A CMPSO algorithm based approach to solve the multi-plant supply chain problem

Felix T. S. Chan[1], Vikas Kumar[2] and Nishikant Mishra[3]
[1] *The University of Hong Kong,* [2]*University of Exeter,* [3]*University of Warwick*
[1]*Hong Kong S.A.R.,* [2,3] *United Kingdom*

1. Introduction

In the era globalisation the emerging technologies are governing the manufacturing industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from the nature and implant it in the engineering sciences. This way of thinking led to emergence of many biologically inspired algorithms that have proven to be efficient in handling the computationally complex problems with great ease and competence such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms the present research proposes a new Cooperative Multiple Particle Swarm Optimization (CMPSO) algorithm. The idea behind this proposed CMPSO algorithm came from the limitations associated with the existing PSO algorithm under the discussed problem scenario. The proposed CMPSO algorithm has been applied in multi-plant supply chain environment which has proven to be NP hard problem. To prove the efficacy and robustness of the proposed CMPSO algorithm it has been compared with the existing evolutionary algorithms (EAs). Furthermore the authors have also shown the statistical validation of CMPSO algorithm.

The changing scenario of the global business urges efficient ways of performing various tasks to sustain the impact of market uncertainty. Especially manufacturing industries are looking for the competent ways of managing their processes. Supply chain being the backbone of any industrial organization demands its efficient management for the shake of profitability, and customer satisfaction point of view. Meeting delivery dates is an increasingly important objective in today’s competitive market, because delivery delays often result in a considerable loss of goodwill and eventually market share. Realizing this important contribution of the effective supply chain management, this research aims to optimize the efficiency of the supply chain by handling the complex task of planning and scheduling. In the context of supply chain management the integrated process planning and scheduling has a key role to play. In order to satisfy timeliness and cost criteria imposed by market competition, generally manufacturing industries are opting for *Multi-Plant Supply Chain* (MPSC). MPSC can be seen as a part of supply chain in which coordination, cooperation, and synchronization actions are deliberately strong and binding, so as to guarantee the accomplishment of the predefined objectives. More specifically, the MPSC

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

represents an extended integration ahead of a single manufacturing site by means of strong distribution management capability; electronic data interchange, and adequately coordinated multiple plant management to ensure the real motives of Computer Integrated Manufacturing (CIM). The intricacy of the problem under such conditions increases exponentially with the increase in number of operations, plants, and customers. The increasing applicability and popularity of the biologically inspired tools, in particular, the growing interest among the researchers in PSO techniques motivated us to use it in our present problem scenario. However, PSO in the original form could not be applied under multiple dimensions problem scenario. This limitation prompted us to propose a new artificial intelligence tool known as Cooperative Multiple Particle Swarm Optimization (CMPSO) algorithm which can solve such computationally intricate problem efficiently. CMPSO algorithm takes its governing traits from the PSO. The proposed algorithm is marked by the cooperation among 'sister swarms' that makes it compatible to the problems pertaining to multiple dimensions. The limitation of restricted applicability to the multi-dimensional problems has been the prime reason of thinking behind the cooperative PSO. The objective of the proposed research aims to generate an efficient operating sequence which would explore maximum utilization of the manufacturing resources simultaneously meeting the customer's due date. To ease the solution strategy the underlying problem has been modelled as a travelling salesman problem (TSP). The traditional PSO uses a random number to determine the position and velocity of the particle during fitness evaluation. In the proposed research the random number has been replaced by the chaotic function because of the ergodic and stochastic properties of the chaotic systems. The idea behind this approach was to overcome the demerits associated with the random number generators such as requirement of more number of generations to converge towards an optimal/near optimal solution, tendency to generate the higher-order part more randomly than their lower-order counterpart etc. The chaotic sequences have been successfully applied in the area of natural phenomena modelling, neural network, DNA computing procedures etc. Different researchers use four chaotic sequences (Logistic Map, Tent Map, Sinusoidal Iterator, and Gauss Map) to generate optimal/near optimal solution preventing the premature convergence. Each of these functions are also associated with some merits and demerits, hence in this present research a hybrid chaotic sequence has also been proposed to overcome these demerits. The proposed research aims towards exploring the applicability of PSO technique under diverse situations by inheriting some new concepts. These hybrid PSO techniques (such as CMPSO) could be applied to efficiently solve number of computationally complex problems prevailing in manufacturing environment.

The chapter is organized as follows. Section 2 of the chapter discusses the literature review and attempts to find the gap in the research work in the proposed filed. Section 3 along with some sub-sections gives a brief idea of the problem scenario, and its mathematical formulation. Section 4 gives a background of the PSO algorithm further discussing the proposed CMPSO in detailed i.e. explaining the steps of the algorithm as well as about the chaotic functions. Section 5 explains a case study. Section 6 discusses the outcomes of the proposed CMPSO algorithm and shows a comparative performance measurement with other existing evolutionary algorithms. And finally, section 7 concludes the chapter with some future research directions.

2. Literature Review

In recent years, the changing business scenarios and escalating complexity especially in manufacturing plants have shifted the inclination of the researchers towards issues that have great impact on overall performance of the plants. The supply chain being one of the important aspects of profitability and performance of any plant have gained considerable attention these days. In the past the attention on the operational issues and the supply chain issues were dealt separately. Even process planning and the scheduling part were independent entities to be handled. However, increased competence & complexity prompted to integrate them together which led to the rigorous researches carried out to integrate the process planning and scheduling problems. Scheduling issues have been discussed by many researchers. Hankins *et al*. (1984) emphasized the advantages of alternative machines to increase the productivity of a machine shop, it also shows how mathematical programming techniques tends to become unaffordable when jobs have to be assigned and scheduled on a large set of alternative machines. To deal with such complexity, Khoshnevis and Chen (1991) suggested the use of various dispatching rules, which however suffers from context-dependence and performance unpredictability issues. Similar strategies are also suggested by Brandimarte and Calderini (1995) and Lin (2001).
The challenges associated to the computational complexity of integrated optimization problems in various types of manufacturing systems have stimulated many researchers to apply advanced approaches based on evolutionary computation (Dimopoulos and Zalzala, 2000) and related forms of meta-heuristics. Palmer (1996) applied Simulated Annealing based random search optimization technique to produce an integrated process plan and schedule for a manufacturing unit. Tiwari and Vidyarthi (2000) recognized the machine loading problem as one of the important planning problem in FMS. They utilized Genetic Algorithm based random search heuristic to determine the part type sequence and operation machine allocations to maximize the throughput and minimize the system unbalance. Swarnkar and Tiwari (2004) applied a hybrid Tabu Simulated Annealing based approach to model the machine loading in flexible manufacturing system (FMS). Similarly Tabu and constructive heuristic-based approaches have been proposed by Kolisch, and Hess (2000), Tan and Khoshnevis (2004), and Kolisch (2000). Hybrid approaches combining evolutionary computation with other tools have also gained increasing attention. Rai *et al.* (2002) solved a machine-tool selection and operation allocation problem in FMS based on a fuzzy goal programming model using a GA-based approach. Chiu and Lin (2004) introduced an approach based on Artificial Neural Networks (ANN) to achieve complete order fulfillment and increased resource utilization in a collaborative supply chain planning problem. Naso *et al.* (2007) have proposed a hybrid meta-heuristic in which Genetic Algorithm has been used as master search algorithm that guides and modifies the operations of subordinate algorithm (a set of very fast constructive heuristics) to achieve efficient solutions in acceptable search time for an integrated production and distribution problem with strict time delivery constraints. Chang and Lee (2004) provided a detailed discussion of a two-stage (production and distribution) case in which the case of one production center and one vehicle with makespan minimization is shown to be a NP-hard problem. Additionally, the authors have proposed number of heuristics with guaranteed worst case performances. Garcia *et al.* (2002) proposes GA based approach for the coordination between production and transportation operations in multi- plant supply chain environment.

The literature review gives a clear indication of the ever growing interest in problems related to distributed production planning and scheduling. It may be worth mentioning that while the broad umbrella of distributed scheduling also covers research studies on distributed optimization e.g. Nishi & Konishi (2005), this paper is focused on multi-plant (hence distributed) environments governed by a centralized decision system. This problem is also referred to as distributed scheduling (Chan *et al.*, 2005) with reference to the fact that the assignment of jobs to alternative suitable factories must be solved before (or jointly with) the overall production scheduling. Integrated process planning and scheduling problem is a NP hard problem. In order to solve such problems in past various types of evolutionary algorithms (EAs), heuristics and Meta heuristics have been proposed. However, all those heuristics were not able to completely solve the problem efficiently in real time. In the present research integrated process planning and scheduling problem under MPSC environment has been considered which is more complex than previous scenarios. Recently Particle Swarm Optimization (PSO) algorithm has appeared to be one of the powerful tools to solve such complex problems, which could be envisaged by its implementation in health sectors, manufacturing sectors, etc.

