

Selection of Best Alternative Process Plan in Automated Manufacturing Environment: An Approach Based on Particle Swarm Optimization

F.T.S. Chan¹, M.K. Tiwari² and Y. Dashora²

¹*Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong*

²*Department of Industrial Engineering and Management, Indian Institute of Technology, Kharagpur*

¹*Hong Kong, ²India*

1. Introduction

In the present flexible and automated manufacturing environment, selection of optimal process plan is a crucial decision making problem. The systematic determination of processing steps for the transformation of raw material to its finished product is identified as process planning. The real world dynamic shop floor is characterized by the availability of several machines, tools, fixtures/jigs etc., and demands the completion of several design tasks before the commencement of manufacturing actual manufacturing of a part type. Different geometrical and tolerance relationships among several features of the part types necessitate the arrangement of different setups to carry out various and hence, diverse alternative process plans to manufacture a part come into existence. Any of these feasible process plans can be used to produce the particular part type from its raw material [1], [2]. Due to the incorporation of dynamic shop floor situations such as bottleneck machines, non availability of tools, machine breakdown, etc., the process plan selection problem becomes non linear and NP hard in nature. The proliferation of Computer Aided Process Planning (CAPP) systems has made it easy and more efficient to tackle these types of non linear process planning systems. The scheduling complexity in the manufacturing systems was discussed in [2] and it was proposed that this can be reduced with the limited number of tools and auxiliary devices. The three reasons given by [2] to solve the process plan selection problem are: production cost, tool magazine capacity limitation, and reduction of auxiliary devices. Later, the process plan selection problem was attempted in [1] considering three objectives such as to minimize total time, minimize number of setups and to minimize dissimilarity among process plans. Reference [3] contributed in solving process plan selection problem using fuzzy approach to deal with the imprecise information. Reference [4] incorporated the factors such as similarity index within a process plan and degree of similarity among various process plans. They used fuzzy approach to take care of the part type processing sequence. PPS problem has also been attempted using Hybrid Hopfield Neural network and Genetic Algorithm Approach [13].

Open Access Database www.i-techonline.com

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

However, in this paper an attempt has been made to solve the PPS problem by giving a more rational view to part type processing sequence. Here, in addition to fuzzy membership vector a new feature called Similarity Attribute (λ) has been introduced that takes care of the similarity among different part types. Based on the consolidated approach incorporating fuzzy membership vector and similarity attribute, the part type processing sequence is evaluated. To ease the solution strategy, the undertaken PPS problem is modeled as a Traveling Salesman Problem (TSP) that helps to do away the problem complexity and ensures the easy application of various Artificial intelligence (AI) tools. The PPS problem is mapped as a TSP considering the distance of the tour in the terms of the objective function. Due to its NP-hard nature [5] and wide range applicability, TSP has been one of the most studied combinatorial optimization problem. This paper proposes a new Intelligent Particle Swarm Optimization algorithm with the modified concept of Local Repeller (IPSO-LR) to solve the aforementioned PPS problem.

Particle Swarm Optimization (PSO) is a new population based evolutionary computation technique that proceed via self adaptive search. In general, the evolutionary algorithms are based on population of individuals simulating some biological phenomenon. Particle Swarm Optimization is one of the recent developments of evolutionary systems first introduced by Kennedy and Eberhart in 1995 [6]. Unlike other evolutionary systems, no direct recombination of genetic material is incorporated in PSO while the search is in progress. The most important and distinctive feature of PSO is its working that is based on social behavior of particles or individuals in the swarm. The algorithm develops the search strategy by adjusting the trajectory of each particle towards own previous best location and best position of neighboring particles within the search space. Since its introduction, PSO has been tested invariably on several computationally complex NP hard problems [7]. The recent challenges are to employ the algorithm to the real world problems of various complexities than those on which initial versions of it have been applied. Most of the recent developments in the PSO are based on improving its ability to come out of local optima, as it is recognized as common problem encountered by swarms. In this paper, a new improved swarm algorithm is used that has enhanced capability to come out of local minima.

