Hybrid Tabu-Sample Sort Simulated Annealing (SSA) with Fuzzy Logic Controller: CIM System Context

Subhash Wadhwa*

Professor, Department of Mechanical Engineering

Indian Institute of Technology, Hauz Khas, New Delhi-110016, India

(Research Areas: CIMS, GA, SA, Simulation, AI, Fuzzy Systems)

Anuj Prakash

Department of Mechanical Engineering,

Madan Mohan Malviya Engineering College, Gorakhpur- 273010, India

(Research Areas: CIMS, GA, Fuzzy Systems, Optimization)

Bibhushan

Department of Mechanical Engineering

Indian Institute of Technology, Hauz Khas, New Delhi-110016, India

(Research Areas: CIMS, GA, Dynamic SCM, Flexibility, Simulation)

*To whom correspondence should be addressed:

Email: swadhwa@mech.iitd.ernet.in

Abstract: In this research paper, process planning problem of a CIM system has been discussed, wherein minimization of cost of the finished product is considered as the main objective. For determining the cost of finished product, scrap cost, that is ignored by most of the previous researchers, has been considered along with other costs like raw material cost, processing cost etc. In the present environment of concurrent engineering, optimization of process planning is a very challenging problem. To solve this complex problem a hybrid type novel search algorithm, known as Tabu-Sample Sort Simulated Annealing (TSSA) has been proposed. The novelty of the proposed algorithm is that the features gleaned from both random search techniques have been incorporated in it. To update the swapping rate of TSSA, an adaptive controller, regulated by fuzzy rule base, has been also embedded. Furthermore, a modified transition probability based on Cauchy function has been investigated. To demonstrate the efficacy of the proposed Tabu-SSA with Fuzzy Logic Controller (FLC), a bench mark problem has been considered. Intensive computational experiments have also been performed on randomly generated datasets to reveal the supremacy of the proposed algorithm over other existing heuristics.

Keywords: Tabu-search, Sample Sort Simulated Annealing (SSA), Fuzzy Logic controller, Computer Aided Process Planning (CAPP)

Professor Subhash received his PhD from NUI, Ireland, while working on an ESPRIT project at CIM Research Unit, NUI, Galway.. He is currently Professor of Industrial Engineering, at the Indian Institute of Technology, Delhi. He is an active contributor to EC projects in the IT&C domain and has coordinated several EU workshops in India. He has over 100 research publications including those in renowned international journals such as IJFMS, IJPR, Computers & Industrial Engineering, International Journal of Technology Management and SIC. He has proposed and demonstrated the utility of new IT based architectures in Intelligent Systems such as SAMIN (ANN based proactive planning), DRIS (GA based integrated planning and scheduling in re-configurable Agile Systems) and DIS (Decision Information Synchronization) applications to Flexible Supply Chain and Enterprise Synchronization. He has been a Consultant/Contributor/National-Expert to many international bodies including EC, UNIDO, CW and APO (Tokyo). He is dedicated to the goal of bringing synergy between Academics, Industry and Research.

Mr Anuj Prakash is a young researcher who has been working in the CIM lab, IIT, Delhi. After his Bachelors degree he has done Masters in Computer Integrated Manufacturing Systems. He is aiming to do Phd at IIT, Delhi under Prof. S. Wadhwa. He is interested to carry out research in the domain of CIM systems involving multiple entity flow modelling. His research interests involve GA, SA, Fuzzy Systems applications to the CIM system problems.

Mr Bibhushan is working for his Phd degree at the CIM lab, IIT Delhi. He has done his Bachelors in Mechanical Engineering at DCE, New Delhi. He completed his Masters at IIT, Delhi and contributed to CRM domain. He is an active researcher with Prof. S. Wadhwa and is involved in developing virtual enterprise models. He has developed several dynamic control models in flexible supply chains and is researching the role of various control strategies to exploit supply chain flexibility CIM Systems

1. Introduction

One of the challenges of CIMS is to integrate effectively several resources with different capacities, capabilities and facilities. Effective operation of CIMS is not guaranteed because of some critical decision making issues like allocation of resources, planning, scheduling etc. There can be alternative operations on a variety of parts that can be done with alternative resources. This makes process planning one of the most intricate decision making problems for the CIM systems. Further its multi-objective nature adds more challenges. Computer Aided Process Planning (CAPP) can be very helpful in such a manufacturing scenario because it can generate efficient and cost effective process plans promptly. In this paper, process plan selection is chosen for analysis in CIM environment.

Process planning is the task of selecting the appropriate routing of manufacturing process(es) that a part need to follow in order to transform from raw material to the finished product. Hence, process plan works like a bridge between design and manufacturing process. It plays a vital and crucial role in design-to-manufacturing chain of processes. Therefore, it is essential to develop a process plan that has adequate knowledge of both upstream (design) and downstream (manufacturing) processes. In the present manufacturing paradigm, multiple machines, tools etc. are available to manufacture any product. Therefore, there may be multiple feasible processing routes consisting of several alternative machines, tools, fixtures, etc for the manufacturing of each part type. In other words, various kinds of flexibilities like machine flexibility, routing flexibility, etc. are available to the decision maker to arrive at the best schedule. Thus a synergy between flexibility, automation and integration is essential in CIM environment to fulfill system objectives (Wadhwa and Aggarwal 2000). Flexibility of such a system could be employed to significantly reduce the manufacturing lead times and consequently to reduce the costs (Wadhwa *et. al.* 2005). Flexibility also increases the number of decision points in the manufacturing system. These decision points could be utilized to further improve the system performance (Wadhwa and Browne 1989). In a highly flexible system like CIMS, the decision points are too many. This necessitates the need for an effective CAPP leading to best utilization of resources and minimization of operating costs. But the system flexibility should be judiciously chosen as it can turn out to be productive or counter-productive for the system (Chan *et. al.* 2006). In addition to this, the requirement of production volume and cost of processing have a decisive role in the selection of a process plan. In CIM systems, the role of CAPP is more critical as it impacts many integrated processes involving decision-information synchronization. Further, the resource costs are higher due to flexible automation and IT investments etc.

Generally a process plan has been optimized based on two criteria: cost of manufacturing and mean flow time. Some researchers like Palmer (1996), Husbands and Mill (1991) have worked on time aspects of alternative machines and tools during the optimization process. While, some others like Zhang *et al.* (1997), Ma et al. (2000) etc. have worked on cost optimization because cost is a major determinant of profitability. The researchers have mostly not taken into account the *scrap* or *rejection* cost. On the contrary, rejections at each machine are generally taken as guideline for developing a new process plan in a real shop floor environment. Therefore, by considering the scrap factor, this paper attempts to make a more realistic insight to the CAPP problem.

The concept of CAPP has been originated in 1960's (Niebel 1965) to reduce the lead time and manufacturing cost. In the present scenario, the generative CAPP is admired because of responsiveness to the rapid changes in product varieties and its design. To optimize the processing cost in process planning problem, Genetic Algorithm (GA) has been explored by Zhang *et al.* (1997) and Li *et al.* (2005). These approaches consider the precedence relationship among the operations and employ various decision making techniques to arrive at the best solution. Ben Arieh and Chopra (1997) have modeled a process plan selection problem on the basis of case based reasoning approach. A graph theoretical approach with minimum cost constraints, for minimization of number of tools and auxiliary devices, has been proposed by Kusiak and Finke (1998). In this approach, all the formulations and computational work have been done with the help of integer programming (needing prohibitive computational time to arrive at near optimal solution).

In manufacturing environment, most of the information is imprecise and vague. Owing to these facts, Tiwari and Vidyarthi (1998) and Rai *et al.* (2002) have applied the fuzzy based model to select a process plan considering machining time and cost, processing sequence, setup etc. A two step approach is deployed to incorporate the various similarities like machine similarity, operation sequence similarity in the given approach. First, the alternative process plans are generated and then the same are consolidated. Neural Network based approach for set up planning has been exercised by Ming and Mak (2000 a), whereas Ming and Mak (2000 b) used a Hopfield Neural Network with genetic approach for an optimal process plan selection problem. Ben Arieh *et al*. (2003) have used generalized Traveling Salesman Problem (TSP) to select the process plan for a rotational part considering the tolerance analysis.

From the aforementioned literature review, it is observed that, many of the researchers reported that the process planning problem consist of several decision making problems like appropriate machine selection, tool selection, and selection of Tool Approach Direction (TAD). Therefore, it is a well known computationally complex problem (Morad and Zalzala, 1999) and warrants the application of random search techniques. From the contribution of past literature, it can be found that random search techniques like Simulated Annealing (SA), Tabu search, Sample Sort Simulated Annealing (SSA), Ant Colony Algorithm etc. can efficiently map such complex problems but all of these have their own limitations and drawbacks. Therefore, it can be said that, nowadays we are drowning in the ocean of random search techniques but still starving for such a search technique that performs efficiently in each decision making problem. Due to this starvation, researchers are motivated to develop a hybrid search technique that enjoys the exquisiteness of more than one search techniques (Mishra *et al.* 2005, Su and Shuie 2003, Jones *et al.* 1995). Keeping this in mind, a hybrid Tabu-Sample Sort Simulated Annealing (TSSA) has been proposed in this paper. TSSA incorporate the delicacy of Tabu, Sample Sort Simulated Annealing and Fuzzy Logic Controller.

The remainder of the paper is arranged in the following sequence: section 2 delineates about the problem environment along with mathematical modeling of the problem. The overview of the proposed heuristic has been described in section 3. Section 4 reveals the different parameters and steps of the proposed heuristic. To demonstrate the efficacy of the proposed heuristic, an illustrative example, adopted from literature along with randomly generated data sets have been explained under the section 5. To validate the performance of the proposed algorithm results and discussions are presented in the section 6. Finally, the summary and conclusions with a note about its future scope is reported in the section 7.