PSO being one of the emerging computational techniques for optimality has received a lot of attention in recent years. This could be visualized both in terms of number of research output produced, as well as conferences organized on this topic in past few years, such as Congress on Evolutionary Computation (CEC) and Genetic and Evolutionary Computation Conference (GECCO), (Hu *et al.* 2004). The successful applicability of PSO ranges in a broad domain of research areas such as in artificial neural network training (Eberhart and Shi 1998b, Messerschmidt and Engelbrecht 2004), the optimal power flow (OPF) problem (Abido 2002), the task assignment problem (Salman *et al.* 2002), the unit commitment (UC) problem (Ting *et al.* 2003), quantitative structure–activity relationship (QSAR) model construction (Cedeno and Agrafiotis 2003), multiple sequence alignment (MSA) (Rasmussen and Krink 2004), multi-modal biomedical image registration (Wachowiak *et al.* 2004), multi-objective optimization (Coello *et al* 2004), electromagnetic optimizations (Robinson and Rahmat-Samii 2003, Boeringer and Werner 2004), blind source separation (BBS) (Gao and Xie 2004), protein motif discovery (Chang *et al.* 2004), etc.

The accomplishment of the PSO technique lies in its ability to produce competitive or even better results in a faster way, compared to other heuristic methods such as GA. The general applicable areas where the other evolutionary computation techniques are practiced are the good application areas for PSO (Eberhart and Shi 2004). PSO and GA have many similarities, such as both the algorithm starts with the random population generation and both of them have fitness values to evaluate the population. Also in both cases the updation process and optimality search procedure is based on the random techniques. The difference lies in the fact that PSO does not have genetic operators such as crossover and mutation. In PSO particles update themselves with the internal velocity and have memory of the previous best solution which is an important aspect of the algorithm. Several key issues related to PSO and GA has been pointed out by Rahmat-Samii (2003). The prime advantage of the PSO over the GA is its algorithmic simplicity. Both GA and PSO have several numerical parameters that need to be carefully selected. However, the robustness to control parameters makes their selection even easier for PSO (Trelea 2003). Another advantage of PSO over GA is the ability to control convergence. It has been shown that the decrease of inertial weight dramatically increases the swarm's convergence. Stagnation may occur in

GA however, in case of PSO this effect can be controlled or prevented easily, for example, by decreasing the inertial weight during the evolution process (Eberhart and Shi 1998a, Clerc and Kennedy 2002). The capability of PSO to converge towards optimality or near optimality faster makes it a preferable option as compared to GA.

However, certain limitations are as applicable to PSO, as in case of other evolutionary algorithms. To overcome these limitations many researchers have proposed numerous variants of PSO. Angeline (1998) in his study showed that though PSO converges to reasonable quality solutions much faster than other evolutionary algorithms, the quality of the solutions does not improves with the increased number of generations. Xie *et al.* (2002) proposed an opening Dissipative System (DPSO) to prevent the stagnation in PSO by introducing negative entropy through additional chaos for particles. Overshooting in PSO is an important situation that is often used to occur, which causes premature convergence and is essential for the performance of PSO. The overshooting problem affects the velocity update mechanism leading the particles to the wrong or opposite directions against the direction of the global optimum. As a consequence, the pace of convergence of the whole swarm to the global optimum slows down. In order to overcome the this Liu *et al.* (2005) proposed a novel Memetic-Particle Swarm Optimization that integrates the standard PSO with the Solis and Wets local search strategy to avoid the overshooting problem, and that is based on the recent probability of success to efficiently generate a new candidate solution around the current particle.

The present work considers the MPSC environment where it is very difficult to apply normal PSO because of its inability to handle multi-dimensional problems. The shifting trend of the industries towards the new supply chain environment prompts to develop an evolutionary algorithm that could be efficiently employed to solve the complex problems. Realizing the applicability and efficacy of PSO in solving the complex operational sequencing problems prompted us to use as a powerful tool in the present research. Hence, to overcome the difficulty/limitation of applicability of normal PSO in multi-supply chain scenario the present research attempts to propose a new type of PSO termed as Cooperative Multi plant particle Swarm Optimization (CMPSO) algorithm that could be successfully applied in case of Multi-plant supply chain problem. In order to solve Multi-dimensional problem scenario in the proposed CMPSO algorithm the sister swarms explores the search space to reach towards optimality/sub-optimal by cooperating each other.

3. Problem Formulation

Globalization, increased competence, and continuously changing business environment have driven the manufacturing industries to a new era of enhanced complexity and uncertainty. These changes have great impact on the performance of the manufacturing industries. The manufacturing entities are suffering from operational difficulties, such as due to economies of scale of production and long operational time preparation; it has been quite difficult for them to prepare the production schedule in accordance with their due dates. A schematic representation of integrated process planning and scheduling model describing its various components in a multi plant supply chain environment has been shown in Figure 1. The process planning module is responsible for the generation of an effective process plan, incorporating the features of part design specification, available machine characteristics and their mutual relationship. The scheduling module is responsible for allocating the available resources in the shop floor as well as overall management of the flow of production order and their attendant information flows (Huang *et al.*, 1995).

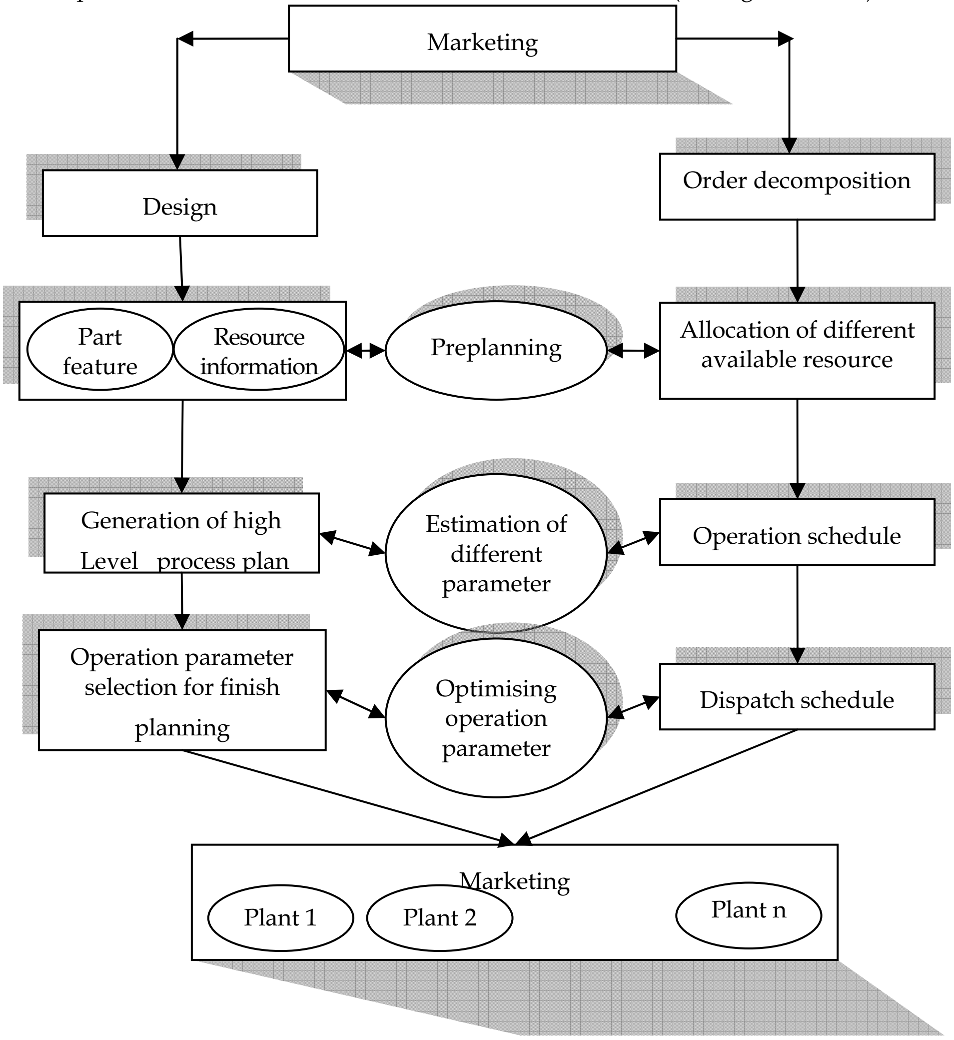


Figure 1. General architecture of integrated process planning and scheduling model

This model consists of four layers: (a) Supply (b) Fabrication (c) Assembly and (d) Customers, respectively. Out of these four layers, fabrication layer and assembly layers can be treated as directly linked to the production process which needs to be optimized. Hence, these two layers play a crucial role in the MPSC optimization. In general, MPSC industries posses the property of having multiple orders with different due dates. Under such scenario each order may have several parts with dissimilar array of operations. Some of these operations may have precedence constraints relation, whilst some others might be iterative in nature. These variations typically make the nature of Integrated Process Planning and

Scheduling a NP hard problem. The proper representation of such types of problems were proposed by, Finke *et al.*(1984) who modeled such problem state as a *Traveling Salesman Problem* (TSP) model with precedence relationships [Weiqi *et al.* (2002), Pepper *et al.* (2002)]. The travel distance between the two nodes corresponds to the transition costs between the operations. The selection of machine for each operation is not uncomplicated, because there may be numerous alternative machines for each operation. A classic selection criterion considers operational time, set up time and transportation time as decision attributes. Moreover, each TSP determines the process planning and scheduling for each part type. Accordingly, for multiple part type problems, multiple TSP has been considered. The fundamental characteristic for these types of systems are constituted by lot sizes (Nasr and Elsayed, 1990). The TSP model is based on some rules which involve transferring of the parts. In TSP environment if the transfer batch is equal to the process batch, then the part is transferred to subsequent stage after the completion of the entire batch processing, whereas if transferred batch differs from the process batch, then the part is immediately moved to the subsequent operation after the completion of current operation.

