates the solution representation of target vector $X_{ij,g}$. According to the SPV rule, the smallest parameter value is $X_{ij,g}$= 0.058, so the dimension j=5 is assigned to be the first operation P_{i1} = 5 .The second smallest parameter value is $X_{ij,g}$ = 0.072, so the dimension j =23 is assigned to be the second operation P_{i2} = 23 and so on. In other words, dimensions are sorted according to the SPV rule, i.e., according to the parameter values $X_{ij,g}$ to construct the permutation $X_{ij,g}$.

3.2.Objective Function

An attempt has been made to determine optimal/near optimal operation sequence using high quality approximation algorithm known as Differential evaluation (DE). Sequencing of operations is affected by parameters related to geometric complexities and non-geometric complexities. Parameters pertaining to geometric complexities include tool compatibility, feature symmetry, feature accessibility, feature orientation etc., whereas non-geometric parameters include dimensional tolerance geometric tolerance, location tolerance and surface finish. The effects of parameters generating complexities are broadly included in the form of two indices: setup changeover index and motion continuity index. By combining these indices and by assigning suitable weights, an objective function has been formulated to evaluate the optimality of operation sequence. The objective function has been termed as operation sequence rating index (OSRI).The sequence which offers maximum value of OSRI will be the optimal /near optimal sequence. This index encompasses the following operation sequencing objectives i.e., Minimize the setup changeover and Maximize motion continuity. These objectives have been represented in the term of setup changeover index (SCI), motion continuity index (MCI).

A set of features that are to be machined by holding on a given datum /reference feature (in a single setup) are shown in Table 4. Using production rules which group similar type features in each setup, feature adjacency templates are generated as shown in Table 5.

Table 4: Details of holding the features in same set

Set 1	1,$2_{(1)}$,3,4,5,6,7,19,20,21,22
Set 2	0,$2_{(2)}$,8,9,10,11,12,13,14,15,16,17,18,23,24,25

Table 5: Continuity templates

Template 1	1→$2_{(1)}$→4→5→6→22→21→3→19
Template 2	0→$2_{(2)}$→9→10→11→12→23→24→14→8

The value of the objective function denotes the strength of each string based on the aforementioned optimality criteria. It is calculated for each sequence by the scoring method as discussed below:

Maximize OSRI [Qi] =

$$\frac{w1 * SCI[Qi] + w2 * MCI[Qi]^{-}}{w1 + w2} \quad (5)$$

Where SCI [Qi] is the setup change over index=

$$\frac{\text{Maximum number of set up change over} - \text{number of set up change over required in Qi}}{\text{Maximum number of set up change over} - \text{minimum number of set up change over}} \quad (6)$$

MCI [Qi] is the motion continuity index = $\frac{\sum_{i=1}(B_i-1)}{\sum_{i=1}(A_i-1)}$, i is the number of continuity template, Ai is the number of features in the continuity template i, B_i is the features in sequence which are compatible with the continuity template i

w1, w2 depends on user preference and are taken to be 0.7 and 0.3 respectively

3.3.Initial Population

The initial population is randomly generated. Since the random number generator only generates floating-point numbers between [0,1], SPV rule as explained above is used to transform the values into discrete numbers within the lower bound 0 and upper bound 23 The discrete values are passed to the array, but all the values are checked against the values already in the array. If a repeated value is detected, that value is rejected. The complete initial population is shown in Table 6. The objective function values for the initial population are calculated and are shown in the Table 7.

Table 6: Discrete initial population array

1	2	3	4	5
5	8	10	21	13
23	14	8	5	18
17	5	7	19	20
6	9	12	24	11
20	6	5	20	15
15	0	9	16	8
$2_{(1)}$	10	14	11	$2_{(2)}$

Table 8: Initial floating-point population

	1	2	3	4	5
0	0.773	0.199	0.627	0.462	0.305
1	0.702	0.328	0.703	0.653	0.463
$2_{(1)}$	0.278	0.919	0.37	0.446	0.432
$2_{(2)}$	0.911	0.271	0.512	0.531	0.295
3	0.962	0.308	0.648	0.932	0.513
4	0.863	0.456	0.679	0.699	0.321
5	0.058	0.105	0.204	0.108	0.568