2. Problem Description

In CIMS, several resources persist with different capabilities and capacities with different cost implications. In this paper, the cost of the finished product has been considered as criterion for optimization. Most of the earlier researchers have considered only machining cost, tooling cost, machine changing cost, tool changing cost, and set up changing cost but neglected the scrap or rejection cost. The parts that do not meet the design specification or the parts below and above the tolerance limit will be rejected and are known as *scrap*. For example, in the manufacturing of a part, the process involves several operations that can be performed on alternative machines. Each process like milling, drilling etc. is subjected to certain design specifications. After inspection, if some parts do not meet the tolerance limit, then the same is rejected as shown in figure 1.

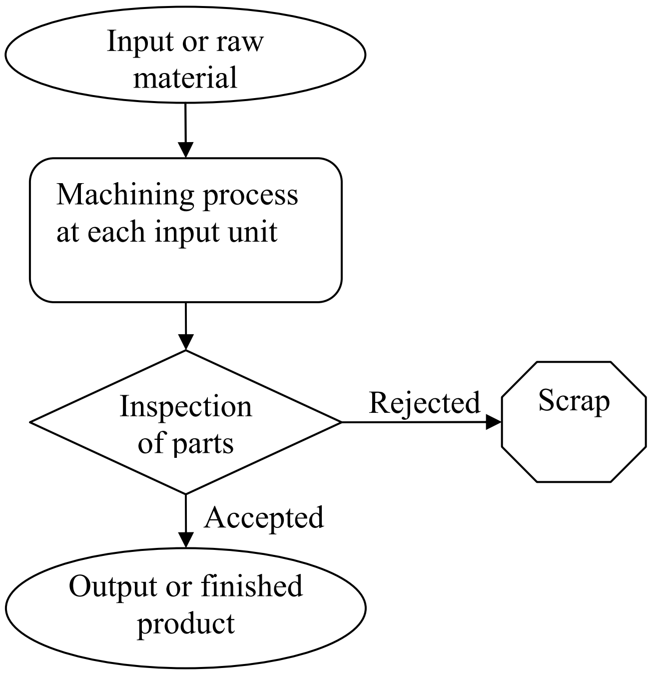

Figure 1: Transformation Process of a Part from raw Material to Finished Product

To ease the solution strategy undertaken, CAPP problem is modeled as Traveling Salesman Problem (TSP) with precedence relationship. In this model traveling distance between the two nodes corresponds to operation cost between the operations. Selection of machines and tools for each operation is not trivial due to the availability of alternative machines, tools, fixtures and TAD and are selected on the basis of their operation cost. The cost of finished product is associated with processing cost of each operation, raw material cost and scrap cost. The processing cost is primary concern that includes, machining cost, tooling cost, machine changing cost, tool changing cost, and set-up changing cost. At this stage, any detailed information about tool paths and machine parameters is not available, therefore, only operation type and their sequences are determined. The detailed mathematical formulations of these costs are presented below.

(i) Machining Cost (MC): the cost of required machining to perform the operations is known as machining cost. It is mathematically defined as:

$$MC = \sum_{i=1}^{n} MCI_i \tag{1}$$

Where

MCI_i is the machining cost index of machine i.

n is the total number of operations

(ii) Tooling Cost (TC): Tooling cost is the cost of tools required to perform all the machining operations. The mathematical formulation is as follows:

$$TC = \sum_{i=1}^{n} TCI_i \tag{2}$$

Where

TCI_i is the tooling cost index of tool i.

(iii) Machine Changing Cost (MCC): The cost of changing of machines between two operations is known as machine changing cost. Machine change cost between machine M_i and M_{i+1} is mathematically defined as:

$$MCC = MCCI \sum_{i=1}^{n-1} \mu\left(M_{i+1} - M_i\right) \tag{3}$$

When

$$\mu\left(M_i - M_j\right) = \begin{cases} 1 & if\ M_i \neq M_j \\ 0 & otherwise \end{cases} \tag{4}$$

Where

$MCCI$ is the machine changing cost index.

(iv) Tool Changing Cost (TCC): The cost of the changing of tools between two operations on the same machine is known as tool changing cost. Tool change cost between tool T_i and T_{i+1} is defined as follows:

$$TCC = TCCI \sum_{i=1}^{n-1} \left(1 - \mu\left(M_{i+1} - M_i\right)\right)\left(\mu\left(T_{i+1} - T_i\right)\right) \tag{5}$$

When

$$\mu\left(T_i - T_j\right) = \begin{cases} 1 & if\ T_i \neq T_j \\ 0 & otherwise \end{cases} \tag{6}$$

Where

$TCCI$ is the tool changing cost index.

(v) Set-up Changing Cost (SCC): This cost is taken in account when two operations are performed on same machine but have different Tool Approach Direction (TAD). TAD change cost between TAD_i and TAD_{i+1} is mathematically defined as:

$$SCC = SCCI \sum_{i=1}^{n-1} \left(1-\mu\left(M_{i+1}-M_i\right)\right)\left(\mu\left(TAD_{i+1}-TAD_i\right)\right) \tag{7}$$

Where

$$\mu\left(TAD_i - TAD_j\right) = \begin{cases} 1 & if\ TAD_i \neq TAD_j \\ 0 & otherwise \end{cases}$$

Overall Processing Cost (OPC): The overall processing cost is the sum of all the aforementioned costs and it is defined as:

$$OPC = MC + TC + MCC + TCC + SCC \tag{8}$$

Output Cost: This is the main objective of this paper. This cost is associated with overall processing cost, scrap cost and raw material cost. For calculating the output cost, other factors like the scrap fraction, input coefficient, scrap coefficients etc. should be taken into account. The calculation of these factors along with the objective function is as given below.

Scrap Fraction: It is the ratio between the number of rejected units and number of input units at each operation. It is given as follows:

$$\left(S_f\right)_i = \frac{\left(N_s\right)_i}{\left(N_{in}\right)_i} \tag{9}$$

Where

$\left(S_f\right)_i$ = Scrap fraction at i^{th} operation

$\left(N_s\right)_i$ = Number of part rejected at i^{th} operation

$\left(N_{in}\right)_i$ = Number of input units of i^{th} operation

Input Coefficient: It is a technological coefficient and represents the requirement of input per unit of output. It can be expressed as follows:

$$\left(K_{in}\right)_i = \frac{\left(N_{in}\right)_i}{\left(N_{out}\right)_i} \tag{10}$$

$\left(K_{in}\right)_i$ = Input coefficient of i^{th} operation

Scrap Coefficient: It is the ratio of generated scrap and number of output unit. It is defined as:

$$\left(K_s\right)_i = \frac{\left(N_s\right)_i}{\left(N_{out}\right)_i} \tag{11}$$

Where

$\left(N_{out}\right)_i$ = Number of output units of i^{th} operation

$\left(K_s\right)_i$ = Scrap coefficient of i^{th} operation

If all the aforementioned costs are changed in the currency flow system, the cost flow balance equation will be as follows:

$$\left(C_{in}\right)_i\left(N_{in}\right)_i + \left(N_{in}\right)_i\left(OPC\right)_i = \left(C_{out}\right)_i\left(N_{out}\right)_i + \left(C_s\right)_i\left(N_s\right)_i \tag{12}$$

Where

$(C_{in})_i$ = Average input cost per unit for operation i

$(C_{out})_i$ = Average output cost per unit for operation i

$(C_s)_i$ = Average scrap cost per unit for operation i

$(OPC)_i$ = Operation cost per unit for operation i

This formula can be written as

$$(C_{out})_i = (C_{in})_i (K_{in})_i - (C_S)_i (K_S)_i + (K_{in})_i (OPC)_i \tag{13}$$

And the overall objective function will be:

$$min(F) = min \sum_{i=1}^{n} (C_{out})_i = min \sum_{i=1}^{n} \left((C_{in})_i (K_{in})_i - (C_s)_i (K_s)_i + (K_{in})_i (OPC)_i \right) \tag{14}$$

The main objective of this paper is to minimize the value of function F.

The next section delineates the search algorithms and their specific features in the chronological manner.

3. Background of Proposed Algorithm:

3.1. Sample Sort Simulated Annealing

Simulated Annealing (SA) is a random search optimization technique based on physical annealing of solid. In physical annealing, a metal is brought to its lowest energy state, first by heating it to a very high temperature and then cooling down to a low temperature (Metropolis *et al.,* 1953). The main features of SA that make it more sophisticated as compared to others are perturbation, annealing schedule and transition probability (Kirkpatrick *et al.,* 1983). But SA performs well only at higher temperature due to the large number of accepted moves and requires more number of generations to search an optimal/ near optimal solution. In order to overcome these drawbacks of SA, parallel SA has been utilized by many of the researchers like Cassotto *et al.* (1987), Witte *et al.* (1991) etc and it works efficiently at lower temperature. But it performs worse in the absence of serialization and synchronization property. To overcome above drawbacks, Thompson and Bilbro (2005) proposed a new multi start algorithm known as Sample Sort Simulated Annealing (SSA) that is an extended version of single Markov chain over the array of samplers. It has m samplers and all the samplers search the optimal/near optimal solution in parallel mode. The main property of Sample Sort Simulated Annealing (SSA) is that it enjoys both the flavor of serial SA and parallel SA. Hence, it can perform efficiently at higher and lower temperatures and gives consistent performance. The number of samplers is predefined at the starting of Sample Sort Simulated Annealing algorithm and each sampler operates at different static temperature. A minimal cost solution is propagated to other samplers probabilistically. It permits the samplers in exchanging the solution with their neighbor sampler. All the samplers are arranged in a row and any sampler can exchange the solution with next one or it may be exchanged after skipping more samplers. Serialization is maintained by adjusting the probability of accepting a higher cost solution. Thus this property is maintained in very easier way than other existing periodically interacting parallel SA algorithms. The pseudo code for sample sort annealing algorithm is given as follows (Thompson and Bilbro, 2005):

$for\ \ n = 1\ to\ n_{iterations}$

$for\ \ i = m\ to\ 1$

$for\ \ j = max\ (1, i - N)\ to\ min\ (n, i + N)$

$accept\ \ s' = s_j\ \ with\ \ probabilit\ y\ e_1(s'/s_j)$

$end\ \ j$

$end\ i$

$for\ i{=}m\ to\ 1$

$accept\ s'\ with\ probability\ e_2\ (s'/s_i)$

$end\ i$

$end\ n$

Here, n = The maximum number of iterations.
m = Number of samplers
T_k = Operating temperature of k^{th} sampler

In this procedure, each sampler updates itself two times in all iterations. First, it exchanges with the state of neighboring sampler, then with candidate solution which is uniformly generated around its current state in its neighborhood. The state s_j of a neighboring sampler j acts as a candidate s' for sampler i, by assuming that the j^{th} sampler has reached equilibrium. The probability e_1 is used to update the current state of sampler j whereas e_2 for the other kind of updating (Thompson and Bilbro, 2005). The probability e_1 and e_2 can be defined as:

$$e_1 = (s'/s_i) = \min\left(1, \exp\left(-\left(f(s') - f(s_i)\right)\left(\frac{1}{T(s_i)} - \frac{1}{T(s')}\right)\right)\right) \tag{15}$$

$$e_2\ (s'/s_i) = min\left(1, exp\left(-\frac{f\ (s') + f\ (s_i)}{T_i}\right)\right) \tag{16}$$

SSA algorithm with all its advantages has some demerits too. In SSA there exists no concepts of memory list of recently visited solutions and hence there is a possibility to return to previously visited solution. This also leads to more iteration and longer computational time.