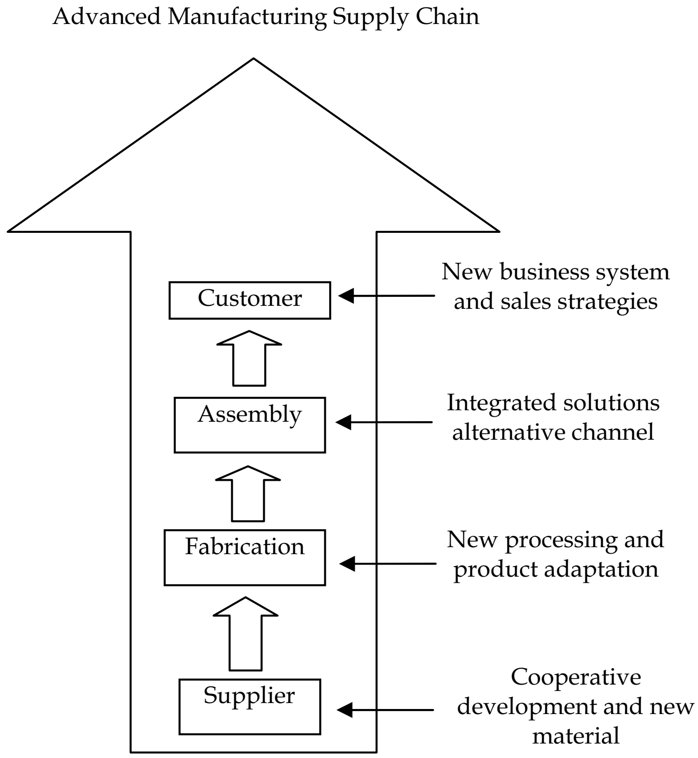


Figure 2. Schematic structure of flexible manufacturing processes

Two-commodity network flow model can be used to generate a feasible operation sequence with precedence constraints criteria in TSP problems (Kusiak and Finke, 1987). The edges of the flow network symbolize the precedence constraints. Let q and r be two distinct commodities in the network with k nodes. The selected starting node q provides $k\text{-}1$ units to the commodity q whereas, on the rest of the nodes, q is used by one unit each. On the other hand, r represents the commodity that utilizes k-1 units at the starting node and is supplied by one unit at the other nodes. Such network flows of the commodities are characterized by two properties, first the sum of the commodities q and r in any feasible tour should be equal to k-1 (Moon *et al.*, 2002). Also, as the tour proceeds, the quantity of commodity q or r out bonded from a node decreases. Precedence relationships of constrained TSP are modeled using these characteristics.

3.1 Operation Sequence

In this research, we develop a CMPSO algorithm with amalgamated features of directed graph and *Topological Sort* (TS) techniques to generate an optimal/nearly-optimal feasible solution. In a directed graph, vertices represent operations and edges represent precedence relations between different operations. The directed edge of the directed graph can be represented by *emi* , *emj*; where vertex *emi* must be completed before the vertex *emj*. The search algorithm is executed first to assign a fixed priority number corresponding to each vertex of the directed graph. Thereafter, TS technique is applied to generate a unique feasible operation sequence according to the assigned priority numbers. Directed graph of a manufacturing process carried out in the two plants is illustrated in the Figure 3.

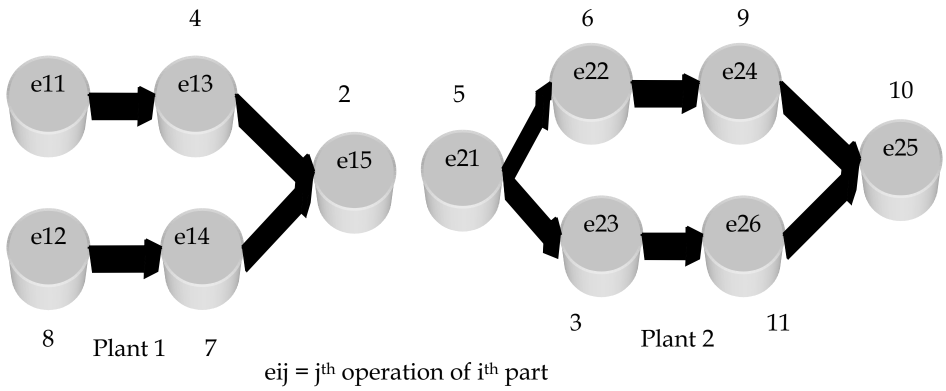


Figure 3. Directed graph of a manufacturing process with precedence relationship

The vertex *e11* is selected as first operation because it has no precedence edge and has higher priority number as compared to vertex *e12* and *e13*. Select vertex *e11* as the first operation and remove the edges connecting to the vertex *e11-e13*. This procedure is repeated until all the vertices are selected. Finally, a feasible path {*e11, e13, e21, e23, e22, e12,,e14, e15, e24, ,e26, e25* } is uniquely obtained. Therefore, operation sequence for each part types may be written as follows.

Part1 = { *e11,e13,e12,e14,e15*}, Part2 = { *e21,e23,e22,e24,e26 , e25*}.

3.2 Objective Function

The main objective considered in the proposed research involves the generation of a feasible optimal/near optimal operational sequence with minimum tardiness. The solution generated by the proposed CMPSO algorithm is also subjected to satisfy all the constraints and decision variables imposed by the considered manufacturing scenario. The mathematical formulations of the objective function including the various constraints, and decision variables have been explained below. The notations used in the equations have been explained in the Appendix A attached at the end of the chapter.

Decision Variables:

$$\phi_{pij} = \begin{cases} 1; \textit{if operation of part type is performed immediately after the completion of opertation } j \\ 0; \textit{otherwise} \end{cases}$$

$$\phi_{pij} = \begin{cases} 1; \textit{if machine } m \textit{ is selected for operation } i \textit{ of the part type} \\ 0; \textit{otherwise} \end{cases}$$

$$\phi_{pij} = \begin{cases} 1; \textit{if operation } i \textit{ precedes opertation } j \textit{ on the machine } m \\ 0; \textit{otherwise} \end{cases} \qquad \forall [i, j] \in \beta_m, m = 1,2,3,...M$$

Constraints:

C_1: Commodity feasibility constraints:

Between the operations w_{pi} to w_{pj} commodity q of the part type p is feasible if it satisfied the following constraints.

$$\sum_{j=1}^{J_p} \phi_{pij}^q - \sum_{j=1}^{J_p} \phi_{pji}^q = \begin{cases} J_p - 1, & \text{for i} = \eta_\text{p} \\ -1, & \text{otherwise} \end{cases} \qquad \forall \text{ P} \tag{1}$$

$$\phi_{pij}^q, \phi_{\text{pji}}^{\text{q}} \geq 0 \;\; \forall \text{ p, i, j} \tag{2}$$

Similarly between the operations w_{pi} to w_{pj} commodity r is feasible if it satisfied the following constraints.

$$\sum_{j=1}^{J_p} \phi_{pij}^r - \sum_{j=1}^{J_p} \phi_{pji}^r = \begin{cases} -(J_p - 1), & \text{for i} = \eta_\text{p} \\ -1, & \text{otherwise} \end{cases} \qquad \forall \text{ P} \tag{3}$$

$$\phi_{pij}^r, \phi_{\text{pji}}^{\text{r}} \geq 0 \;\; \forall \text{ p, i, j} \tag{4}$$

C_2 Precedence constraints:

Precedence relations between operations are feasible if the difference between sum of the commodity q from operation $w_{p\alpha}$ to w_{pj} and from operation $w_{p\beta}$ to w_{pj} for all the part type p is greater then or equal to 1.

$$\sum_{j=1}^{J_p} \phi_{p\alpha j}^q - \sum_{j=1}^{J_p} \phi_{p\beta j}^q = \sum_{j=1}^{J} \phi_{\beta j}^q \geq 1, \qquad \forall \text{ P and} \left(\text{w}_{\text{p}\alpha} \to \text{w}_{\text{p}\beta}\right)\left(\text{w}_{\text{p}\beta} \neq \eta_p\right) \tag{5}$$

C_3: Sum of commodity constraints:

For a feasible operation sequence sum of commodities q and r between the operation w_{pi} and w_{pj} is equal to J_p-1..

$$\phi_{pij}^q + \phi_{\text{pij}}^{\text{r}} = \left(J_p - 1\right)\phi_{pij} \qquad \forall \text{ p, i and j.} \tag{6}$$

C_4: Machine Constraints:
This constraint implies that machine will start a new operation only after completing the previous operation. This constraint can be express as:

$$\xi_{hjm} - \xi_{pim} + \theta(1 - \gamma_{ijm}) \geq \mu_{pjm} \tag{7}$$

Where θ is a very large positive number
C_5: Operational time constraints:
The completion time of each operation should always be positive or zero.