11	$2_{(2)}$	13	14	0
8	3	11	23	4
22	1	$2_{(1)}$	13	25
21	12	18	10	10
12	13	16	$2_{(1)}$	14
25	24	25	0	19
18	4	20	$2_{(2)}$	23
19	7	$2_{(2)}$	12	$2_{(1)}$
9	11	24	22	1
24	19	23	15	3
10	25	22	1	6
16	18	21	4	9
1	21	0	6	5
0	23	3	17	12
7	16	4	9	16
14	15	1	7	22
4	17	6	25	24
$2_{(2)}$	$2_{(1)}$	15	18	17
13	20	17	3	7
3	22	19	8	21

6	0.184	0.186	0.781	0.719	0.535
7	0.792	0.495	0.144	0.811	0.923
8	0.325	0.04	0.093	0.985	0.268
9	0.533	0.166	0.233	0.779	0.544
10	0.619	0.232	0.052	0.405	0.366
11	0.296	0.513	0.301	0.324	0.163
12	0.402	0.392	0.184	0.542	0.613
13	0.946	0.416	0.281	0.386	0.054
14	0.854	0.071	0.276	0.339	0.409
15	0.254	0.854	0.821	0.637	0.221
16	0.667	0.786	0.426	0.285	0.676
17	0.124	0.873	0.938	0.746	0.874
18	0.462	0.616	0.406	0.891	0.071
19	0.486	0.559	0.978	0.154	0.418
20	0.213	948	0.487	0.235	0.136
21	0.392	0.632	0.604	0.033	0.966
22	0.336	0.983	0.588	0.602	0.706
23	0.072	0.762	0.554	0.357	0.423
24	0.547	0.443	0.526	0.185	0.789
25	0.426	0.569	0.446	0.835	0.36

3.4.Mutation

For the mutation scheme, the initial floating-point array is shown in Table 8. The *mutation scheme* acts upon the population at this point. The first vector member is mutated in the second vector i = 2 as shown in Table 9. Thus the first value to be mutated is 0.199. Three random numbers *r* 1, *r* 2 and *r* 3 are generated. These numbers are within the population range [1, 5]. The numbers generated are *r1*= 5, *r 2* = 4 and *r3* = 1. These are listed in Table 10. Since the value pointed to by *r1* is 7, the value is 0.304.Secondly the value pointed to by *r 2* is 6, and the value is 0.462. Finally the value pointed to by *r3* is 1, the value is 0.733. Then the target vector is mutated using equation (1) as 0.304+ 0.8 x (0.462-0.773) = 0.055

3.5.Crossover

A random number *rnd* is generated between the bounds [0, 1], and this is checked against the value for CR, 0.5. The table showing the trial vectors randomly generated is shown in the Table 12 If the random generated number *rnd* is less than CR, as in the case for this array, the value in the object array of trail vector will be same as mutant vector value otherwise trail vector value will be replaced by target vector value. The trial array is then transformed back into discrete values. This is achieved through the SPV rule as shown in Table 13. Other conversions are similarly made, leading to the trial vector sequence {9,7,10,13,$2_{(2)}$,5,3,12,24,14,11,4,1,0,16,18,6,21,15,25, 2 $_{(1)}$,19,8,20,22,23,17}.

Table 9: Vector for mutation operation

D	i=2
0	0.199
1	0.328
2^a	0.919
2^b	0.271
3	0.308
4	0.456
5	0.105
6	0.186
7	0.495
8	0.04

Table 10: Randomly selected variables for mutated array i=2

i=5	i=4	i=1
0.305	0.462	0.773
0.463	0.653	0.702
0.432	0.446	0.278
0.295	0.531	0.911
0.513	0.932	0.962
0.321	0.699	0.863
0.568	0.108	0.058
0.535	0.719	0.184
0.923	0.811	0.792
0.268	0.985	0.325

Table 11: Mutated array for i =2

D	i
0	0.055
1	0.423
$2_{(1)}$	0.566
$2_{(2)}$	-0.009
3	0.489
4	0.189
5	0.626
6	0.963
7	0.938
8	0.796