3.2. Tabu Search

Tabu Search, a higher-level search heuristic, is used for solving optimization problems (Glover (1990)). It explores the solution space by repeatedly making moves from one solution, *s*, to another solution *s'*, located in the neighborhood of *s*. The process of searching the neighborhood solution continues until it reaches the global optima; at each step the generated solution is scrutinized by the evaluation of objective function. During optimization, it utilizes the list of prohibited neighboring solutions known as tabu-list, to escape from local optima (Swarnkar and Tiwari, 2004). The pseudo code of Tabu search algorithm is given below:

begin

s ← *initial solution*

TL ← *tabulist*

initialize

for i = 1 to n

generate the solution s' from s by perturbation

if f(s') > f(s) or f(s') > aspiration

$s = s'$

$TL \leftarrow s$

if $f(s) \geq f(s_b)$

$s_b = s$

endif

endif

end

The main features of Tabu search are aspiration and diversification. Aspiration restricts the search from being trapped into a solution surrounded by tabu neighbors, thus helping in the checking condition for the acceptance of new generated solutions. It allows search to override the tabu status of the solution and also provide backtracking of recent solutions as they lead to a new path towards a better solution. Whereas, diversification is used, to explore sub-domains that may not be reached otherwise. Thus, the search is redirected from a different initial solution. The negative aspect of tabu search is cycling-loop will be encountered if the search moves to a previously visited solution that has not been existed in tabu list for the last two iterations.

From the above discussion, it can be comprehended that both algorithms i.e. Sample Sort Simulated Annealing (SSA) algorithm and Tabu search, have some limitations. Therefore, to take the advantages of both algorithms and to mitigate their adverse effects, a hybrid algorithm has been proposed known as Tabu-Sample Sort Simulated Annealing (TSSA). A fuzzy logic controller is also applied in SSA for adaptive change in swapping rate. The mechanism of FLC is delineated in the upcoming sub-section.

3.3. Fuzzy Logic Controller

The CAPP information is generally imprecise and vague. Therefore, any mathematical relationship can not be established among them for attaining any fruitful results. To map and control such imprecise information, the fuzzy logic is the best suited idea. Kim *et al.* (2003) here used a fuzzy logic controller for controlling the mutation rate and crossover rate at each generation. In this article, a fuzzy logic controller has been employed for updating the swapping rate. During the whole optimization process, the adaptive fuzzy logic controller changes the value of swapping rate according to the average fitness value of samplers at the previous and current generation. First, a hypothetical inference engine has been constructed as shown in figure 2.

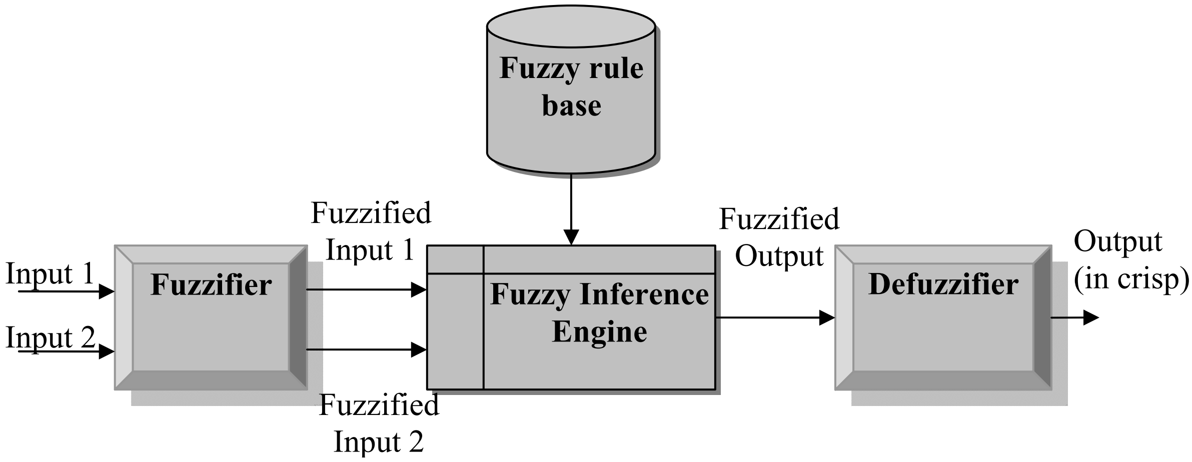

Figure 2: A Fuzzy Inference Engine

The inference engine has three parts: first is the fuzzifier second is an inference section and third one is the defuzzifier. This whole system is based on the human intelligence, their past experience and the experts' knowledge in the specified area. This inference engine establishes the relations between causes and its effects. In the present case there is no mathematical relationship between the fitness value and the swapping rate, therefore, the fuzzy rule base has been applied to establish a relationship between them.

The working of the inference engine can be understood from the following steps:

1. Calculate the average fitness value at *t* and *t-1* generation using following formula:

$$\Delta\bar{f}(v_t)=\left(\frac{\sum_{k=1}^{ss} f\left(v_{k,t}\right)}{ss}\right)w \tag{17}$$

$$\Delta\bar{f}\left(v_{t-1}\right)=\left(\frac{\sum_{k=1}^{ss} f\left(v_{k,t-1}\right)}{ss}\right)w \tag{18}$$

Where, $f\left(v_{k,t}\right)$= Fitness value of kth sampler at tth iteration

ss = Sampler size

w = Scaling factor

2. Normalize the average fitness value $\Delta\bar{f}(v_t)$ and $\Delta\bar{f}\left(v_{t-1}\right)$ at t and $t-1$ generation in the range of [-1, 1].

3. Determine the membership function for fitness value according to the linguistic variables (Zadeh 1974, 1975, 1976, 1978). These variables are negative larger (NR), negative large (NL), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), positive large (PL), positive larger (PR). The triangular fuzzy number for each linguistic variable is shown in the figure 3.

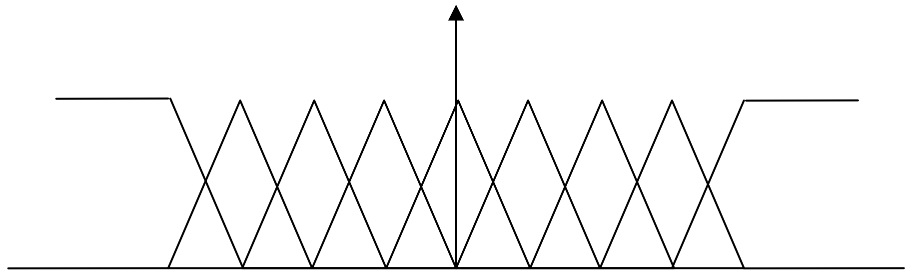

Figure 3

4. Fuzzify the normalized average fitness value according to their membership function.
5. The decision making process starts after fuzzification, which is a very critical issue of FLC. The decision is made on the basis of the fuzzy membership value of both inputs (average fitness value at previous and current iteration). The output, which is swapping rate, is guided by the rule base system. This rule base system is based on the IF/THEN rules (Mamdani 1974, Mamdani and Gains 1981), where IF statements or the antecedent part of rule represent the inputs and THEN statement or the consequent part represents output. In this rule base each input has 9 membership functions, so the number of rules is 9*9=81. The fuzzified decision rule base is shown in table 1, and table 2 shows the defuzzified values of the linguistic decision variable.

Table 1. Fuzzified Decision table for Control Action of Swapping Rate

$\overline{f}(t-1)$	$\overline{f}(t)$								
	NR	**NL**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PL**	**PR**
NR	NR	NL	NL	NM	NM	NS	NS	ZE	ZE
NL	NL	NL	NM	NM	NS	NS	ZE	ZE	PS
NM	NL	NM	NM	NS	NS	ZE	ZE	PS	PS
NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM
ZE	NM	NS	NS	ZE	ZE	PS	PS	PM	PM
PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PL
PM	NS	ZE	ZE	PS	PS	PM	PM	PL	PL
PL	ZE	ZE	PS	PS	PM	PM	PL	PL	PR
PR	ZE	PS	PS	PM	PM	PL	PL	PR	PR

(Here, NR=negative larger, NL=negative large, NM=negative medium, NS=negative small, ZE=zero, PS=positive small, PM=positive medium, PL= positive large, PR= positive larger)

Table 2. Defuzzufified Decision Table for Control Action of Swapping Rate

j i	D (i, j)								
	-4	-3	-2	-1	0	1	2	3	4
-4	-4	-3	-3	-2	-2	-1	-1	0	0
-3	-3	-3	-2	-2	-1	-1	0	0	1
-2	-3	-2	-2	-1	-1	0	0	1	1
-1	-2	-2	-1	-1	0	0	1	1	2
0	-2	-1	-1	0	0	1	1	2	2
1	-1	-1	0	0	1	1	2	2	3
2	-1	0	0	1	1	2	2	3	3
3	0	0	1	1	2	2	3	3	4
4	0	1	1	2	2	3	3	4	4

6. After the decision making process, a fuzzy output will be obtained. For using it as a crisp value, this value must be defuzzified. Defuzzifier, the last part of inference engine, is employed for defuzzification. It is normalized in the range of [-0.1, 0.1]. The membership function for the output, change in swapping rate $\Delta s_W(t)$, is shown in the figure 4.