$$\xi_{pim} \geq 0 \tag{8}$$

C_6: Feasibility of tour constraints:
Operation sequence of the part type *p* is feasible if sum of commodities *q* and r is equal to J_p-*1*.

$$\sum_{j=1}^{J_P} \left(\phi^q_{pij} + \phi^r_{pij}\right) = J_p - 1 \qquad \forall \text{p and i}, \tag{9}$$

Objective Function:

The number of transportation from the operation w_{pi} to w_{pj} for the part type p having lot size of production r_p is

$$\hbar_{pij} = \left[\frac{1_p}{\alpha_{pij}}\right] \tag{10}$$

Symbol $[\ ]$ represents the greatest integer function.
The transition time from operation *i* performed on the machine m to operation *j* performed on the machine n of the part type *p* can be expressed as follows

$$t_{pij} = \{r_p \ \mu_{pim} \ \varphi_{pim} + \hbar_{pij} \ \upsilon_{mn} \ \varphi_{pim} \ \varphi_{pjn} + \delta_{pij} \ \phi_{pij}\} \tag{11}$$

Total transition time for all the part types in 0-1 integer programming model is given by

$$T = \sum_{p=1}^{P} \sum_{j=1}^{J_p} \sum_{i=1}^{I_p} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{1}{J_p - 1} t_{pij} \left(\varphi^q_{pij} + \varphi^r_{pij}\right) \tag{12}$$

Subjected to

$$\sum_{m=1}^{M} \varphi_{pim} = 1 \qquad \forall \ p \ and \ i \tag{13}$$

And $$\varphi_{pim} \in \{0,1\} \qquad \forall_p, i \ and \ m \tag{14}$$

For part type p tardiness of an order Ω_p is the amount of time by which the completion time of it exceeds from its due date. It can be express as

$$\Omega_p = \max \begin{cases} C_p - du_p \\ 0 \end{cases} \tag{15}$$

Total tardiness of all the part type of an order Ω is

$$\Omega = \sum_{p=1}^{P} \Omega_p \tag{16}$$

The overall objective of integrated process planning and scheduling is to minimize the total tardiness of all the part type of an order.

$$\min imize \sum_{p=1}^{P} \Omega_p \tag{17}$$

It is possible only if total transition time T for all the part type is minimum that is

$$Minimize \sum_{p=1}^{P} \sum_{\substack{j=1 \\ i \neq j}}^{J_p} \sum_{i=1}^{I_p} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{1}{J_p - 1} t_{pij} \left(\varphi_{pij}^{q} + \varphi_{pij}^{r} \right) \tag{18}$$

Where, $$t_{pij} = \left\{ r_p \, \mu_{pim} \, \varphi_{pim} + \hbar_{pij} \, \upsilon_{mn} \, \varphi_{pim} \, \varphi_{pjn} + \delta_{pij} \, \phi_{pij} \right\} \tag{19}$$

The next section gives an insight on the PSO algorithm further expalining the necessity to propose the CMPSO algorithm to find the optimal/sub-optimal solution of the abovementioned objective funtion. The section also breifly explains the steps of the proposed algorithm, a hybrid chaotic sequencing, and a case study.

4. Particle Swarm Optimization (PSO)

The inclination of the researchers towards the implementation of the biologically inspired algorithms in solving the engineering problems have led to the invention of many algorithms such as Genetic Algorithm, Ant colony optimization, Artificial Immune System based algorithms etc. Particle swarm optimization (PSO) is one of the biologically inspired evolutionary algorithms which drive the idea from the flocking of birds. Abundant examples could be extracted from the nature that demonstrates that social sharing of information among the individuals of a population may provide an evolutionary advantage. PSO was first proposed by Kennedy and Eberhart (1995) and it has been deserved considerable attention in recent years in the global optimization areas. PSO originally intends to graphically mimic the elegant way in which swarms find their food sources and save themselves from predators (Eberhart and Kennedy 1995). It is a population-based stochastic optimization paradigm, in which each individual termed as *particle* from the population of *swarm* changes their position with time and represent a potential solution. PSO in some ways resembles with the other existing Evolutionary Algorithms, such as Genetic Algorithm, but the difference lies in its definition in a social context rather than biological context. According to Eberhart and Shi (2001) PSO is based on simple concepts with the ease of implementation and computational efficacy.

Particle Swarm Optimization (PSO) algorithm motivated by the flocking of the birds works on the social behavioral interaction among the particles in the swarm. It begins with the random initialization of a population of particles in the search space. These particles are considered to be in multidimensional space (D-dimensional) where each particle has a position and velocity. These two factors i.e. the position and velocity demonstrates the particles status in the search space. Hence in a PSO system, particles fly/move around in multi directions in the search space, and the position of each particle is guided accordingly by the memory of their own best position, as well as of a neighbouring particle. These particles communicate the best positions to each other and adjust their own position and velocity accordingly. Parsopoulos and Vrahatis (2002) proposed basically two main variants of the PSO algorithm:

- Global neighhborhood, where best global position is communicated to all partilces and updated immeditely in the swarm
- Local neighborhood, where each particle moves towards its best previous position and towards the best particle in its restricted neighborhood.

In the proposed work the global variant has been adapted. The reason behind opting for the global neighborhood is due to the fact that local neighborhood even though allows better exploration of the search space and reduces the susceptibility of PSO to falling into local minima; it slows down the convergence speed. The position and velocity vectors of the *i*th particle can be represented as

$$xi = (xi1, xi2, \ldots, xiD) \tag{20}$$

$$pi = (pi1, pi2, \ldots, piD) \tag{21}$$

The fittest particle among all the particles in the population is represented by

$$F = (f1, f2, fD) \tag{22}$$

The velocity vector for the *i*th particle can be represented as

$$Vi = (vi1, vi2, \ldots viD) \tag{23}$$

The updated velocity and position for the next fitness evaluation of each particle could be determined according to the following equations:

$$\upsilon_{id}^{k+1} = \omega \bullet \upsilon_{id}^{k} + c_1 \bullet R_1() \bullet \left(p_{id}^{k} - x_{id}^{k}\right) + c_2 \bullet R_2() \bullet \left(f_d^{k} - x_{id}^{k}\right) \tag{24}$$

$$x_{id}^{k+1} = x_{id}^{k} + \upsilon_{id}^{k+1} \tag{25}$$

Here k is the iteration number, d = 1, 2. . . D; i = 1, 2. . . N, and N is the size of the population (swarm). $c1$ and $c2$ are two positive values called acceleration constants, $R_1()$ and $R_2()$ are two independent random numbers that uniformly distribute between 0 and 1 and are used to stochastically vary the relative pull of *pi* and *f* (Clerc andKennedy 2002). The introduction of such random elements into the optimization is intended to simulate the slightly unpredictable component of natural swarm behavior. 'ω' is the inertial weight introduced by Shi and Eberhart (1998b) in order to improve the performance of the particle swarm optimizer.

The equation (24) contains the three terms on the right hand side in which the inertial effects of the movement is represented by the first term. The memory of the individual and whole is referred by second and third terms respectively. Basically, equation (24) is used to calculate the particle's new velocity which depends on its preceding velocity and the distances of its present position from both its own best past position and the group's best past position. All the other particles follow the best position found and moves closer to it, exploring the region more thoroughly in the process. According to Robinson and Rahmat-Samii (2004) equation (24) is the central constituent of the entire optimization. The stochastic tendency to return towards particle's preceding best position is represented by the second term of the equation (24). The third term in equation (24) is referred as *social influence* term. The variable *F* keeps moving towards the best solution i.e. optimal solution found by the neighbor particles in the search space. Particle farther from the global best is more strongly pulled from its location, and moves rapidly than a closer particle. The particles velocity comes to halt after it reaches the locations of best fitness. This is the point from where the particles are pulled back in the opposite direction. The performances of the individual particles are evaluated by a predefined fitness function dependent on the problem during the evolution of the swarm. In case of maximization of the fitness function Fitness *(xi)*, the individual best position of each particle *pi* and the global best position *f* are updated after each iteration using the following two equations, respectively:

$$p_i^{k+1} = \begin{cases} x_i^{k+1} : Fit\,(x_i^{k+1}) > Fit\,(p_i^{k+1}) \\ p_i^k \quad : Fit\,(x_i^{k+1}) \le Fit\,(p_i^{k+1}) \end{cases} \tag{26}$$

$$F^{k+1} = \arg\max Fit\,(p_i^{k+1}) \tag{27}$$

Where 'Fit' refers to the fitness value for the respective iteration.

4.1 CMPSO Algorithm & its implementation

This section describes the formulations of the CMPSO algorithm for the process planning and scheduling in multi plant supply chain scenario. The prime objective of the problem considered is to generate an operation sequence and simultaneously select an appropriate machine corresponding to each operation from existing alternatives. It is a multiple dimensional problem as shown in the Figure 4. In the figure, first row represents an operation sequence while the second row represents the machine corresponding to each operation. In order to resolve the complexity of the problem in this piece of research CMPSO algorithm has been proposed. One of the key issues in successful implementation of PSO to a specified engineering problem is the representation scheme, i.e. finding a suitable mapping between the problem solution and the PSO particle. In the proposed methodology during the exploration of the search space the sister swarms cooperate with each other.