9	0.166		0.544	0.779	0.533	9	0.75
10	0.232		0.366	0.405	0.619	10	0.194
11	0.513		0.163	0.324	0.296	11	0.185
12	0.392		0.613	0.542	0.402	12	0.725
13	0.416		0.054	0.386	0.946	13	-0.394
14	0.071		0.409	0.339	0.854	14	-0.003
15	0.854		0.221	0.637	0.254	15	0.527
16	0.786		0.676	0.285	0.667	16	0.374
17	0.873		0.874	0.746	0.124	17	1.371
18	0.616		0.071	0.891	0.462	18	0.414
19	0.559		0.418	0.154	0.486	19	0.152
20	948		0.136	0.235	0.213	20	0.153
21	0.632		0.966	0.033	0.392	21	0.678
22	0.983		0.706	0.602	0.336	22	0.918
23	0.762		0.423	0.357	0.072	23	0.651
24	0.443		0.789	0.185	0.547	24	0.499
25	0.569		0.36	0.835	0.426	25	0.687

3.6.Selection

Selection is the mechanism for determining whether the trial vector replaces the current target vector or not. The objective function of the trial-vector {9, 7, 10, 13, $2_{(2)}$, 5, 3, 12, 24, 14, 11, 4, 1, 0, 16, 18, 6, 21, 15, 25, $2_{(21}$, 19, 8, 20, 22, 23, 17} is 0.26. Since this value is less than the target vector's objective function value of 0.3, trial vector is copied to the next generation. Table 14 lists the objective function values of trial vectors. The above procedure is repeated for the remaining vectors of the initial population.

3.7.Parameter Setting

A large value of F increases the probability of escaping local optimum. Therefore, it is set to be 0.8 in the present case study. Although a large value of C_r speeds up convergence but it makes the population converge prematurely. Therefore a reasonable choice of C_r equal to 0.5 is used. The algorithm is run for a maximum of 50 generations.

IV. RESULTS AND DISCUSSION

The proposed technique is coded in JAVA and executed on a Pentium IV with 2.8 Ghz processor. In the context of process planning, the most important aspect is to minimize the number of setups. While minimizing the number of setup, it is observed that the fixtures of a part in a setup, is an expensive process and possibly needs considerable human involvement. Every setup change results in the delay in the production plan and of course, lead to the loss of accuracy. Because of the strong interconnection of the above objectives of process planning, it is essential that they may be considered simultaneously. For the case study, the optimal sequence obtained is 0-15-17-18-$2_{(2)}$-9-23-8-10-11-12-16-14-13-25-24-1-$2_{(1)}$-4-5-3-6-19-20-22-21-7 with the total cost of 0.858 units. It is possible that sometimes there may be alternative sequences of manufacturing operations for a product. The alternative sequences correspond to the same least total cost. Since the computation time required to generate the optimal solution is very low, this algorithm can be run a few times to generate the alternative operation sequences.

Table 12: Trial vectors				
1	2	3	4	5
0.912	0.175	0.326	0.66	0.83
0.994	0.029	0.499	0.185	0.099
0.864	0.684	0.487	0.27	0.288
0.887	0.741	0.216	0.232	0.629
0.403	0.054	0.972	0.357	0.475
0.275	0.256	0.365	0.214	0.941
0.342	0.905	0.404	0.934	0.662
0.878	0.819	0.312	0.493	0.247
0.204	0.856	0.609	0.341	0.693
0.709	0.701	0.022	0.843	0.955
0.326	0.126	0.525	0.408	0.981
0.514	0.907	0.992	0.39	0.565
0.841	0.861	0.835	0.975	0.016
0.344	0.817	0.227	0.441	0.388
0.924	0.806	0.24	0.163	0.924
0.357	0.549	0.54	0.765	0.268
0.221	0.648	0.436	0.61	0.18
0.097	0.405	0.033	0.045	0.159
0.241	0.194	0.78	0.054	0.668
0.562	0.475	0.962	0.285	0.374
0.945	0.941	0.138	0.884	0.612
0.746	0.665	0.177	0.353	0.088
0.158	0.314	0.94	0.133	0.031
0.65	0.23	0.64	0.06	0.732
0.965	0.291	0.732	0.391	0.838
0.201	0.698	0.858	0.995	0.274
0.625	0.057	0.74	0.905	0.311