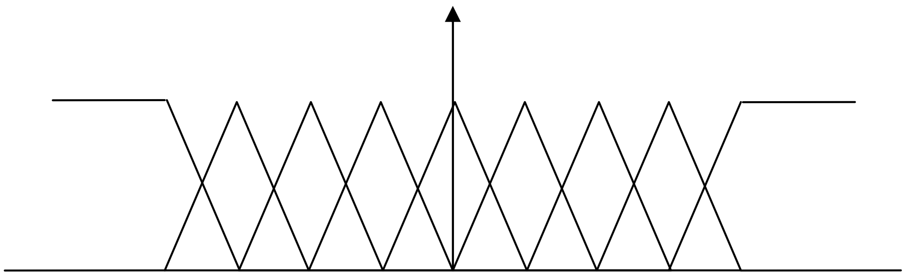

Figure 4.

Finally update the change of swapping rate by following equation.

$$s_w(t) = s_w(t-1) + \Delta s_w(t) \qquad (19)$$

4. Proposed Heuristic

The proposed hybrid Tabu-Sample sort Simulated Annealing (TSSA) based heuristic is applied to a multi constraint process planning problem. The algorithm can run on any randomly generated initial solution. From experiments it is found that performance becomes better with randomly started solutions than sequential rule based starting solutions. It is also observed that random initial solutions effectively diversify the search space and simultaneously minimize computational time (Anderson and Ferris 1994). Hence, in this research initial population is generated randomly. The performance of hybrid Tabu-SSA based heuristic is governed by following parameters.

(i) Neighborhood Generation (Perturbation):

A new solution can be generated from previous solution by applying different perturbation scheme. In the present problem of process planning, the simple perturbation scheme can not be performed. In this problem, each sampler has four rows. Where first row stands for the operation sequence, second row for machine, third row for selected tools, and forth row for TAD. The sampler with its all four rows has been shown in figure 5. A perturbation scheme known as feasible insertion method has been proposed for swapping operation sequence. An operation is selected randomly from the operation sequence and simultaneously, the potential positions at which the selected operation can be placed should be identified. After it, the machines, tools and TAD are legalized.

Operation sequence	11	12	1	2	3	4	5	13	14	15	10	9	6	7	8
Machine ID	2	2	2	1	3	1	2	2	2	1	1	1	1	1	2
Tool ID	2	6	2	1	5	7	8	7	8	9	1	1	1	1	2
TAD	-X	-Z	-X	+Y	-Z	-Z	+Z	-Z	+Z	-Z	+Y	+Y	+Y	+Y	+Y

Figure 5: Representation of a Sampler

(ii) Transition Probability:

When a random neighboring solution is generated, its fitness function is evaluated. If the fitness value is improved it is accepted otherwise transition probability should be checked with a uniform random number. If transition probability is higher the solution is accepted otherwise rejected. In proposed Tabu-SSA algorithm, Boltzmann function is replaced by Cauchy function during the annealing process. This not only increases the annealing schedule but also provide more opportunity to escape from local minima. A comparison between the behavior of Cauchy function and the Boltzman function is shown in figure 6. It illustrates that the Cauchy function's wing has reached the deeper valley than Boltzman function, thus Cauchy function attains a significant value. Kumar *et al.* (2003) stated that transition probabilities must satisfy two conditions: *Non-negativity* and *Normalization* and the same have been verified for the Boltzmann function. *Appendix A* quantitatively demonstrates and validates both the properties for the Cauchy function. Another promising feature of Cauchy function is that it obeys the *principle of detailed balance*, which is also described in *Appendix A*. in this paper Cauchy function based transition probability is denoted by TP. Inferior offspring is selected if, TP >R where R is a random number.

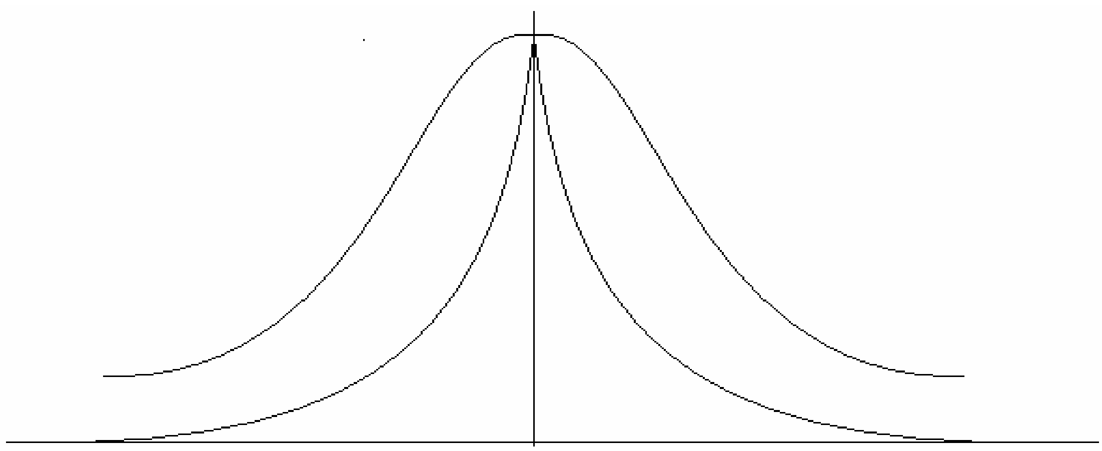

Figure 6: Comparison between Cauchy Distribution Function and Boltzman Distribution Function

For each solution, which is inferior in comparison with previous solution, transition probability can be calculated using equation (20).

$$TP = \frac{T(K)}{T^2(K) + (\Delta f)^2} \tag{20}$$

Where

$T(K)$= Annealing temperature at k^{th} generation

(Δf)= difference between the fitness values of the previous solution and perturbated solution.

(iii) Annealing Schedule:

Cooling schedule determines the annealing temperature and transition probability during each iteration. The quality of the solution also depends on the lowering of temperature or cooling. If cooling rate is less, the better solution is accepted but the convergence rate will be low. A theoretical optimal cooling schedule given by Hajek (1988) is shown below:

$$T_k = \frac{T_0}{\ln(1+k)} \tag{21}$$

Where, T_0 is the initial temperature and T_k is the temperature at k^{th} generation. The above cooling schedule is asymptotically approaches the global minimum or maximum as $k \to \infty$. Hence, it takes more computational time to converge. In this paper, a new annealing schedule has been explored to overcome above drawback. This new annealing schedule gives more opportunity to the searching procedure to escape from local minima and converged quickly toward global minima.

$$T = \alpha * T \tag{22}$$

Where, α is a cooling co-efficient and its value lies between 0.80 and 1.0.

(iv) Tabu List:

Tabu list, a set of recently visited solutions, is used to check whether the perturbated solution is previously visited or not during the each iteration of algorithm. Thus, it helps in preventing the algorithm to revisit the pre-visited solutions. This feature of hybrid Tabu-SSA algorithm helps the searching procedure to converge toward the global optima in minimum number of iterations and saving significant computational time.

(v) Aspiration:

Aspiration is connected with tabu search and its function is to restrict the search from being trapped at a solution surrounded by tabu neighbors. If a solution has neighborhood of only tabu solutions, the one with objective function value greater than aspiration is chosen to explore further. It is denoted by variable '*A*'.

(vi) Reject:

The perturbated solutions are often rejected on the basis of probability consideration. Rejection gets increased by 1 whenever, a solution is not selected through the probability consideration. After a certain limit of rejection, it is assumed that no superior solution exists in the neighborhood and the search reached a near optimal / optimal solution.

(vii) Stopping Criteria:

To stop the Tabu-SSA algorithm from roaming into the solution space, two stopping criteria are adopted:

- *Number of Generations:* During intensive computational experiment, it has been found that as the number of iteration increases, equivalent temperature falls to a minimum level. Any further reduction in temperature would not be useful because at low temperature the possibility of accepting inferior solution is very small and results obtained are virtually indistinguishable from the optimal solution.
- *Variable reject*: When reject variable acquired predetermined fixed value (In this paper 4 is determined.), no optimal or near optimal solution is achieved during the last four iteration, therefore the probability of obtaining any better solution is small and stop the search procedure from exploring the search space.

(viii) Steps of Algorithm:

The mechanism of the proposed hybrid Tabu-SSA (TSSA) algorithm with fuzzy logic controller is delineated in the following steps:

Step 1. Generate m (m=10) initial samplers (s_1, s_2, s_3 …s_m). These samplers are generated according to the following steps:

Step 1.1 Randomly select one operation having no predecessors.

Step 1.2 Remove the edges between the selected operations and its successors.

Step 1.3 Among the remaining operations, again randomly select another operation having no predecessors.

Step 1.4 Repeat step 1.2 and 1.3 until all the operations are selected.

Step 1.5 Revisit all selected operations, from the first operation, start randomly assigning of the machine, tool and tool approach direction from their available alternatives.

Step 1.6 Repeat the step 1.5 until a feasible selection of machine, tool, and TAD for each operation has been performed.

Step 1.7 Repeat all aforementioned steps until all the samplers are generated.

Step 2. All the individuals are evaluated and measure their fitness value.

Step 3. Initialization:

Iteration i =1, tabu list (TL) = Φ, reject = 0, best solution (s_b) = s; for each sampler a static temperature or initial temperature is assigned. Let the static temperatures are T_1, T_2, T_3… T_m for 1, 2, 3… m samplers respectively. The calculation of static temperatures is delineated in the next step.