4	7	9	2	1	5	3	6	8	1
2	3	1	2	2	4	3	2	1	2

Figure 4. Systematic representation of solution

In this paper each bit of solution are positive integers and comprises a non-continuous integer search space. Since the original PSO works on a real-valued search space, especially on the particle positions (i.e., the operation sequence and corresponding machine in this paper) is calculated using equation (25) which are real numbers. Hence, a conversion is needed between the real-valued positions and the positive-integer-valued indices. In order to meet the criteria the sign is ignored and value is changed to the closest integer. After calcualting the x_{id}^{k+1} term in equation (25) the changes mentioned earlier are applied leaving the rest part of the equations (24) and (25) same as in the original PSO. These changes does

not have any affect on the performance of the algorithm and has been proven to be feasible (Salman *et al.* 2002, Laskari *et al.* 2002, Parsopoulos and Vrahatis 2002).
The individuals in the swarm are initialized by randomly setting their positions and velocities using operation sequence or machine depending on the nature of the swarm. During the iteration, reset is performed only when the value of a new position component calculated from equation (25) is greater than the upper limitation of the search space. It should be noted that, in the first row at any time of the operation process two bits cannot have the same values. Hence, if a new component value is calculated using equation (25) (for first row) that already exists, a random small integer will be added to this value till no collision exists. This combination pfacilitates fast convergence and ensures near-optimal solutions by establishing a proper balance between exploration and exploitation. In case of simple PSO in equation (24) random numbers were genrated using the Randon function. Howver, during experimentation it has been found that random function are associated with some demerits. Hence, in order to overcome the demerits of the random number in this research not only it is being replaced by chaotic sequences, but also a new hybrid chaotic sequence has been proposed.
This paragraph explains the significance of applying a chaotic sequence generator to update the velocity instead of the random number generator. The random function used in equation (24) has been replaced with a chaotic function because of the ergodic and stochastic properties of the chaotic systems. One of the limitations coupled with the random number generators is that the solution becomes conserved by sequential correlation of successive cells; hence requiring more number of generations to converge towards an optimal or near-optimal solution. Also the commonly used random number generators have a tendency to generate the higher-order part more randomly than their lower-order counterpart (Caponetto *et al.*, 2003). Therefore, it requires a consistent random number generator which can explore search space without being biased. Recently, various chaotic sequences have been applied in areas related to secure transmission, neural networks, natural phenomena modelling, deoxyribonucleic acid computing procedures, and non-linear circuits (Arena *et al.*, (2000), Determan and Foster (1999), Manganaro and Pineda (1997), Sugantahn (1999), Nozawa (1992), and Wang and Smith (1998)) and encouraging results have been obtained with random number generators. The unpredictability characteristics, i.e. spread spectrum characteristics, justify theoretically the use of a chaotic sequence. Thus, the recent research drift towards the implementation of chaotic sequence generators in various AI tools motivated us to use in the present problem scenario. The commonly used chaotic equations by researchers are Logistic map-based chaos equation (LM), Tent map-based chaos equation (TM), Sinusoidal integrator-based chaos equation (SI), and Gauss map-based chaos equation (GM). As usual each equation has some advantage and some disadvantage and in order to overcome the demerit of each chaotic equation in this present research a hybrid chaotic equation termed as Chaotic Sequence-based Hybrid chaos equation (HC) has been proposed. The proposed chaotic equation incorporates the advantages of each chaotic equations mentioned below;
(A) Logistic map-based chaos equation (LM): In this method logistic map-based chaotic sequence is used to generate random numbers. It is one of the simplest dynamic systems evidencing chaotic behavior. The logistic map chaotic equation is delineated as follows.

$$Y_{k+1} = \omega Y_k (1 - Y_k) \tag{28}$$

Where ω is tuning parameter

(B) Tent map-based chaos equation (TM): In this method, random numbers are generated using Tent map-based chaotic sequence. It resemble as the logistic map which follows the following equations.

$$Y_{k+1} = \mu_k(Y_k) \tag{29}$$

$$\mu(Y_k) = \begin{cases} \dfrac{Y_k}{0.7}, & \textit{if } Y_k < 0.7 \\ \dfrac{1}{0.3} Y_k (1 - Y_k), & \textit{Otherwise} \end{cases} \tag{30}$$

(C) Sinusoidal integrator-based chaos equation (SI): In this chaotic equation, random numbers are generated using the following Sinusoidal Integrator relation:

$$Y_{k+1} = \sin(\pi Y_k) \tag{31}$$

(D) Gauss map-based chaotic equation(GM): In this chaotic equation, Gauss Map function is used to generate the random numbers and it transfers from one stage to another stage in a quadratic manner. Gauss Map function can be expressed as follows:

$$Y_{k+1} = \mu_k(Y_k) \tag{32}$$

$$\mu(Y_k) = \begin{cases} 0, & \textit{if } Y_k = 0 \\ \dfrac{1}{Y_k} \bmod 1, & Y_k \in (0,1) \end{cases} \tag{33}$$

(F) Chaotic sequence-based hybrid chaotic equation (HC): Randomly select a chaotic sequence strategy among aforementioned four strategies and generate random number using selected chaotic equation.

As mentioned earlier first row of each solution represents the operation sequence and second row represents the corresponding machine. For each individual row the proposed CMPSO algorithm runs separately. After updating the position and velocity for each row the sister swarm will cooperate with each other to evaluate the fitness function. On the basis of the computed fitness value the global and local best positions are decided. And after certain number of iterations the solution will tend to converge towards the optimality or sub-optimality. The stpes of the proposed CMPSO algorithm are shown below:

Step 1: Generate discrete search space for first and second row of solution i.e. maximum numbers of operation and number of possible machine corresponding to each option.

Step 2: Generate random initial solution, and assign random position X and velocity V vectors corresponding to each particle swarm (For both sister swarms) and assign number of generation (num_gen=1)

Step 3: Calculate the fitness value by the help of sister swarm and update the personal best position and global best position of each of sister swarm using equation (26) and (27) respectively.

Step 4: Update the velocity and position of the each swarm using the chaotic sequence mentioned above.

Step 5: num_gen=num_gen+1;

Step 6: If num_gen=max num_gen; go to step 7 otherwise go to step 3

Step7: Terminate the algorithm and global position is the optimal solution obtained by the algorithm.

Corresponding to the global best solution the optimal sequence and the corresponding machines are decided. A case study has been discussed in the next section.

5. Case Study

A case study has been taken from Moon *et al.* (2002) in this present work to reveal the efficacy and robustness of the proposed model, and to disclose the quality of the solutions found with the CMPSO. The case study involves an illustrative example which has been derived considering a constrained integrated process planning and scheduling problem. The case study involves three coupled decision problems i.e. selection of parts, priority of operation sequences and selection of appropriate machine for each operation to minimize the total tardiness in context of Advanced Integrated Process Planning and Scheduling (AIPPS) model. To make the case study more realistic additionally, various system constraints (precedence, commodity feasibility, sum of commodity, machine, operational time, and feasibility of tour constraints) have been considered. In the present case study the process planning and scheduling has been carried out simultaneously. The study also involves consideration of set up times between the operations, and transportation times between machines, thus closely resembling to the typical characteristics of industrial contexts. The case study consists of 5 different parts with different due dates {d1=1000, d2=1300, d3=2000, d4=1600, d5=1400}. These parts are manufactured in two plants (plant 1 and plant 2) having six different machine. Plant 1 consists of machines {M1, M2, M3} and plant 2 consists of machines {M4, M5, M6}. The production lot size for the each part type is considered {40, 70, 60, 30, 60} respectively. Also, the transportation time between plants is assumed to be 50, and unit size is considered equal to the lot size of each part. The remaining data needed in the problem such as initial load, transportation time, and set up time between alternative machines etc., are shown in Table 1-3, whereas the operation and there precedence constraints relation are shown in Figure 5.