Table 13 Discrete final population array				
1	2	3	4	5
23	8	13	9	16
15	0	0	7	20
6	14	$2_{(2)}$	10	$2_{(1)}$
20	5	10	13	21
$2_{(1)}$	6	7	$2_{(2)}$	13
11	4	5	5	11
8	10	19	3	24
22	$2_{(2)}$	9	12	8
4	16	14	24	$2_{(2)}$
25	12	11	14	0
14	13	1	11	4
18	18	4	4	10
19	1	16	1	12
5	24	18	0	20
21	3	25	16	23
24	7	24	18	14
16	11	12	6	9
10	19	23	21	5
1	23	22	15	15
17	21	21	25	22
3	25	3	$2_{(1)}$	18
0	9	6	19	17
12	15	8	8	25
$2_{(2)}$	22	20	20	7
13	$2_{(1)}$	17	22	1
9	20	$2_{(1)}$	23	6
7	17	15	17	3

Table 14: Objective function values of Trial vectors				
i				NP
1	2	3	4	5
0.36	0.329	0.487	0.26	0.33

V. CONCLUSIONS AND FUTURE WORK

For a computer-aided process planning system (CAPP) to handle complex parts comprised of a large number of interacting features, an efficient means is needed for exploring and reducing the size of the search space of valid operation sequences. Thus in this, an attempt has been made to address the issues to an extent and convert the problem into a multi-objective optimization problem by associating adjustable weights with individual objectives as Minimization of setup changeovers and Maximization of motion continuity

Thus, the proposed methodology of operation sequencing based on the determination of OSRI is flexible enough to handle a variety of situations and complexities. As has been pointed out earlier, that the presence of several constraints related to CASP and CAPP greatly complicate the sequencing tasks and thereby enlarge the size of search space in determining the optimal or near optimal sequence. The stimulated annealing based algorithm discussed in this paper, simplifies the selection of the optimal sequence for a part having a fairly large number of features compared to other optimization tools such as integer programming, dynamic programming, branch and bound method etc.the proposed simulated annealing algorithm considerably reduces the computational time and search space to reach to the optimality of the solution. For the effective performance of proposed simulated annealing algorithm, a modified shifting scheme has been proposed which generates feasible operation sequence. This avoids the repetitive procedure of checking the feasibility of the sequence generated. The computation time and search space have been reduced significantly. As far

as control parameters are concerned, the same control settings have been used for each problem except the initial temperature setting which is based on the cost of the initial sequence. Such initial temperature settings generally perform better than arbitrary settings. The proposed methodology to determine the optimal operation sequences is recommended to meet the requirements of integrated CAPP system.

Differential Evolution (DE) algorithm proposed in this paper is a new heuristic approach mainly having three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using few control parameters. However, the proposed methodology can be applied to other part drawings that can produce each form feature of the part by agreeable appropriate scientific constraints specified.

REFERENCES

[1] Leo Alting & Hong-Chao Zhang, (1989) "Computer aided process planning: the state-of-the art survey", International *Journal of Production Research*, Vol. 27, No. 4, pp553-585.

[2] Dimitris Kiritis, (1985) "Review of knowledge-based expert systems for process planning, methods and problems", *International Journal of Advanced Manufacturing*, Vol. 10, No. 4, pp240-262.

[3] F.Cay & C.Chasspis, (1997) "An IT view on perspectives of Computer aided process planning research", *Computers in Industry*, Vol. 34, No. 3, pp307-337.

[4] R. M. Sundaram, (1986) "Process planning and machining sequence", *Computers and Industrial Engineering*, Vol. 11, No. 4, pp184-188.

[5] P. Prabhu, S. S. Elhance, H.Wang & R.A. Wysk, (1990) "An operations network generator for computer aided process planning", *Journal of Manufacturing Systems*, Vol. 9, No. 4, pp283-291.

[6] H.M. Rho, R. Geelink, A. H. Van't Erve and H. J. J. Kals (1992) "An integrated cutting tool selection and operation sequencing method, *Annals of CIRP*, Vol. 41 No. 1, pp517-520.

[7] S.A. Irani, H.Y. Koo & S. Raman, (1995) "Feature-based operation sequence generation in CAPP", *International Journal of Production Research*, Vol. 33 No. 1, pp17-39.