The application of IPSO-LR algorithm has been demonstrated considering one illustrative example. To assess the robustness of IPSO-LR based solution strategy, five well known test parts from the literature have been considered and five new parts have been developed. Rest of the paper has been organized as follows: The Process Plan Selection problem and its TSP formulation have been discussed in section 2. Section 3 gives an overview of PSO and details IPSO-LR algorithm. Section 4 illustrates the application of IPSO-LR to solve PPS problem with the help of an illustrative example. Computational experiments and the discussion of the results are provided in section 5. Section 6 concludes the paper.

2. Problem Environment

The PPS problem is concerned to the problem of making optimal choices among several alternatives and is featured by the selection of machines, cutting tools, fixtures, setups, etc. The problem is to select exactly one process plan for each part type from a number of accessible and feasible process plans and to provide optimal processing sequence for the manufacturing of part types. The problem formulation adopted in this paper is the extended and modified version of the formulation proposed by [4] and [8]. This paper aims to select a process plan for each part type, keeping in view the wider range of objectives as

minimization of batch size, time remaining from due dates and number of machinable features, as well as calculating the part type processing sequence, that determine the processing cost and optimum utilization of resources, in a much more justifiable way. The various parameters involved in the process plan selection (PPS) problem can be summarized as:

1. A seven digit code to denote the machines, operations, tools, fixtures, etc., for each step of a process plan.
2. Material handling time for a process plan and the machining time on different machines for a process plan.
3. Batch size, due dates remaining and other manufacturing related features of a part type.

Minimization of batch size, total time remaining from due dates, number of machinable features and process plan execution time, as well as calculating part type processing sequence along with optimum utilization of resources are considered as main objectives targeted in the paper. To pursue these objectives, an integrated objective function is formulated that incorporates the parameters defined in [4] along with addition of a new parameter λ . These parameters warrants due attention because of their immense impact on the solution strategy and objective function formulation, and can be summed up as follows:

1. Similarity Index (SI) of a process plan of a part type.
2. Degree of Similarity (DS) among various process plans of a part type.
3. Membership vector (μ) of a part type.
4. Similarity attribute (λ) of different part types.

The details about the calculations and authenticity of first three parameters can be referred in [4]. This paper adds a new dimension to the solution of the PPS problem by the incorporation of a new parameter termed as Similarity Attribute. The formulation proposed by [4] did not take into account the similarity among the processing of part types that is a crucial parameter affecting the dynamics and cost efficiency of the shop floor. Hence, Similarity attribute (λ) has been incorporated in the objective function to make it more authentic. Its calculation strategy has been provided in section 4, where solution strategy for the underlying problem has been illustrated. Minimization of the aforementioned first two parameters provides cost efficiency and the values of rest two determine the part type processing sequence. Thus, to accomplish objectives highlighted in this paper, they have been integrated into single sub objective that is a trade-off between all the aforementioned features (detailed in section 3). To develop the solution strategy for the PPS problem, it is formulated as a Traveling Salesman Problem (TSP). It's basic TSP formulation is described in the following discussion:

2.1 TSP Formulation of the PPS Problem

In a TSP with one salesman, the salesman has to visit each city in his/her designated area and then come back to the home town [5]. Here, each process plan is considered as a city (*i.e.* a node) and a salesman is restricted to move through only one node among the nodes characterizing a part type. A tour is considered to be complete when the particle has moved through a node of each part type. In the TSP model of the problem, the value of objective function represents the total distance covered by the salesman in a tour. The criterion to move from one node to another depends upon the solution strategy; in the context of the PPS problem, it is based on the probability to choose, thus, not based on the integer model.

Hence the formulation can be considered as TSP with Mixed Integer Programming (MIP). The details are provided in the next section that provides an insight to the basics of PSO as well as details IPSO-LR algorithm.

3. Proposed IPSO-LR Algorithm

PSO belongs to a broad class of population based optimization technique that is guided by the social behavior of flocking organisms, like birds, honeybees, etc. The fundamental rules adhered by the individuals comprising a flock may be outlined as to match velocities with nearest neighbors, and to be closer with the others in the swarm. Thus mutation with conscience has been claimed for PSO [9]-[12]. In this case, each particle tends to accelerate towards its own previous best position and towards the best position of neighbor particles encountered, with the usual result being clustering of individuals in optimal regions of space. Since the advent of PSO, the challenge has been to apply PSO to the problems of various domains. In this paper, a new Intelligent Particle Swarm Optimization Algorithm (IPSO-LR) with the modified concept of local repeller has been developed to efficiently model the problem in the algorithmic context as well as to avoid the problem of entrapment in local optima.