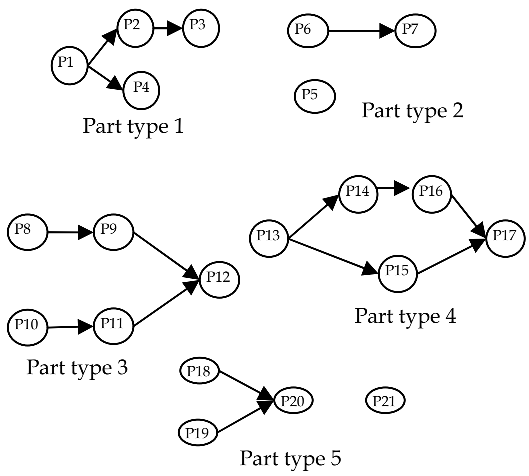


Figure 5. Precedence Relationship between different Operations

Operations performed in different plants	Operation performed in different plants																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
01	00	35	46	39	28	32	20	16	12	39	40	23	49	08	26	20	47	14	03	36	40
02	39	00	36	23	47	33	46	36	06	08	34	02	48	15	12	04	12	24	30	44	34
03	15	08	00	27	21	34	33	25	33	13	49	08	35	40	26	32	46	28	14	41	26
04	13	38	19	00	01	44	14	27	48	36	38	35	37	39	14	32	22	10	26	25	29
05	34	48	24	10	00	36	12	48	09	24	28	19	08	36	17	06	28	06	22	22	45
06	05	18	30	12	01	00	12	32	00	39	12	18	19	19	05	27	42	37	16	24	31
07	28	37	48	15	17	13	00	44	38	15	09	01	05	36	02	20	17	15	06	10	34
08	17	15	00	42	41	13	17	00	30	22	33	25	02	33	26	35	41	35	04	09	22
09	37	10	41	27	35	46	30	16	00	35	33	15	28	06	08	30	22	25	39	10	36
10	06	05	49	47	00	01	18	37	06	00	44	30	23	07	04	02	04	30	03	16	19
11	44	30	21	11	46	15	30	17	46	14	00	44	06	04	06	14	48	29	27	29	15
12	34	25	34	04	10	14	27	07	26	49	13	00	40	20	15	15	49	07	28	43	47
13	09	09	04	42	01	36	21	15	36	27	20	11	00	33	41	46	02	33	44	02	07
14	29	29	29	24	47	28	28	30	46	23	26	20	49	00	39	24	33	26	06	29	36
15	07	19	10	14	17	44	27	13	10	14	09	17	48	15	00	28	17	46	45	21	36
16	31	01	37	03	21	09	23	46	13	41	21	47	15	16	43	00	18	03	24	27	11
17	29	39	03	30	48	39	02	45	03	39	36	26	28	23	40	29	00	32	17	38	23
18	02	13	10	09	32	14	45	11	24	43	15	02	16	06	32	15	30	00	15	37	38
19	09	20	35	08	18	48	27	12	41	30	47	16	02	41	13	29	23	07	00	08	08
20	27	49	40	29	09	36	29	12	24	45	30	10	16	34	05	06	08	33	38	00	31
21	27	04	34	31	29	03	32	47	12	09	44	30	42	21	25	02	40	26	26	25	00

Table 1. Set up time between different operations

To further prove the efficacy of the proposed CMPSO algorithm, it has been tested on several randomly generated data set problems with increasing complexity as shown in the Table 4.

6. Result & Discussion

It has been concluded after comprehensive survey of research contributions in the broad domain of PSO application that number of iterations required to achieve optimal/near optimal solution is relatively high. Therefore, it is not only desirable but also inevitable to develop a meta-heuristic, which can overcome the drawbacks associated with simple PSO, and can solve large size combinatorial problems in lesser number of iterations and CPU time. The incapability associated with simple PSO algorithm in solving multidimensional problem prompted to develop an evolutionary meta-heuristic termed as CMPSO algorithm. The CMPSO algorithm has been proposed to solve Multi plant supply chain problem in the distributed manufacturing environments.

CMPSO algorithm achieves the optimal/near optimal solutions for the objective considered (Tardiness minimization) and emphasizes it as a powerful meta-heuristic algorithm. Use of hybrid chaotic sequence function empowers the algorithm to obtain optimal/near optimal results in significantly less number of generations. These features of the algorithm make it more effective as compared to simple PSO algorithm. Performance of the proposed CMPSO algorithm is found superior, when compared with GA and Tabu Search, on a set of problems adopted from the literature. For the case study mentioned in section 5, total tardiness obtained in 35 iterations is 32. The optimal operation sequence obtained by CMPSO algorithm is shown in Table 5. Table 6 presents a comparative analysis of proposed approach with others. Figure 6, shows a convergence trend of CMPSO algorithm along with number of generations. It is evident from the Table 6 that proposed approach outperformed the results obtained using existing methodologies. The operation sequence generated by Genetic Algorithm (Li *et al.*, 2005) shows total tardiness to be 39 in 42 generations.

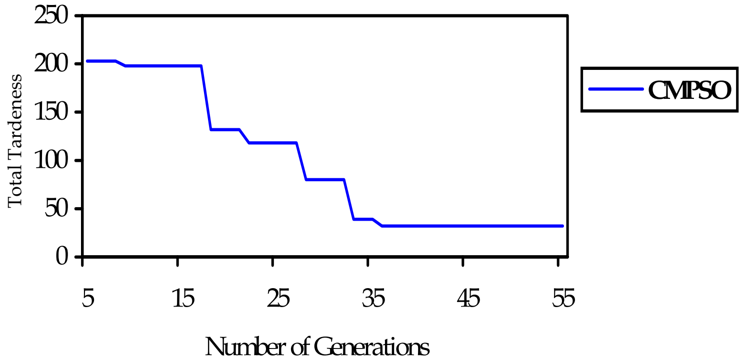


Figure 6. Convergence trend of CMPSO with number of generations

Various chaotic sequence operators that are used in literature were tested on sample problem and found that hybrid chaotic sequence based particle swarm optimization outperforms other PSOs. Their comparative results are shown in Figure 7. To have a better appraisal of the algorithm performance, a new parameter 'percentage Heuristic Gap (PHG)' (Huang et al. (2002)) has been utilized. Percentage heuristic gap can be mathematically defined as:

$$PHG = \frac{Best\ uppper\ bound - Best\ lower\ bound}{Best\ lower\ bound} \times 100 \qquad (34)$$

Here, lower bound is defined as the objective value obtained by relaxing some of the bounds or constraints pertaining to the problem environment, whereas upper bound is the objective function value of the feasible solution that fulfils all the constraints. In our case, one of the precedence constraints has been relaxed to obtain the lower bound. From the definition of PHG, it can be envisaged that the near optimal solution of the problem is guaranteed if its value is very small (Huang *et al.* (2002)) as shown in Figure 8. PHG for the test problems defined in the table 4 are shown in table 7-10. While average value of PHG for different size of data sets are shown in Table 11 which are less than 3%. Thus, from the definition of heuristic gap, it is inferred that solution obtained by CMPSO algorithm is near optimal one. A two way ANOVA without replication was also performed to assess the significance of the problem parameters. The results of the ANOVA test are in listed the Table 12-13. The results obtained by ANOVA test, performed at 99.5 % confidence level, validates the robustness of the CMPSO algorithm pertaining to MPSC problems.

The CMPSO algorithm has been coded in C++ programming language and experiments have been carried out on an Intel Pentium IV 2.4 GHz processor. In the nutshell, aforementioned computational results not only authenticate the efficacy and supremacy of the proposed algorithm, but also provide new dimensions to the solution of complex combinatorial problems like integrated process planning and scheduling problem in MPSC environment.

Plant	-	1	1	1	2	2	2
part	Operation	M1	M2	M3	M4	M5	M6
1	1	7	-	-	5	-	-
1	2	7	-	-	6	-	-
1	3	-	6	5	-	8	-
1	4	6	-	-	-	-	5
2	5	-	9	-	8	-	-
2	6	3	5	-	-	6	-
2	7	8	-	12	9	-	8
3	8	-	-	5	-	8	-
3	9	10	-	-	10	-	7
3	10	6	5	-	-	6	-
3	11	15	-	-	6	-	5
3	12	-	6	-	-	5	-
4	13	-	-	6	6	-	8
4	14	-	5	-	-	9	-
4	15	-	-	6	4	-	-
5	18	-	8	-	6	-	8
5	19	-	7	10	-	8	-
5	20	13	-	-	-	8	9
5	21	-	-	7	6	-	-

Where M1, M2, M3, M4, M5, M6 are different machines in plant 1 and 2.

Table 2. Machining time for different operation and alternative available

7. Conclusions

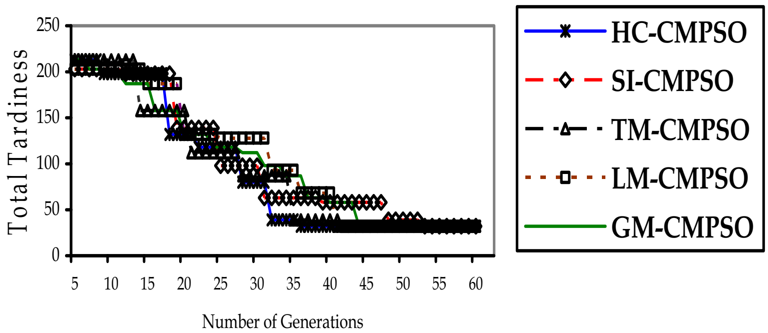


Figure 7. Comparative plot of CMPSO based on different chaotic equations

The efficient process planning and scheduling in multi-plant supply chain environment is gaining prime importance these days. Especially the manufacturing units are keen on finding efficient ways of handling such problems. This popularity and interest motivated our present research to build a realistic MPSC model and emphasize on its proper scheduling aiming to reduce the overall tardiness. Recognizing the fact that MPSC problem is a NP hard problem and quite complex to be solved by most of the existing evolutionary algorithms (EAs), this paper also proposes a new Cooperative Multiple Particle Swarm Optimization (CMPSO) algorithm to overcome the drawbacks of existing EAs. The prime objective considered in this paper was to reduce the overall tardiness considering several constraints, selection criteria's, decision variables and operational sequences. To build the model more close to realistic situation the setup time and transportation time has also been considered during problem formulation. The limitations associated with normal PSO while its application in multi plant situation was overcome by the newly proposed CMPSO algorithm. The comparative analysis of the CMPSO algorithm with other existing EAs shows its superiority over others. The CMPSO algorithm has also been statistically validated by performing ANOVA and Percentage heuristic gap analysis. The comparative analysis reveals that CMPSO algorithm not only does performs well to converge towards optimality/sub-optimality, the computational time required is also relatively less as compared to others. The paper also attempts to overcome the demerit of the normal PSO regarding the random number generators by using the newly proposed chaotic function instead.