[8] J. Vancza & A. Markus, (1991) "Genetic algorithms in process planning", *Computers in Industry*, Vol. 17, No. 2, pp181 -194.

[9] D.Hip-Hoi & D.Dutta, (1996) "A genetic algorithm application for sequencing operation in process planning for parallel machining, *IIE Transactions*, Vol. 28, No. 2, pp55-68.

[10] R. Storn & K. Price, (1997) "Differential evolution – a simple evolution strategy for fast optimization , *Dr. Dobb's J*, Vol. 22, No. 4, pp18-24.

[11] R. Storn, (1995) "Differential evolution design of an IIR-filter with requirements for magnitude and group delay", TR-95-018, ICSI.

[12] T. Masters & W. Land, (1997) "A new training algorithm for the general regression neural network, *IEEE Int. conf. on Systems, Man and Cybernetics, Computational cybernetics and simulation*, 31990-1994

[13] F.S. Wang, C.H. Jing & G.T. Tsao, (1998) "Fuzzy-decision-making problems of fuel ethanol production, using genetically engineered yeast", Ind. *Engng. Chem. Res*, Vol. 37. No. 8, pp3434-3443.

[14] R. Joshi & A. C. Sanderson, (1999) "Minimal representation multi-sensor fusion using differential evolution", *IEEE Tran. on Systems, Man and Cybernetics, Part A* , Vol. 29, No. 1, pp 63-76.

[15] M. H. Lee, C. Han & K.S. Chang, (1999) "Dynamic optimization of a continuous polymer reactor using a modified differential evolution", *Ind. Engg. Chem. Res*, Vol. 38, No. 12, pp4825-4831.

[16] J.P.Chiou & F.S. Wang, (1999) "Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process, *Comput.Chem. Engng*, Vol. 23, No. 9, pp1277-1291.

[17] B.V.Babu & K.K.N.Sastry, (1999) "Estimation of heat transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation", Comp. Chem. Engg, Vol. 23, pp327 – 339.

[18] K. Price & R. Storn, Differential evolution home page (web site of Price and Storn) http://www.icsi.berkely.edu/~storn /code.html as at 2001

[19] M. F. Tasgetiren, M. Sevkli, Y-C. Liang & G. Gencyilmaz, (2004) "Particle swarm optimization algorithm for single-machine total weighted tardiness problem", Congress *on Evolutionary Computation, CEC2004*, Portland, Oregan, USA

[20] M. F. Tasgetiren, Y-C. Liang, M. Sevkli, G. Gencyilmaz, (2004) " Particle swarm optimization algorithm for permutation flowshop sequencing problem", *Fourth International Workshop on Ant Algorithms and Swarm Intelligence, ANTS2004*, Brussel, Belgium.

AUTHORS

Lakshmi Chaitanya Maddila received her Ph.D. degree in Mechanical Engineering in 2004 from the University College of Engineering, JNTU, Kakinada. Dr. Lakshmi Chaitanya research interests include Process Modelling and optimization, Manufacturing, Metal matrix composites, Nano composites. Currently, she is an Associate Professor at the Mechanical Engineering Department, Pragati Engineering College, Surampalem, Andhra Pradesh.

Avinash Gudimetla received M.Tech degree from GIET Engineering College, Rajahmundry, Andhra Pradesh, India in 2012. He has been working in teaching since 2007. He is now working as Assistant Professor in Department of Mechanical Engineering at Mechanical Engineering Department, Pragati Engineering College, Surampalem, Andhra Pradesh, India.

Sunil Raj Musinada received M.Tech degree from B.V.C Engineering College, Odalarevu, Andhra Pradesh, India in 2014. He has been working in teaching since 2010. He is now working as Assistant Professor in Department of Mechanical Engineering at Mechanical Engineering Department, Pragati Engineering College, Surampalem, Andhra Pradesh, India.

Venkata Lakshmi Medisetti received B.Tech degree from Pragati Engineering College, Surampalem, Andhra Pradesh, India in 2012. She is currently pursuing the M. Tech. (Mechanical Engineering) from Pragati Engineering College, Surampalem, Andhrapradesh, India