Step 4. For determining the static temperature of each sampler, the following steps are carried out:

Step 4.1 For each sampler generate 100 random solutions.

Step 4.2 Evaluate objective function for all the solutions and estimate the average fitness value $\overline{f}$.

Step 4.3 Calculate the standard deviation σ considering the fitness value of each operation.

Step 4.4 Using the value of standard deviation and average fitness value, energy is calculated. The following formula is applied for calculation of energy.

$$\Delta E = \overline{f} + \sigma \tag{23}$$

Step 4.5 For determining the highest temperature $\left(T_H\right)$, following formula is employed:

$$p = exp\left(-\frac{\Delta E}{T_H}\right) \tag{24}$$

Where $p = 0.75$,

T_H = highest temperature

Step 4.6 Arrange the solution in the ascending order according to their fitness values. The energy (ΔE) is calculated by taking the difference between highest fitness value and fitness value of next solution in the order.

Step 4.7 For determining the lowest temperature (T_L), the following expression is implemented:

$$p = exp\left(-\frac{\Delta E}{T_L}\right) \tag{25}$$

Where p =0.01

T_L = Lowest Temperature

Step 4.8 Static temperature of k^{th} sampler can be found out by the following formula:

$$T_k = T_H \alpha^{m-k} \tag{26}$$

And $1 \leq k \leq m$. It is monotonically increasing function.
Where α = cooling co-efficient

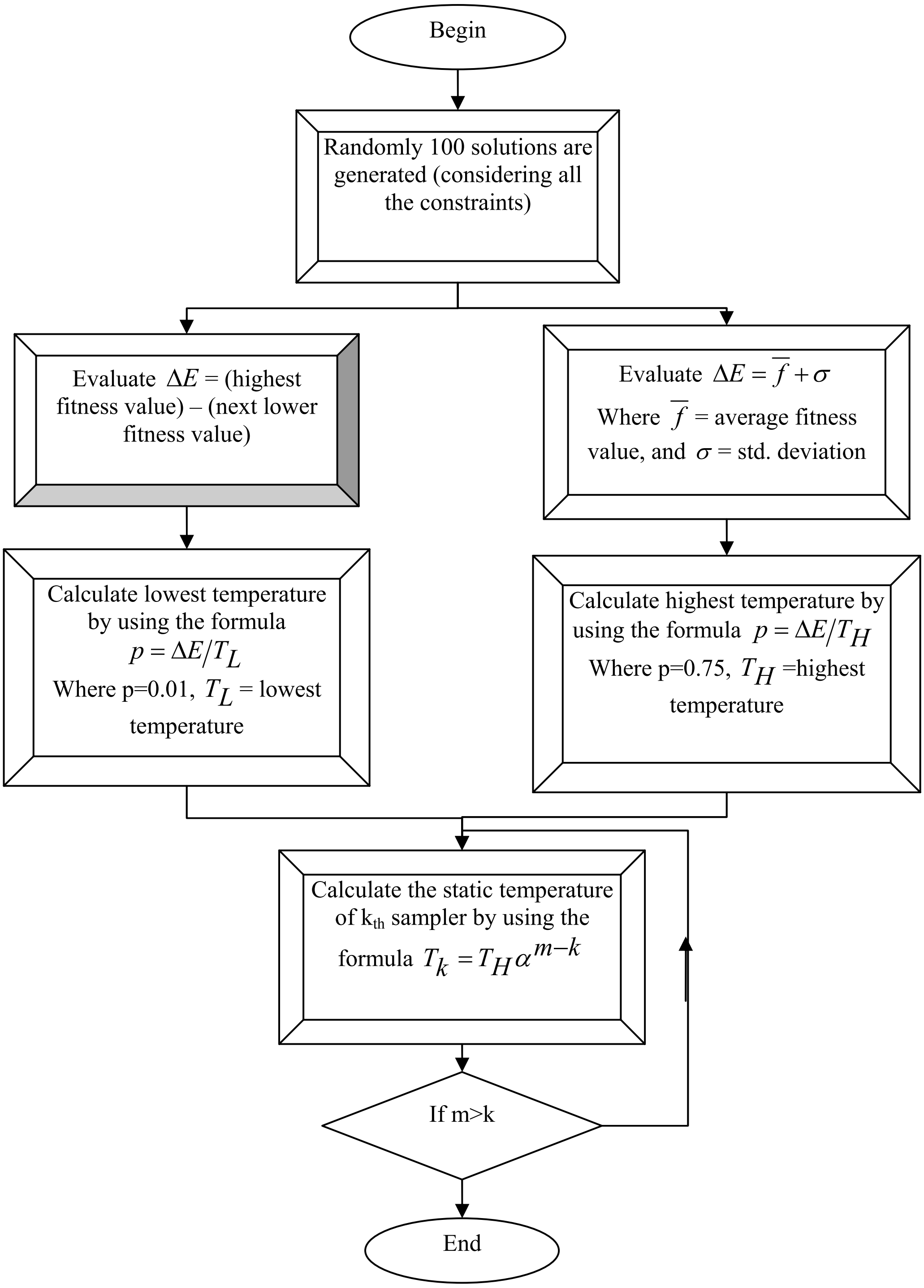

Figure 7: Flow Chart for Allocation of the Static Temperature to the Samplers

Step 4.9 For estimating the value of cooling co-efficient (α), first fix the temperature T_L and T_H for first and last sampler and use the following expression:

$$T_L = T_H \alpha^{m-1} \tag{27}$$

The pictorial representation of step 5 has been shown in figure 7.

Step 5. For updating the samplers or generation of the neighboring samplers, a perturbation scheme is employed. The perturbation scheme has been described in sub-section 4.1(i).

Step 6. The feasibility of the newly generated solution is also checked and if the generated solution is feasible than go to step 7 else go to step 5 for generation of a feasible solution.

Step 7. Check the solution by tabu list. If the neighboring solution exists in tabu list then go to step 8 otherwise go to step 9.

Step 8. Check the solution by aspiration value. If the fitness value of new generated solution is higher than aspiration value, go to next step otherwise go to step for the generation of a new solution.

Step 9. Calculate the difference (ΔE) between the previous and its neighboring solution's fitness values. Let the initial solution is s and the solution generated by perturbation is s'. Energy (ΔE) can be calculated as follows:

$$\Delta E = f\left(s'\right) - f(s) \tag{28}$$

If the value of $\Delta E \geq 0$, go to next step otherwise go to step 11.

Step 10. Replace the previous solution *s* by its neighboring solution s'. This solution is also included in the tabu list. The aspiration value is updated by the fitness value of the solution s', a perturbated solution.

Step 11. If the difference ΔE_b between the fitness values of perturbated solution and the best solution is greater than 0, then go to next step else go to step 16. it is mathematically shown as follows:

$$\Delta E_b = f\left(s'\right) - f\left(s_b\right) \tag{29}$$

Where, s' = perturbated solution
s_b = best solution

Step 12. Replace the best solution s_b by the perturbated solution s' and go to step 16 and assign zero value to reject.

Step 13. Calculate the probability by using the following equation:

$$p = \exp(-\Delta E/T) \tag{30}$$

Where, T is the temperature at that iteration.

Now a random number (R_a) is generated between (0, 1). If the calculated probability is less than random number, go to step 15 else go to next step.

Step 14. Replace the existing solution by the perturbated solution and it is included in the tabu list and updates the aspiration value by replacing it with perturbated solution's fitness value.

Step 15. If the probability is less than random number, reject the solution and increase the number of reject by one. If reject reaches to a pre determined maximum limit (in this paper, it is taken as 4), then go to step 20 other wise go to step 16.

Step 16. Now a Fuzzy Logic Controller (FLC) is used for regulation of the swapping rate. This regulation will be adaptive in nature. The working of FLC is described in the following steps:

Step 16.1 Take the inputs for FLC that are change in the average fitness value $\Delta\overline{f}\left(v_t\right)$ and $\Delta\overline{f}\left(v_{t-1}\right)$ in the generation *t* and *t-1* respectively, which are calculated in equation 17 and 18.

Step 16.2 A fuzzy decision table, which is based on the fuzzy values of $\Delta\overline{f}\left(v_t\right)$ and $\Delta\overline{f}\left(v_{t-1}\right)$, is used to control the swapping rate.

Step 16.3 Now calculate the change in swapping rate $\left(\Delta sw(t)\right)$ applying the following formula:

$$\Delta s_w(t) = \phi D(i,j) \quad (31)$$

Where ϕ = given value to regulate the increasing and decreasing range for the swapping rate.

$D(i,j)$ = output from the defuzzified decision table

Step 16.4 Update the swapping rate according to the equation 19.

Step 17. The samplers are swapped according to updated swapping rate.

Step 18. Increase the number of iterations by one and update the temperature according to equation 22. If the number of iterations is not equal the maximum number of iterations, repeat step 5 to step 17 otherwise go to next step.

Step 19. Stop the iteration for that sampler, if reject for all samplers has reached on its maximum limit, go to next step else go to step 18.

Step 20. Stop the iteration and s_b is the optimal or near optimal solution.

The proposed algorithm performs according to the above mentioned steps.

5. Test Problems

To exhibit the efficacy and robustness of proposed model and investigate the capability of Tabu-SSA algorithm, data set with increasing complexity have been considered as benchmark dataset. These test problems are outlined as follows.

Case study-I: First case study maps the scenario described by Zhang *et al.* (1997) and has been considered with an intension to reveal the efficacy of the proposed algorithm. In this case, the manufacturing unit produces a multi feature prismatic part having 19 design features. Here, eight different types of tools are assigned on three different types of machines and three type of Tool Approach Directions (TAD) are taken into account. Each machine and tool has different cost of processing. The information related to the operations, machines, tools, and Tool Access Direction (TAD) and precedence relations between different operations are given in Zhang *et al.* (1997).