At each position, the velocity and position of each particle is being updated using some basic equations and rules. The velocity of particle at each position is updated utilizing the aforementioned characteristics and the relation detailed in the following subsection:

3.1 Velocity Evaluation

The model for velocity and position updating signifies the intelligence of the swarm and can be mathematically formulated as:

$$\forall i \in N, v_{i \text{ next}} = \chi [v_i + c_c \times \text{rand}(\cdot) \times (\Delta x_{ci}) + c_s \times \text{rand}(\cdot) \times (\Delta x_{ni})] \quad (1)$$

where, v_i is the current velocity of the particle; N is the number of particles; χ is the constriction coefficient, and is mathematically expressed as:

$$\chi = \frac{2\kappa}{[2 - \beta - \sqrt{\beta^2 - 4\beta}]} \quad (2)$$

$$\text{s.t.} \quad \beta = c_1 + c_2, \quad \beta > 4, \quad \kappa \in]0, 1]$$

Further, $\text{rand}(\cdot)$ is a random function with a range $[0, 1]$; c_c and c_s are positive constant parameters, called acceleration coefficients (which control the maximum step size the particle can do). c_c and c_s controls the impact of previous values of particle positions and velocities on its current one. Suitable selection of acceleration coefficients can provide a balance between the global and the local search. The constriction factor χ helps to ensure convergence [9], whereas the factors such as c_c and c_s along with $\text{rand}(\cdot)$ guarantee the thorough search in the region near to o_i and n_i . Different configurations of χ as well as their theoretical analysis can be found in [9].

In the velocity relation, Δx_{ci} and Δx_{ni} are self best positional differences and neighborhood best positional difference. In the equation (2), Δx_{ci} and Δx_{ni} are calculated by the following relations:

$$\Delta x_{ci} = o_i - x_{ni}; \text{ and } \Delta x_{ni} = n_i - x_{ni}, \quad (3)$$

where,

o_i : Position of previous best position of particle.

x_{ni} : Position of n^{th} feasible node. Here, $n \in N_{fi}$ (i.e. set denoting feasible nodes to move, for particle i .)

n_i : Previous best position of neighboring particles.

In the above discussion, the position of a particle is characterized by the set of variables characterizing a node. The velocity of i^{th} particle to each feasible node is calculated as per the aforementioned equation that is followed by the position updating according to the relation:

$$\forall i \in N, x_i = x_{i \text{ vmax } f}, \quad (4)$$

where, x_i denotes the position of the particle; $x_{i \text{ vmax } f}$ is the position of the node for which the velocity found is maximum. The self previous best position of each particle is updated using the following relation:

$$\forall i \in N, o_i = \begin{cases} o_i & \text{if } f(x_i) \geq f(o_i) \\ x_i & \text{if } f(x_i) < f(o_i) \end{cases} \quad (5)$$

where, $f(x_i)$ denotes the respective objective function value considered in the problem. The previous best position of neighboring particles is updated according to the following relation:

$$\forall i \in N, n_i = \min_{o_i} f(o_i) \quad (6)$$

where, N_i is the set denoting neighbors of particle i .

to reduce the probability of leaving the search space, the velocity of particles is restricted to the range of $[-V_{max}, +V_{max}]$, where,

$$V_{max} = v \times x_{max}; \quad 0.1 \leq v \leq 10 \quad (7)$$

3.2 Sociometry of IPSO-LR

Neighborhood is the most decisive criterion that directs the search procedure of swarms. It signifies how the movement a particle is influenced by the information carried by the other particles. The neighborhood is exploited for the mutual sharing of crucial information among particles that helps them in further movement and diversify search technique. The topological structure of population controls its propensity of exploration versus exploitation [11]. The initial versions of particle swarm select a particle from the specified neighbors as a source of influence and ignore others. This type of strategy only provides a choice of

choosing a particle from the neighborhood; the more is its size, the more is the likeliness of choosing the better one.