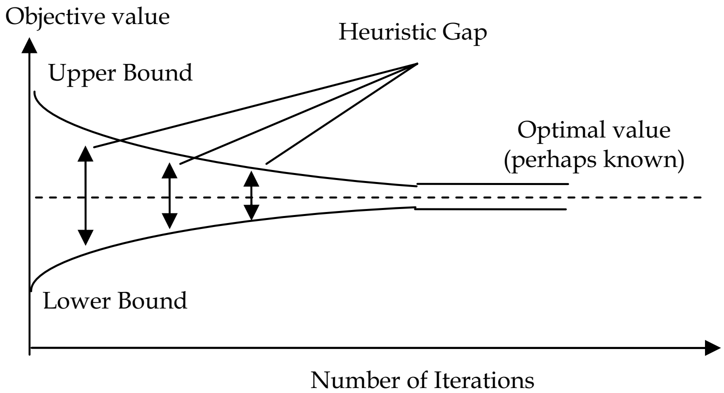


Figure 8. Heuristic gaps as a function of number of iterations

		Plant 1			Plant2		
	Machine	M1	M2	M3	M4	M5	M6
Plant 1	M1	0	5	6	-	-	-
	M2	5	0	7	-	-	-
	M3	6	7	0	-	-	-
Plant 2	M4	-	-	-	0	5	6
	M5	-	-	-	5	0	7
	M6	-	-	-	6	7	0

Table 3. Transportation times between machines

Though, CMPSO algorithm possesses many advantages over the traditional PSO and other existing evolutionary algorithms it has plenty of scope for its future extension. Future research can be directed towards its implementation in diverse areas of manufacturing fields. CMPSO algorithm also needs to be tested on multi-objective problems under various constraints, decision variables etc. to disclose its capability of handling diverse complex problems. The proposed algorithm has promising aspects that deserve further investigations; therefore work also needs to be focussed on further improving its efficacy and robustness.

Classification	Number of jobs	Number of operations
Very Small problem (VSP)	3	10-20
	12	20-30
Small Problem (SP)	15	30-40
	20	40-50
Large Problem (LP)	22	50-60
	25	60-70
Very Large Problem (VLP)	28	70-80
	35	80-90

Table 4. Detailed randomly generated data sets

		Operation		Operation		Operation		Operation	
Plant	Machine	start time	end time	start time	end time	start time	end time	start time	end time
	M1	10		7		17			
1			709	727	1287	1413	1563		
	M2	6		3		14		16	
		100	450	775	1015	1062	1212	1243	1393
	M3	8		13		21			15
		250	550	552	732	782	1202	1227	1407
	M4		1	2		5			
2		250	450	485	725	772	1332		
	M5		19	20			12		
		187	667	675	1155	1357	1657		
	M6	18		4		9		11	
		100	580	589	789	839	1257	1290	1590

Table 5. Final operations schedules of the Case Study using CMPSO

Total Tardiness	Using GA (Moon *et al*. 2002)	Using Tabu Search (Moon *et al*. 2002)	CMPSO Algorithm
Total Tardiness	39	39	32
Number of generations	42	>>GA	35
CPU Time	7 sec	48 sec	4 sec

Table 6. Comparative Result of the proposed CMPSO algorithm

Number of Jobs	Number of operations	% Heuristic gap (HG)
3	10	1.382
3	15	2.052
12	20	1.678
12	25	2.236

Table 7. Computational results for very small sized problem

Number of Jobs	Number of operations	% Heuristic gap (HG)
15	30	1.302
15	35	1.524
20	40	2.134
20	45	1.182

Table 8. Computational results for small sized problem

Number of Jobs	Number of operations	% Heuristic gap (HG)
22	50	2.167
22	55	2.005
25	60	2.761
25	65	1.755

Table 9. Computational results for largel sized problem

Number of Jobs	Number of operations	% Heuristic gap (HG)
28	70	1.854
28	75	2.530
35	80	2.345
35	85	2.449

Table 10. Computational results for very large sized problem

	L	H	Average
VSP	1.717	1.957	1.837
SP	1.413	1.658	1.6005
LP	2.087	2.258	2.1675
VLP	2.192	2.397	2.2943

L : Average *PHP* values for the smaller number of customers in the respective categories.
H : Average *PHP* values for the larger number of customers in the respective categories.
Table 11. Average heuristic gap for different problem sizes

SUMMARY	Row 1	Row 2	Row 3	Row 4	Column 1	Column 2
Count	2	2	2	2	4	4
Sum	3.674	3.071	4.345	4.589	7.409	8.27
Average	1.837	1.5355	2.1725	2.2945	1.85225	2.0675
Variance	0.0288	0.030013	0.01462	0.021012	0.127257	0.108254

Table 12. Intermediate values of the two way ANOVA test without replication

Source of Variation	Rows	Columns	Error	Total
SS	0.704751	0.092665	0.00178	0.799197
Df	3	1	3	7
MS	0.234917	0.092665	0.000593	
F	395.8443	156.1443		
P-value	0.000215	0.001105		
F crit*	47.46835	55.55194		
* α =0.005				

Table 13. Results of ANOVA test

Appendix A: List of notations used in the Mathematical Formulation

C_p	:	Completion time of the part type *p*.
du_p	:	Due date of the part type *p*.
$\hbar_{pij}$	:	Number of transportation from the operation w_{pi} to w_{pj} of the part type *p*.
IML_m	:	Initial mean load on the machines *m*
J_P	:	Total number of operations of part type p.
M	:	Set of different machines, $M = \{1,2,\ldots,\ldots,m,\ldots M\}$
$MT_{K,J}$	:	Machining time for k th operation corresponding to the machine assign in the i th male chromosome.
$MT_{k,j}$	:	Machining time for k th operation corresponding to the machine assign in the j th female chromosome.
n	:	Number of male or female population.
P	:	Set of different part types, $P = \{1, 2, 3, 4, \ldots, P\}$
R	:	Lot size of production of different part type, $R = \{r_1, r_2, r_3, \ldots, r_p\}$ where r_p is lot size of the part type p.
t_{pij}	:	Transition time from operation i to j for the part type p.
w_{pi}	:	i th operation of the part type *p*.
w_p	:	Set of operations for part type p, $W_p = \{w_{pi} \mid \forall\ i = 1,2,3,4, \ldots, j_p\}$, where w_{pi} is i th operation of the part type *p*.
$Z_{i,k}$	:	k^{th} bit of i th chromosome.
α_{pij}	:	Lot size of transportation between the operation w_{pi} to w_{pj}.
β_m	:	Set of operations on the machine *m*.
δ_{pij}	:	Setup time of machine between the operation w_{pi} to w_{pj} of the part type *p*.
ε_i	:	Number of alternative machine available for the operation i.
η_p	:	First selected operation for the part type *p*.
θ	:	An arbitrarily large positive number.
μ_{pim}	:	Processing time of *i* th operation of the part type *p* on the machine m.
ξ_{pim}	:	Completion time of operation i for part type *p* on the machine *m*.

ξ_{hjm}	:	Completion time of operation j for part type *h* on the machine *m*.
υ_{mn}	:	Transportation time from machines *m* to *n*.
ϕ^{q}_{pji}	:	Quantity of commodity q transferred from operation w_{pj} to w_{pi} for the part type P
ϕ^{r}_{pij}	:	Quantity of commodity r transferred from operation w_{pi} to w_{pj} for the part type P.
ϕ^{r}_{pji}	:	Quantity of commodity r transferred from operation w_{pj} to w_{pi} for the part type P
$\phi^{q}_{p\alpha j}$	:	Quantity of commodity q transferred from operation $w_{p\alpha}$ to w_{pj} for the part type P.
$\phi^{q}_{p\beta j}$	:	Quantity of commodity q transfer from operation $w_{p\beta}$ to w_{pj} for the part type P.
Ω_p	:	Total tardiness of part type p.
Ω	:	Total tardiness of all jobs.

8. References

Abido, M A., (2002). Optimal power flow using particle swarm optimization, *Electric Power Energy Systems*, Vol.26, Pages 563–571

Angeline, P.J. (1998). Using selection to improve particle swarm optimization, *The 1998 IEEE International Conference and IEEE World Congress on Computational Intelligence*, Evolutionary Computation Proceedings, Anchorage, AK, USA, Pages 84-89, ISBN: 0-7803-4869-9

Boeringer, D., and WandWerner, D. H (2004). Particle swarm optimization versus genetic algorithms for phased array synthesis, *IEEE Trans. Antennas Propag*., Vol. 52, Pages 771–779

Brandimarte, P., and Calderini, M., (1995). A heuristic bi-criterion approach to integrated process plan selection and job shop scheduling. *International Journal of Production Research*, Vol.33, Pages 161-181

Carlisle, A., and Dozier, G. (2001). An off-the-shelf PSO, *Proceedings of the Particle Swarm Optimization Workshop*, Pages 1–6

Caponetto, R., Fortuna, L., Fazzino, S., and Xibilia, M. G. (2003).Chaotic sequences to improve the performance of evolutionary algorithm. *IEEE Trans.*, Vol.7, No.3, Pages 289–304.