Case study-II: In this case, data set incorporating the feature of proposed model (process plan selection problem considering scrap cost) has been generated randomly. The case study consists of a part having 15 different operations. The precedence relation between the operations, due to different technological rules such as filterability, tolerance factor, operation stages requirement, for machining is listed in table 3. All these operations are performed on the 4 different machines. The available machines are CNC lathe (M-01), CNC milling (M-02), drilling machine (M-03) and a boring machine (M-04). Nine different tools can be engaged on these machines for performing all the operations. The available tools (T_1, T_2, T_3 …T9) are turning tool, milling cutter, drills, reamers etc. For each operation, the percentage of generated scrap is fixed at each machine. The information related to the scrap, operations, machines, tools, and Tool Access Direction (TAD) is shown in table 4. The cost indexes such as machine and tool cost index are listed in the table 5-6.The other cost index of machine (MCCI), Tool (TCCI), setup (SCCI) changes is shown in table 7. The raw material cost, scrap cost and the number of input units for this case study have been represented in table 8.

Table 3. Precedence Relationship among the Different Operations for case study II

	No. of Operation														
No. of Operations		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
	2	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4. Available Resources for the Different Machining Operation for Case Study II

S. NO.	FEATURE ID	OPERATIONS	MACHINE ID	TOOL ID	T.A.D.	SCRAP (%)
1	F1	Facing	M-01, M-02	T-01, T-02	-X	2,5
2	F2	Turning	M-01	T-01	+Y	2
3	F3	Drilling	M-01, M-02, M-03	T-05	+Z,-Z	5,5,2
4	F3	Boring	M-01, M-02, M-04	T-07	+Z,-Z	5,5,2
5	F3	Reaming	M-01, M-02, M-03	T-08	+Z,-Z	5,10,2
6	F4	Turning	M-01	T-01	+Y	10
7	F5	Turning	M-01	T-01	+Y	2
8	F6	Undercutting	M-01, M-02	T-03, T-02	+Y	2,5
9	F7	Turning	M-01	T-01	+Y	2
10	F8	Turning	M-01	T-01	+Y	2
11	F9	Facing	M-01, M-02	T-01, T-02	-X	2,5
12	F10	Drilling	M-01, M-02, M-03	T-06	+Z,-Z	5,5,2
13	F10	Boring	M-01, M-02, M-04	T-07	+Z,-Z	5,8,2
14	F10	Reaming	M-01, M-02, M-03	T-08	+Z,-Z	5,8,2
15	F10	Threading	M-01, M-03	T-09	+Z,-Z	2,5

Table 5. Machining Cost Index (for Case Study II)

S.NO.	Machine ID	Type	Cost of Machining ($)
1	M-01	CNC Lathe	52
2	M-02	CNC Milling	60
3	M-03	Drilling Machine	22
4	M-04	Boring Machine	50

Table 6. Tooling Cost Index (for Case Study II)

S.NO.	Tool ID	Type	Cost of Tooling ($)
1	T-01	Turning Tool	10
2	T-02	Milling Cutter	15
3	T-03	Parting Tool	10
4	T-04	Facing Tool	10
5	T-05	Drill φ 0.2	3
6	T-06	Drill φ 1.2	3
7	T-07	Boring Tool	15
8	T-08	Reamer	8
9	T-09	Threading Tool	8

Table 7. Changing Cost Index (for Case Study II)

S.NO.	TYPE	COST ($)
1.	Machine Changing Cost Index	300
2.	Tool Changing Cost Index	10
3.	Set up Changing Cost Index	90

Table 8. Some other Parameters (for Case Study II)

S. No.	Parameters	Units
1.	Raw Material Cost	45 $
2.	Scrap Cost	30 $
3.	Number of Input units	100

Case study-III: To demonstrate the strength of the proposed approach, 10 process plan selection problems with increased complexity have been considered. The data set for each problem is generated randomly to reproduce arbitrarily complex scenarios. Detailed description of the test problems such as: number of features, number of machines, number of tools, number of operations, are given.

7. Computational Experience

From the prominent research contribution in the broad realm of SSA and Tabu applications, this conclusion has been extracted that the requirement of number of iterations, for obtaining the optimal or sub-optimal solution, is significantly high. Therefore, it is a desirable and unavoidable issue to develop a metaheuristic by combining both the random search techniques to overcome the drawbacks of both the algorithms. Keeping the same, an evolutionary meta-heuristic termed as Tabu-Sample Sort Simulated Annealing (TSSA) has been developed to solve large sized combinatorial optimization problems in lesser number of generations. TSSA algorithm achieves the optimal/near optimal solutions for the objective considered in process planning problem and emphasizes it as a powerful meta-heuristic algorithm. Cauchy Probability Distribution function and Fuzzy Logic Controller (FLC) based swapping rate empower the algorithm to obtain optimal/near optimal solution in significantly less number of generations. At the initial stage, more interaction among all the samplers is required i.e. the swapping rate should be high so that sampler will be redirected towards the best sampler. After some iteration, all the samplers start to move towards global optima and less interaction is required. Therefore, swapping rate should be less. This variation in the swapping rate is adaptively controlled by Fuzzy Logic controller (FLC).

Performance of the proposed TSSA algorithm is found superior on comparing with GA, considering a well known data set adopted from literature. From extensive computational experiments, it is found that for first case study (mentioned in section 5) results obtained by proposed algorithm is an optimal one and is in tune with the result obtained by Zhang *et al.* (1997). Table 9 represents the results obtained from Genetic Algorithm (GA) while the results obtained from TSSA have been represented in table 10. It is evident from the results that proposed approach can perform efficiently on process plan selection problem. The other alluring property of proposed heuristic is that it can find the best process plan with reduced cost and other alternative solutions in lesser number of iterations. Alternative process plans are much important because if any machine or tool has been broken down; an alternative process plan without that machine or tool can be efficiently used.

Table 9. Results Obtained for case study I (adopted from (Zhang *et. al.* 1997))

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
OPID	14	5	6	4	21	18	17	22	23	15	16	19	20	1	2	7	8	9	11	12	13	3	10
MID	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
TID	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	2	3	4	2	3	4	7	5
TAD	+X	+Y	+Y	+Y	-Y	-Y	-Y	-Y	-Y	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z
The Total Operation Cost: 1739																							
Number of Tool Changes: 09																							
Number of Machine Changes: 01																							
Number of TAD Changes: 03																							

Table 10. Results obtained by the Proposed Heuristic (Tabu+ SSA+ FLC) for Case Study I

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
OPID	14	5	6	4	21	18	17	22	23	20	1	15	16	19	2	7	8	3	9	11	12	13	10
MID	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
TID	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	2	3	5	4	2	3	4	5
TAD	+X	+Y	+Y	+Y	-Y	-Y	-Y	-Y	-Y	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z
The Total Operation Cost: 1739																							
Number of Tool Changes: 09																							
Number of Machine Changes: 01																							
Number of TAD Changes: 03																							

Case study-II is more complex as it is applicable with scrap value. After executing all the steps of algorithm as stated in section 4 the cost of the finished product is found 2103.78 in 45 iterations. The final operation sequences with selected machine, tool, TAD, and scrap are shown in table 11.

Table 11. Operation Sequence Obtained by Proposed Heuristic (Tabu+ SSA+ FLC) for Case Study II

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP-ID	11	10	9	12	13	14	15	1	2	3	4	5	6	7	8
MID	1	1	1	3	2	1	1	1	1	1	2	2	1	1	2
TID	1	1	1	6	7	8	9	1	1	5	7	8	1	1	2
TAD	-X	+Y	+Y	+Z	+Z	+Z	+Z	-X	+Y	+Z	-Z	+Z	+Y	+Y	+Y
SCRAP (%)	2	2	2	2	8	5	2	2	2	5	5	10	10	2	5
The Total Cost Of Finished Product= 2103.78															
Number of Tool Changes = 10															
Number of machine change = 06															
Number of TAD changes = 08															

The performance of other heuristics is also experienced for the same dataset and the cost obtained by SA is 3311.45 with 76 numbers of iterations and its process plan is shown in table 12. Whereas, Sample SSA gives better results than SA that is 3165.84 and it converges quickly and requires only 68 iterations. The process plan for SSA is shown in table 13. When the adaptability of SSA is enhanced by embedding an FLC, the result is improved and obtained cost of finished product is 2457.20 and its process plan is represented in table 14. The number of iterations which it required is 62. When two search techniques (Tabu and SSA) are hybrid, the performance is increased significantly. The cost of finished product, obtained from it, is 2933.45 and this result has been achieved in 53 iterations. The process plan for the hybrid Tabu-SSA is given in table 15.

Table 12. Operation Sequence Obtained by SSA with FLC for Case Study II

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP-ID	11	12	1	2	3	4	5	13	14	15	10	9	6	7	8
MID	2	2	2	1	3	1	2	2	2	1	1	1	1	1	2
TID	2	6	2	1	5	7	8	7	8	9	1	1	1	1	2
TAD	-X	-Z	-X	+Y	-Z	-Z	+Z	-Z	+Z	-Z	+Y	+Y	+Y	+Y	+Y
SCRAP (%)	5	5	5	2	5	5	10	8	8	2	2	2	10	2	5
The Total Cost Of Finished Product= 2457.20															
Number of Tool Changes = 11															
Number of machine change = 06															
Number of TAD changes = 09															

Table 13. Operation Sequence Obtained by Tabu and SSA for Case Study II

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP-ID	1	11	12	13	14	15	10	2	6	9	7	8	3	4	5
MID	1	2	2	2	1	3	1	1	1	1	1	2	2	1	3
TID	1	2	6	7	8	9	1	1	1	1	1	2	5	7	8
TAD	-X	-X	-Z	-Z	+Z	+Z	+Y	+Y	+Y	+Y	+Y	+Y	-Z	-Z	+Z
SCRAP (%)	2	5	5	8	5	5	2	2	10	2	2	5	5	5	2
The Total Cost Of Finished Product= 2933.45															
Number of Tool Changes = 10															
Number of machine change = 07															
Number of TAD changes = 05															