In this paper, a cluster type of network topology is adopted as it produces promising results as compared to that of other neighborhood topologies like ring, all, pyramid, triangular, frame, etc. [12]. In the proposed strategy, various process plans of a part type are in neighborhood with each other as they characterize same attributes of a part type. Thus, various process plans (particles) of a part type form a cluster that shares information among the members. In this case, the number of clusters formed is equal to the number of process plans. As evident from Figure 1, each cluster is in further interaction with other clusters through the arcs joining the two closest nodes of each pair of clusters.

3.3 Modified Strategy to Avoid Local Optima

Entrapment in the local optima is the situation where the algorithm sticks to some premature solutions and does not show any improvement. To alleviate this problem, the concept of local repeller [12] with some modifications to suit the problem structure has been utilized. The most alluring trait of this technique is its simplicity and efficacy to avoid local optima. As and when the path corresponding to local optima is encountered, the sequence of process plans that is identified to be constituent of local optimum is made 'repelling' *i.e.* the particles are compelled to explore the search space more thoroughly and hence, the search is directed towards global optimum. This strategy guarantees the escape from the local optima and thus is very effective.

4. Implementation of IPSO-LR Algorithm on the PPS Problem (Illustrative Example)

4.1 Problem Characteristics

This paper adopts the formulation of PPS problem from [4] and [8]. To denote the machines, operations, tools, fixtures etc. a seven-digit code has been used. The data related to alternative process plans, processing time on different machines, material handling time for the different process plans, the batch size, due dates remaining and features of each part type are adopted from [8].

In the undertaken problem, the objectives considered are minimization of batch size, time remaining from due dates and number of machinable features, as well as calculating part type processing sequence along with optimum utilization of resources. The objectives like maximization of batch size, minimization of time remaining from due dates and minimization of number of machinable features are incorporated in the definition of membership vector μ [4]. The parameter Similarity attribute (λ) quantifies the similarity among the part types that is based on the number of machines, fixtures, tools and operations performed to produce these. Its formulation is given as follows:

$$\lambda_{im} = \frac{C_{im}}{C_m}, \quad \lambda_{io} = \frac{C_{io}}{C_o}, \quad \lambda_i = \frac{C_{it}}{C_t}, \quad \lambda_{if} = \frac{C_{if}}{C_f}; \Rightarrow \lambda_i = \frac{\lambda_{im} + \lambda_i + \lambda_{im} + \lambda_{im}}{4}; \quad (8)$$

where, $\lambda_{im}, \lambda_i, \lambda_{im}, \lambda_{im}$ are the constants denoting contribution of attributes related to machines, operations, tools and fixtures, respectively, to the calculation of similarity

attribute; C_{im} , C_{io} , C_{it} , C_{if} are the cardinalities of the sets denoting number of machines, operations, tools and fixtures, respectively, utilized by the process plans of part type i ; C_m , C_o , C_t , C_f are the cardinalities of sets denoting the total number of machines, operations, tools and fixtures, respectively, used to manufacture all the part types from all the possible alternatives.

The parameter DS is a measure of accounting for the similarity among the several process plans of the different part types. It results from the comparison of their constituent elements, namely the operation codes. The parameter SI defines the similarity contained in a process plan itself. It denotes the ease with which a part can be made from the particular process plan. The details about the calculation of DS and SI can be referred from [4].

4.2 IPSO-LR Algorithm Based Solution Strategy

To initialize the process, all the particles are randomly distributed over the nodes. Here, the number of particles equals the number of nodes present in the TSP formulation. The most critical step in the application of IPSO-LR to solve the PPS problem is the characterization of the parameters that represents position. In the formulation used in the proposed paper, a node (*i.e. a process plan*) characterizes the position of the particle. In this case, the velocity to each feasible node j from the particle on node i is calculated as per the following equation:

$$v_{ij} = \chi [v_i + c_c \times \text{rand}(\cdot) \times (\Delta x_{cij}) + c_s \times \text{rand}(\cdot) \times (\Delta x_{nij})] \quad (9)$$

Here, Δx_{cij} and Δx_{nij} are the positional differences that needs to be defined in the problem context. In the scenario of PPS problem, these can be evaluated using the following relations:

$$\begin{aligned} \Delta x_{cij} &= A \times [1 - DS_{o_i, x_{ji}} \times (SI_{o_i} + SI_{x_{ji}}) \times \mu_{x_{ji}} + \lambda_{x_{ji}}] + B \times t \\ \Delta x_{nij} &= A \times [1 - DS_{n_i, x_{ji}} \times (SI_{n_i} + SI_{x_{ji}}) \times \mu_{x_{ji}} + \lambda_{x_{ji}}] + B \times t \end{aligned} \quad (10)$$

Having updated the velocity, the position of particle is updated as per equation 9. Another major difference in the application of IPSO-LR from the traditional PSO lies in the definition of previous best position of the particle, o_i , and neighbor's previous best position, n_i . Because of the TSP structure of the problem the particles cannot be always in the constant motion and hence, after completing the tour, particles are again randomly distributed over the nodes and set to move. The updated previous best position and neighborhood best position guide the particles in the consecutive generations to choose the better alternatives. The pseudo code for the IPSO-LR algorithm applied to the PPS problem is given below:

Create the initial population P and set $iter_{\max}$, and $iter_{LR\max}$ (*i.e.* number of iterations for which if solution is not improved, then local optimum is considered to be encountered)

$iter = 0$;

$iter_{LR} = 0$;

for each particle $i \in P$:

initialize the x_i , v_i , o_{ij} , n_{ij} , neighborhood N_i , global best position g_{ij} and value of overall

global best tour g_j (of all the particles). // Here, $o_{ij}, n_{ij}, g_{ij}, g_j$ are the arrays containing the values of respective positions in each part type.

```

repeat:
  for (j=1:N)                                // Here, N is the
                                              number of part types.
    repeat:
      for each particle  $i \in P$ :
        if  $f(x_{ij}) < f(o_{ij})$ 
          then  $o_{ij} = x_{ij}$ ;
        endif
        if  $f(o_{ij}) < f(g_{ij})$ 
          then  $g_{ij} = o_{ij}$ ;
        endif
      endfor
      if sum of the tour  $< (g_j)$ 
        then  $g_j =$  sum of the tour;
               $iter_{LR} = 0$ ;
      else  $iter_{LR} = iter_{LR} + 1$ ;
      endif
      if  $iter_{LR} = iter_{LRmax}$ 
        then mark the tour as repelling (i.e. any
              particle trying to complete this
              tour is randomly thrown away.
        Update  $x_i$  and  $v_i$  accordingly.
      endif
       $iter = iter + 1$ ;
    until  $iter = iter_{max}$ ;

```

5. Computational Experience

This section aims to provide the summary of numerical simulation of the proposed algorithm along with the comparative results with other established techniques from the literature in a condensed form. The number of alternative solutions increases exponentially as the number of part types and their alternative process plans increase. The complexity of the undertaken problem can be gauged by the fact that aforementioned 10 parts and their 52

alternative process plans give rise to a total of 1.28×10^{13} feasible solutions. By the application of IPSO-LR the best alternative process plans and their sequence obtained is listed in Table 1.

Application of the ACO strategy [8] also renders similar results. The applicability and efficacy of the proposed algorithm is evident from the fact that IPSO-LR outperforms other established techniques from the literature to solve the complex process plan selection problem with various formulations. In fact, due to its less computational complexity, the proposed IPSO-LR algorithm gains an edge over other techniques when the problems pertaining to real size data sets (like the undertaken data set) are concerned. The proposed algorithm is characterized by faster convergence along with the better and logical escape from the local optima.

Part type processing sequence	Part type	Process plan selected	Route of the plan
1	3	5	M011103,L020201,L070501,L080601,L120101,L100801,L050301
2	2	3	M011103,L020201,L030201,M151304,L080601
3	9	3	M011103, L020201,L080601,L090701,L110901
4	7	3	M011103, L02021,L030201,L080601,L070501,L060401,L120101
5	8	2	M011103, L020201,L030201,L060401,L070501,L080601
6	10	11	M011108,L020201,L030201,L040202,L080601,L090701,L110901
7	6	5	M011103, L020201,L040201,L060401,M100804
8	1	6	M011103,L020201,L030201,M100704
9	5	2	L010101,L020201,L030201,L040202,M151304,M100804
10	4	1	L010101,L020201,L060401,L120101,M171504