Tabl3 14. Operation Sequence Obtained by SSA for Case Study II

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP-ID	11	12	13	14	15	10	9	1	2	6	7	8	3	4	5
MID	2	1	1	1	3	1	1	2	1	1	1	1	3	1	3
TID	2	6	7	8	9	1	1	2	1	1	1	3	5	7	8
TAD	-X	+Z	+Z	-Z	-Z	+Y	+Y	-X	+Y	+Y	+Y	+Y	+Z	-Z	-Z
SCRAP (%)	5	5	5	5	5	2	2	5	2	10	2	2	2	5	2
The Total Cost Of Finished Product= 3165.84															
Number of Tool Changes = 11															
Number of machine change = 08															
Number of TAD changes = 07															

Table 15. Operation Sequence Obtained by SA for Case Study II

	OPERATION SEQUENCE														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP-ID	1	11	10	9	12	13	14	15	2	3	4	5	6	7	8
MID	1	1	1	1	2	4	3	3	1	2	4	2	1	1	1
TID	1	1	1	1	6	7	8	9	1	5	7	8	1	1	3
TAD	-X	-X	+Y	+Y	-Z	+Z	+Z	-Z	+Y	-Z	-Z	+Z	+Y	+Y	+Y
SCRAP (%)	2	2	2	2	5	2	2	5	2	5	2	10	10	2	2
The Total Cost Of Finished Product= 3311.45															
Number of Tool Changes = 10															
Number of machine change = 08															
Number of TAD changes = 08															

A comprehensive analysis of the proposed approach, both in terms of objective function and their convergence trend for this case study has been carried out and are delineated in following subsections. The performances of the TSSA with FLC are finally compared with those obtained by the standard SA, SSA and SSA with FLC and Tabu-SSA approaches. Table 16 summarizes this comparison, highlighting the performance improvements offered by the proposed heuristic.

Table 16. Comparison among other Heuristic and Proposed Heuristic for case study II

S. No.	Heuristics	Cost	Number of Iterations
1.	**Tabu+ SSA+ FLC**	**2103.78**	**45**
2.	SSA+ FLC	2457.20	62
3.	Tabu+ SSA	2933.45	53
4.	SSA	3165.84	68
5.	SA	3311.45	76

Figure 8 compares the evolution of the best solutions of proposed algorithm (Tabu-SSA with FLC) with other existing heuristics. From this comparison, it is evident that the performance of simple SA is not very fine in comparison to other heuristics. SSA converges in less iteration than SA and when FLC is incorporated with SSA, it becomes faster. The convergence rate of hybrid Tabu-SSA is faster but the result is not improved than SSA with FLC. However, it is also marked that the proposed algorithm (Tabu + SSA with FLC) is significantly faster in reaching satisfactory solutions.

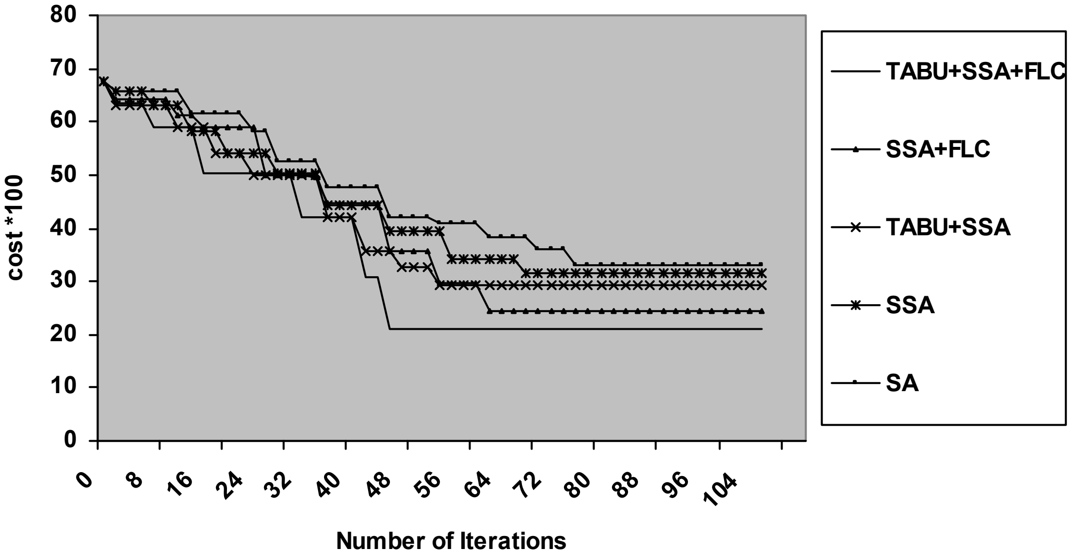

Figure 8: Comparison among the proposed heuristic and other heuristics for case study2

To show the robustness of the proposed approach, 10 randomly generated process plan selection problems with increased complexity have been considered. Comparative studies of the results for these problems are shown in table 17. It is evident from the table that for each case, proposed heuristic outperforms over other existing approaches except two, where result obtained in both the cases are same as that of Tabu-SSA.

Table 17. Results obtained using proposed TSSA heuristic with FLC and other heuristic for 10 problems

Sr. No.	No. of Features	No. of operations	No. of Machines	No. of Tools	Tabu-SSA with FLC	Tabu-SSA	SSA with FLC	SSA	SA
1.	6	11	3	6	2078.35	2478.53	2347.42	2582.82	2847.36
2.	8	12	3	6	2256.32	2675.02	2256.32	2773.61	3008.68
3.	10	17	4	8	2376.32	2832.17	2581.38	2832.35	3136.19
4.	12	19	4	8	2668.86	3258.73	2668.86	3426.18	3729.83
5.	14	20	5	8	2931.89	3397.08	3048.59	3569.87	3896.44
6.	16	23	5	9	3243.61	3679.39	3452.43	3822.91	4097.28
7.	18	27	5	9	3647.22	3803.27	3803.27	4176.52	4418.06
8.	20	28	7	9	3821.31	4311.94	4061.06	4603.25	4957.62
9.	22	31	7	11	4152.18	4862.37	4581.78	4933.47	5224.59
10.	24	37	7	11	4206.53	4902.61	4773.62	5154.16	5593.77

The proposed heuristic (TSSA with FLC) has been coded in C++ programming language and the experiment has been carried out on an IBM PC with a Pentium IV CPU -1.9 GHz processor. To sum up, for all the aforementioned results not only authenticate the supremacy of the proposed algorithm over existing heuristics but provide also a new dimension to the solution of complex combinatorial problems in real time.

7. Conclusion

The main contribution of this paper is to develop an efficient and consistent algorithm taking ideas from the existing ones for solving a Computer Aided Process Planning (CAPP) problem in randomized CIM environment. In the proposed model, an inevitable but neglected factor known as scrap or rejection has been accounted since it affects the overall cost of finished product in adverse manner. The main objective of this paper is to minimize the output unit cost by considering precedence relationships, availability of machines, tools, TAD and scrap. It can be concluded from the prominent literature that CAPP problem is a NP-hard problem and mathematically complicated to solve. Encouraged by the successful implementation of random search techniques to tackle such complex combinatorial problem, a hybrid search algorithm, Tabu-Sample Sort Annealing (TSSA) has been examined to solve the intricacies of process planning problem. The features extracted from Tabu and SSA have been associated in the proposed meta-heuristic and overcome the shortcomings of each algorithm. A fuzzy logic controller has been incorporated with it to provide the adaptability in swapping rate of the proposed algorithm. The transition probability based on Cauchy function provides more opportunity to escape from the local minima, and converges quickly toward the optimal/ near optimal solution. This hybrid evolutionary algorithm simultaneously satisfies the several goals viz. minimize the output unit cost, processing cost, generated scrap, and maximize the number of output units. First, a mathematical model has been formulated by accounting the various technological constraints. Thereafter, a feasible process plan has been generated by implementing the TSSA algorithm. The proposed algorithm is examined over several datasets with increasing complexity. Extensive computational experiments reveal the robustness and supremacy of the proposed algorithm.

The research has enough scope for extension. In future more objective functions such as inventory cost function etc. will be taken into account., and solved by other nature inspired algorithm like particle swarm optimization, ant colony optimization etc.

REFERENCES

1. ANDERSON, E.J., and FERRIS, M.C, **Genetic algorithm for combinatorial optimisation: assembly line balancing problem**, ORSA Journal of computing, **6,** pp.161-173, 1994.
2. BEN-ARIEH, D., and CHOPRA, M., **Case based NC code generating system for prismatic parts**, International Journal of Production Research, **35**, pp. 1925-1943, 1997.
3. BEN-ARIEH, D., GUTIN, G., PENN, M., YEO, A., ZVEROVITCH, **A Process planning for rotational parts using the generalized traveling salesman problem**, International Journal of Production Researc*h*, **41(11)**, pp. 2581-2596, 2003.
4. CASSOTTO, A., ROMEO, F., SANGIOVANNI-VINCENTELLI, A., **A parallel simulated annealing algorithm for the placement of macro cells**, IEEE Transaction Computer-Aided Design, vol. CAD-6, no. Sep., pp. 838-847, 1987.
5. CHAN, F. T. S., BHAGWAT, R. and WADHWA, S, **Increase in flexibility: productive or counterproductive? A study on the physical and operating characteristics of a flexible manufacturing system**, International Journal of Production Research*,* **44(7)**, pp. 1431–1445 2006.
6. GLOVER F., **Tabu search: a tutorial**, **20(4)**, pp. 74-94, 1990.
7. HAJEK, B., **Cooling schedules for optimal annealing, mathematical operation research**, **13 (2)**, pp. 311-329, 1988.
8. HUSBANDS, P., and MILL, F, **Simulated co-evolution as the mechanism for emergent planning and scheduling**, Proceedings of the Forth International Conference on Genetic Algorithms*,* Belew, R.K., Booker, I.B., (eds.), MORGAN Kaufmann Publishers, San Mateo, California, pp. 264-269, 1991.
9. JONES, A., RABELO, L., YIH, Y, **A hybrid approach for real time sequencing and scheduling**, International Journal of Computer Integrated Manufacturing, **8**(2), pp. 145-154, 1995.
10. KIM, K.W., GEN, M., YAMAJAKI, G., **Hybrid genetic algorithm with fizzy logic for resource constrained project scheduling**, Applied Soft Computing, **2/3 F**, pp. 174-188, 2003.