Table 1. Optimal Process Plan Selected

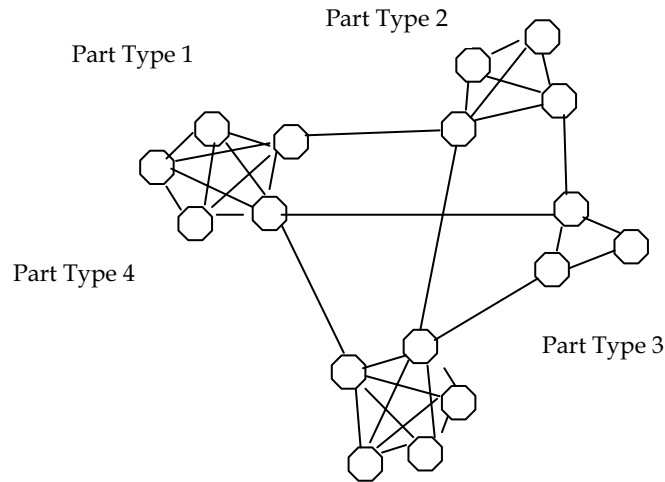


Figure 1. Graphical representation of PPS problem

Figure 2 provides a comparative plot between the fitness index of particles and the number of generations for the proposed strategy and ACO based strategy [8]. Here, the fitness index is defined as:

$$f_i = \frac{Obj_{best} - Obj_{worst}}{Obj_{worst}}; \quad (11)$$

Where, Obj_{best} and Obj_{worst} are the objective function values of the particles covering shortest tour and longest tour respectively.

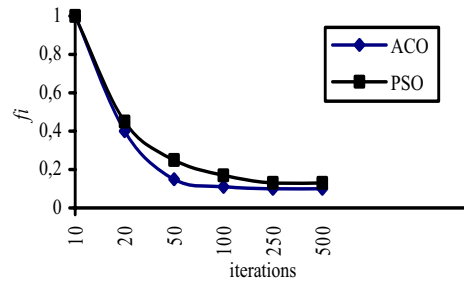


Figure 2. Comparative convergence with ACO [8]

The comparative convergence trend of the algorithm with ACO based approach proves the compatibility of the proposed algorithm, as shown in the Figure 2. Figure 3 plots the CPU time vs number of iterations. From the plot (Figure 2), it can be visualized that the value of fitness index decreases as the number of generations increase that in turn proves the clustering of particles around best solution. This clustering is further proved by the Figure 4 that provides the plot between the percentages of particles that deviates from the best particle by not more than 5%.

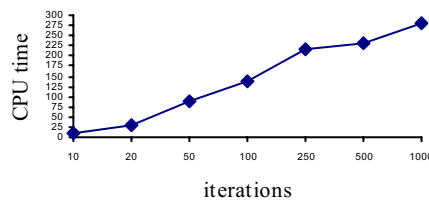


Figure 3. Variation of CPU time (ms) with number of iterations

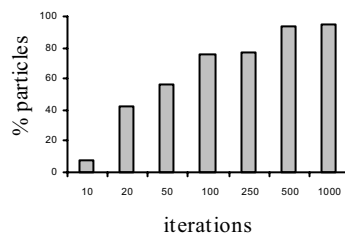


Figure 4. Percentage of deviating particles

In order to illustrate further the effectiveness of the proposed IPSO-LR algorithm, various problems taken from the literature [2], [3], [14] have been tested with the same formulation of objective functions and parameter as is proposed in them. As a matter of fact, the proposed approach obtains the best solutions for different process plan selection examples, the comparative results of which are provided in Table 2.

Applied methodology ↓	Example of [1]		Example of [2]		Example of [3]	
	Objective function value	Process plan selected	Objective function value	Process plan selected	Objective function value	Process plan selected
Reference [1]	33.1	1, 3, 7	30.8	1,4,7,9	61	3, 5, 7
Reference [2]	34.5	1, 3, 5	30.8	1,4,7,9	59	3, 4, 6
Reference [3]	32.5	1, 4, 5	30.8	1,4,7,9	61	3, 5, 6
ACO approach [8]	32.5	1, 4, 5	30.8	1,4,7,9	59	3, 4, 6
Proposed IPSO-LR approach	32.5	1, 4, 5	30.8	1,4,7,9	59	3, 4, 6

Table 2. Comparative results of various methodologies from the literature

In nutshell, the aforementioned computational results not only prove the efficacy and supremacy of the proposed strategy but also provide a new dimension to the solution of complex PPS problems in the practical environment.

6. Conclusive Remarks

Ever so changing competitive manufacturing structure is challenged by the issue to properly optimize resource allocation and their uses in order to get the best out of available alternatives. The PPS problem (amidst unpredictable disruptions observed in shop floor), is of substantial importance in flexible and automated manufacturing systems and needs much attention to be paid. The performance of flexible manufacturing systems is greatly influenced by the selection of viable and economic process plans among the other competing plans. This paper presents a new IPSO-LR algorithm to solve a complex real time PPS problem with the objectives like minimization of batch size, time remaining from due dates and number of machinable features, as well as calculating part type processing sequence along with optimum utilization of resources. The algorithm is characterized by the enhanced capability to come of local optima in a logical manner and has knack to handle the problems pertaining to large alternatives. The proposed work provides a new and broader dimension to the solution of PPS problem by consolidating a new parameter Similarity Attribute ' λ ', that formulates the objective function in a more justifiable way. The real strength of swarms is derived from the interaction among particles while exploring the search space collaboratively. The terms of positional difference introduced in the velocity formula leads the particle to be successful regarding reaching towards optima and guides it by the previous successes of itself and other particles.

This paper finds its contribution in the expanding area of research of intelligent automation in industries as well as in the broad field of interdependent evolutionary computation. The computational experience establishes the fact that the proposed algorithm is effective to model and solve PPS problems of varying complexities. Experimental results have shown the robustness of the algorithm and its outperforming behaviour over established techniques in the process planning field. Also, based on these results, the use of IPSO-LR algorithm seems to be encouraging in supporting the premise of automated and dynamic and intelligent process planning. Future work includes the development of web enabled intelligent Process Planning System with embedded features of e-Manufacturing and application of various tools and techniques related to Data Mining to refine the search algorithms.

7. References:

- Bhaskaran, K., 1990, Process plan selection .*International Journal of Production Research*, 28 (8), 1527-1539. [1]
- Kusiak, A., and Finke, G., 1988, Selection of process plans in automated manufacturing systems. *IEEE Transactions of Robotics and Automation*, 4(4), 397 - 402[2]
- Zhang, H.C., and Huang, S.H., 1994, A fuzzy approach to process plan selection . *International Journal of Production Research*, 32(6), 1265-1279. [3]
- Tiwari, M.K., and Vidyarthi, N.K., 1998, An integrated approach to solving the process plan selection problem in an automated manufacturing system, *International Journal of Production Research*, 36(8), 2167-2184. [4]
- Onwubolu, G. C. and Clerc, M., 2004. Optimal path for automated operations by a new heuristic approach using particle swarm optimization. *International Journal of Production and Research*, 42, 473-491. [5]
- J. Kennedy and R. C. Eberhart, Particle swarm optimization, in *Proc. IEEE Int. Conf. Neural Networks*, vol. 4, Perth, Australia, Dec. 1995, pp. 1942-1948. [6]
- Juang, C. F., 2004, A hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network design, *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 34(2), 997-1006. [7]
- Dashora, Y. ,Kumar, S., Shankar, R. and Tiwari, M. K., Ant Colony Optimization based approach to select the optimum Process Plan in Automated Manufacturing Environment, communicated to *IIE Transactions*, UIIE-0061. [8]
- Clerc, M. and Kennedy, J., 2002, The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space, *IEEE Transactions on Evolutionary Computation*, 6(1), 58-73. [9]
- Wachowiak, M. P., Smolíková, R., Zheng, Y., Zurada, J. M., Elmaghraby, A. S., 2004, An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, 8(3), 289-301. [10]
- Mendes, R., Kennedy, J. and Neves, J., 2004, The Fully Informed Particle Swarm: Simpler, Maybe Better, *IEEE Transactions on Evolutionary Computation*, 8(3), 204-210. [11]
- Parsopoulos, K. E. and Vrahatis, M. N., 2004, On the Computation of All Global Minimizers Through Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, 8(3), 211-224. [12]
- Ming, X.G., and Mak, K.L., 2000, A hybrid Hopfield network-genetic algorithm approach to optimal process plan selection, *International Journal of Production Research*, 38(8), 1823-1839. [13]