11. KIRKPATRICK, S., GELATT Jr., C.D., and VECCHI, M.P, **Optimization by simulated annealing**, Science, **220 (4598)**, pp. 671-680, 1983.

12. KUMAR, S., PRAKASH., KUMAR, S., TIWARI, M.K., and SANKAR, R., **Solving part type selection and operation allocation problem in a FMS : An approach using constrasnts based fast simulated annealing algorithm**, IEEE Systems Man and Cybernetics, pp. 404-408, Article In Press, 2005.

13. KUSIAK, A., and FINKE, G, **Selection of process plans in an automated manufacturing system**, IEEE Journal of Robotics and Automation, 1998, **4(4)**, pp. 397-402., 1998.

14. LI, L., FUH, J.Y.H., ZHANG, Y.F., and NEE, A.Y.C., **Application of Genetic algorithm to computer aided process planning in distributed manufacturing environment**, Robotics and Computer Integrated Manufacturing, article in press 2005.

15. MA, G.H., ZHANG, Y.F., NEE, A.Y.C., **A simulated annealing based optimization algorithm for process planning**, International Journal of Production Research, **38(12),** pp. 2371-2387, 2000.

16. MAMDANI, E.H., **Applications of fuzzy algorithms for simple dynamic plant**, IEE Proceedings, **121**, pp. 1585-1588, 1974.

17. MAMDANI, E.H., and GAINS, B.R., Fuzzy reasoning and its applications, (New York: Academic Press), 1981.

18. METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A., and TELLER, E., **Equation of state calculation by fast computing machines**, Journal of Chem. Phys., **21**, pp. 1087-1092, 1953.

19. MING, X.G. and MAK, K.L., **Intelligent set up planning by neural network based approach**, Journal of Intelligent Manufacturing, **11**, pp. 311-331, 2000a.

20. MING, X.G., and MAK, K.L., **A hybrid Hopfield Network-Genetic approach to optimal process plan selection**, International Journal of Production Research, **38(8)**, pp. 1823-1838, 2000b.

21. MISHRA, N., PRAKASH, TIWARI, M.K., SHANKAR, R., and CHAN, F.T.S., **Hybrid tabu-simulated annealing based approach to solve multi constraint product mix decision problem**, Expert System with Application, **26**, pp. 446-454, 2005.

22. MORAD, N., and ZALZALA, A.M.S., **Genetic algorithms in integrated process planning and scheduling**, Journal of Intelligent Manufacturing, **10**, pp. 169-179, 1999.

23. NIEBEL, B.W., **Mechanized process selection for planning new designs**, ASME papers, **737,** 1965.

24. PALMER, G.J., **A simulated annealing approach to integrated production scheduling**, Journal of Intelligent Manufacturing, **7(3)**, pp. 163-176, 1996.

25. RAI, R., KAMESHWARAN, S., TIWARI, M.K., **Machine tool selection and operation allocation in FMS: solving a fuzzy goal programming model using a genetic algorithm,** International Journal of Production Research, **40(3)**, pp. 641-665, 2002.

26. SU, C.T. and SHIUE, Y.-R., Intelligent scheduling controller for shop floor control systems: a hybrid genetic algorithm/decision tree learning approach, International Journal of Production Research, **41**(2), pp. 2619-2641, 2003.

27. SWARNKAR, R., and TIWARI, M.K., **Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing based heuristic approach**, Robotics and Computer Integrated Manufacturing, **20**, pp. 199-209, 2004.

28. THOMPSON, D., and BILBRO, G. L., **Sample sort simulated annealing**, IEEE Transactions on system, man, and cybernetics- part B: cybernetics, **35 (3),** pp. 625-632, 2005.

29. TIWARI, M.K., VIDYARTHI, N.K., **An integrated approach to solving the process plan selection problem in an automated manufacturing system**, International Journal of Production Research, **36(8)**, pp. 2167-2184, 1998.

30. WADHWA, S. AND BROWNE, J., **Modeling FMS with decision Petri nets**, International Journal of Flexible Manufacturing System, **1**, pp. 253–280, 1989.

31. WADHWA, S., AGGARWAL A**, Synergism Of Flexibility, Integration and Automation in Computer Integrated Manufacturing (CIM) Systems**, Studies in Informatics and Control, 2000.

32. WADHWA, S., RAO, K. S. and CHAN, F. T. S., **Flexibility-enabled lead-time reduction in flexible systems**, International Journal of Production Research, **43 (15)**, pp. 3131–3162, 2005.

33. WITTE, E.E., CHAMBERLAIN, R.D., and FRANKLIN, M.A., **Parallel simulated annealing using speculative computation**, IEEE Transaction parallel distributed system, **2 (4),** pp**.** 483-494, 1991.

34. ZADEH, L.A., **Outline of a new approach to the analysis of complex systems and decision processes,** IEEE Transactions on System, Man, and Cybernetics, **SMC-3**, pp. 28-44, 1973.

35. ZADEH, L.A., The concept of a linguistic variable and its application to approximate reasoning-I & II, Information Sciences, **8**, pp. 199-249, 1975.

36. ZADEH, L.A., **The concept of a linguistic variable and its application to approximate reasoning-III,** Information Sciences, **9**, pp. 43-80, 1976.

37. ZADEH, L.A., **PRUF- A meaning representation language for natural languages**, International Journal of Man-Machine Studies, **10**, pp. 395-46, 1978.

38. ZHANG, F., ZHANG, Y. F., NEE, A.Y.C., **Using genetic algorithm in process planning for job shop machining**, IEEE Transaction on Evolutionary Computation, **1(4)**, pp. 278-289, 1997.

Appendix A

It has been stated that two conditions are essentially satisfied by the transition probability that is known as: Nonnegativity and Normalization (Haykins 1999). Both of these are also verified for Boltzman function. In the following subsections, these properties are quantitatively demonstrated and validated for the Cauchy function:

Non-negativity

Let us assume that for a random variable R_a, which represents an arbitrary Markov chain, is presently in state x_i at time n and it's having the energy E_i. After some processing or time the state is shifted from x_i to x_j and it represents also a new random variable R_n which having the energy E_j. The difference of energy between state I and j can be represented by ΔE. If the energy difference (ΔE) is negative, the transition leads to that state which having the lower energy, and the transition is accepted. Transition probability can be calculated by the following equation:

$$TP = \frac{T(K)}{T^2(K) + (\Delta f)^2} \tag{32}$$

Where, $TP_{ij} = P\left(R_n = x_j / R_a = x_i\right) = P\left(R_n = x_i / R_a = x_j\right)$ (33)

Since, $T(k) > 0, \mathrm{T}^2(k) > 0, \text{and } (\Delta \mathrm{E})^2 > 0$ (34)

Hence, $TP_{ij} > 0$ for all values of I, j. so the condition of Nonnegativity has been satisfied for the transition probability by Cauchy function,

Normalization

The condition of normalization is satisfied if the following condition becomes true:

$$\sum_j TP_{ij} = 1 \qquad \forall \mathrm{i} \tag{35}$$

For the Cauchy distribution function, the probability density function can be represented as following:

$$f(x) = TP_{ij} = \frac{T(K)}{T^2(K) + (\Delta f)^2} \tag{36}$$

By the property of cumulative density function, it can be stated that:

$F(\infty) = P\{x \le \infty\} = 1$ Since $\{x \le \infty\}$ is a secure event.

Then

$$\int_{-\infty}^{\infty} f(x) dx = F(\infty) = 1 \tag{37}$$

The above equation shows and proves the normalization property of Cauchy function.

Cauchy function also follows the principle of detailed balance. This principle can be delineated in the following subsection:

Principle of Detailed Balance

Let an irreducible Markov chain started from a recurrent state i at time n=0. the time elapse between (k-1)th and kth return for the state i can be represented as $T_i(k)$ then the steady state probability of state can be described as follows:

$$\pi_i = \frac{1}{E(T_i(k))} \tag{8}$$

Where, $E(T_i(k))$= mean recurrence time of state i.

Suppose a stochastic matrix $P = TP_{ij}$ of Ergodic Markov chain, which is having states x_1, x_2, x_3…..x_k, be irreducible. This chain has a unique stationary distribution to which it converges from any initial state. For the aforementioned unique set, the essential conditions are as following:

$$\lim_{n \to \infty} \left(TP_{ij}\right)^n = \pi_j \qquad \forall\, i \tag{39}$$

$$\pi_j > 0 \qquad \forall\, j \tag{40}$$

$$\sum_{j=1}^{k} \pi_j = 1 \tag{41}$$

$$\pi_j = \sum_{i=1}^{k} \pi_i TP_{ij} \qquad \text{for } j = 1, 2, 3, \ldots, k \tag{42}$$

The principle of detailed balance states that at the thermal equilibrium the rate of the occurrence of the inverse transition is

$$\pi_i P_{ij} = \pi_j P_{ji} \tag{43}$$

Where, P_{ij} =probability transition from state I to state j.

When the state x_i shifted to state x_j, the transition probability follows some conditions that are given below:

$$P_{ij} = \begin{cases} PT_{ij}\left(\dfrac{\pi_j}{\pi_i}\right) & \text{for} \left(\dfrac{\pi_j}{\pi_i}\right) < 1 \\ PT_{ij} & \text{for} \left(\dfrac{\pi_j}{\pi_i}\right) \geq 1 \end{cases} \tag{44}$$

Where, P_{ij} = transition probability from state I to stat j

In the present paper, the transition probability exist only when ΔE<0. In this condition, energy difference is negative on changing from state x_i to x_j, therefore, $\dfrac{\pi_j}{\pi_i} \geq 1$ (Kumar et al. 2003). From the above equations:

$$\pi_i P_{ij} = \pi_j PT_{ij} = \pi_j PT_{ji} \tag{45}$$

The above equation shows the detailed balance nature of Cauchy function.