Cedeño Walter , and Agrafiotis, Dimitris K. (2003). Using particle swarms for the development of QSAR models based on K-nearest neighbor and kernel regression, *Journal of Computer-Aided Molecular Design*, Vol.17, No.2-4, Pages 255-263, ISSN0920-654X (Print) 1573-4951 (Online)

CedenoWand, Agrafiotis, D K., (2003) Using particle swarms for the development of QSAR models based on K-nearest neighbor and kernel regression, *Journal of Computer Aided Mol. Des*., Vol.17, Pages 255–263

Chan, F.T.S., Chung, S.H., and Chan, P.L.Y. (2005). An adaptive genetic algorithm with dominated genes for distributed scheduling problems, *Expert Systems with Applications*, Vol. 29, Pages 364–371.

Chang, H., Ratnaweera. A., Halgamuge, S. K., and Watson, H. C. (2004). Particle swarm optimization for protein motif discovery Genet. Program*., Evol. Mach*., Vol. 5, Pages 203–214

Chang, Y. C., Lee, C. Y. (2004). Machine scheduling with job delivery coordination, *European Journal of Operational Research*, Vol. 158, Pages 470487

Chiu, M. and Lin, G. (2004). Collaborative supply chain planning using the artificial neural network approach, *Journal of Manufacturing Technology Management*, Vol.15, No.8, Pages 787-796

Dimopoulos, C. and Zalzala, A.M.S. (2000). Recent developments in Evolutionary computation for manufacturing optimisation: problems, solutions, and comparisons, *IEEE Trans. Evolutionary Computation*, Vol.4, No.2, Pages 93-113, ISSN: 1089-778X

Eberhart, R. C., and Shi, Y. (1998a) Comparison between genetic algorithms and particle swarm optimization, *Proc. 7th Ann. Conf. on Evolutionary Programming*, Berlin, Springer, Pages 611–15

Eberhart, R C., and Shi,Y., (1998b). Evolving artificial neural networks *Proc. Int. Congress on Neural Networks and Brain*, Beijing, China, Pages 5–13

Eberhart, R. C., and Kennedy, J. (1995) A new optimizer using particle swarm theory, *Proc. 6th Int. Symp. on Micro Machine and Human Science*, Nagoya, Japan, (Piscataway, NJ: IEEE) Pages 39–43

Eberhart, R. C., and Shi, Y. (2001). Particle swarm optimization: developments, applications and resources*, Proc. Congress on Evolutionary Computation, Hawaii*, Pages 81–6

Eberhart, R. C., and Shi, Y (2004). Guest Editorial special issue on particle swarm optimization. *IEEE Trans. Evolutionary Computation*, Pages 201–203

Finke, G., Claus, A., and Gunn, E. (1984). A two- commodity network flow approach to traveling salesman problem. *Congressus Numerantium*, Vol.41, Pages167-178

Gao, Y., and Xie, S. (2004). A blind source separation algorithm using particle swarm optimization, *Proceedings of the IEEE 6th Circuits and Systems Symposium*, Shanghai, China, Pages 297–300

Garcia, J. M., Lozano, S., Smith, K., Kwok, T., Villa, G. (2002). Coordinated scheduling of production and delivery from multiple plants and with time windows using genetic algorithms. *Proceedings of the 9th International Conference on Neural Information Processing*, *ICONIP '02*, Vol. 3, Pages 1153 1158.

Hankins, S.L., Wysk, R.A., and Fox, K.R. (1984). Using a CATS database for alternative machine loading, *Journal of Manufacturing Systems*, Vol. 3, Pages 115-120,

Hu X, Shi Y and Eberhart, R C., (2004). Recent advances in particle swarm *Proc. IEEE Int. Conf. on Evolutionary Computation*, Pages 90–97

Huang, F., Zhang H., Li, Xiaodong, and Li Heng. (2006). Particle swarm optimization-based schemes for resource-constrained project scheduling, *Automation in Construction*, Vol. 15, No. 2, Page 252

Huang, S.H., Zhang, H.C., and Smith, M.L. (1995). A Progressive Approach for the Integration of Process Planning and Scheduling. *IIE Transaction,* Vol.27, No.4, Pages 456-464, ISSN: 0740-817X

Kennedy. J., and Eberhart, R. C 1995 Particle swarm optimization Proc. IEE 1110 se kab takE *Int. Conf. on Neural Networks (Piscataway,NJ: IEEE)*, Pages 1942–48

Khoshnevis, B., and Chen, Q. (1991). Integration of Process Planning and Scheduling Functions. *Journal of Intelligent Manufacturing*, Vol.2, No.3, Pages 165-176, 1990, ISSN0956-5515 (Print) 1572-8145 (Online)

Kolisch, R. (2000). Integrated assemble and fabrication for make-to-order production, *International Journal of Production Economy*, Vol.65, Pages 189-306

Kolisch, R., and Hess, K. (2000). Efficient methods for scheduling make-to-order assemblies under resource assembly area and part availability constraints, *International Journal of Production Research*, Vol.38, No.1, Pages 207-228

Kusiak, A., and Finke, G., (1987). Modeling and solving the flexible forging module scheduling problem. *Engineering Optimization*, Vol. 12, No. 1, Pages 1-12

Laskari, E. C, Parsopoulos, K. E., and Vrahatis, M. N. (2002). Particle swarm optimization for integer programming, *Proc. IEEE Congress on Evolutionary Computation*, Pages 1582–7

Li ,Y., Yao, J., and Yao, D., (2004) Automatic beam angle selection in IMRT planning using genetic algorithm, *Phys. Med. Biol.*, Vol. 49, Pages 1915–32

Lin, B.M.T. (2001). Scheduling in the two- machine flow shop with due date constraints, *International Journal of Production Economy*, Vol.70, No.2, Pages 117-123

Liu, Bo-Fu, Chen, Hung-Ming, Chen, Jian-Hung, Hwang, Shiow-Fen, and Ho, Shinn-Ying, (2005). MeSwarm: Memetic Particle Swarm Optimization, *Proceedings of the 2005 conference on Genetic and evolutionary computation Washington DC*, USA, Poster Session: Ant colony optimization and swarm intelligence, Pages: 267 – 268

Messerschmidt, L., and Engelbrecht, A P., (2004). Learning to play games using a PSO-based competitive learning approach, *IEEE Trans. Evolutionary Computing*, Vol. 8, Pages 280–288

Moon, C., Kim, J., and Hur, S., (2002). Integrated Process Planning and Scheduling with minimizing total tardiness in multi-plants supply chain. *Computers and Industrial Engineering*, Vol. 43, No. 1-2, Pages 331-349, ISSN:0360-8352

Naso D., Surico, M., Turchiano, B., Kaymak, U. (2007), Genetic algorithms for supply chain scheduling:a case study on ready mixed concrete, *Erasmus Research Institute of Management – Report ERS-2004-096-LIS, European Journal of Operation Research.* Vol.177, No. 3, Pages 2069-2099

Nasr, N. and Elsayed, A. (1990). Job shop scheduling with alternative machines, *International Journal of Production Research*. Vol. 28, No. 9, Pages 1595-1609. 1990

Nishi, Tatsushi, Konishi, Masami. (2005). An autonomous decentralized supply chain planning system for multi-stage production processes, *Journal of Intelligent Manufacturing*, Vol.16, Pages 259-275

Palmer, G.J. (1996), A simulated annealing approach to integrated production scheduling, *Journal of Intelligent Manufacturing*, Vol.7 No.3, Pages 163-176, ISSN0956-5515 (Print) 1572-8145 (Online)

Parsopoulos, K. E., and Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization, *Nat. Comput*. Vol.1, Pages 235–306

Pepper, J.W. Golden, B.L. Wasil, E.A. (2002). Solving the traveling salesman problem with annealing-basedheuristics: a computational study, *IEEE Transactions on man,* *machine, and cybernetics- part A: Systems, Man and Cybernetics*, Vol.32, No.1, Pages 72-77, ISSN: 1083-4427

Pugachev, A., and Xing, L., (2002) Incorporating prior knowledge into beam orientation optimization in IMRT, *Int. J. Radiat. Oncol. Biol. Phys.*, Vol. 54, Pages 1565–74

Rahmat-Samii, Y., (2003) Genetic Algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics *17th Int. Conf. on Applied Electromagnetics and Communications*, Pages 1–5

Rai, R., Kameshwaran, S. and Tiwari, M.K. (2002). Machine-tool selection and operation allocation in FMS: solving a fuzzy goal programming model using a genetic algorithm. *International Journal of Production Research*, Vol.40, No.3, Pages 641-665

Robinson, J., and Rahmat-Samii,Y (2004). Particle swarm optimization in electromagnetics, *IEEE Trans. Antennas Propag*. Vol. 52, Pages 397–407

Robinson, J., and Rahmat-SamiiY. (2004) Particle swarm optimization in electromagnetics *IEEE Trans. Antennas Propa*g. Vol. 52, Pages 397–407

Salman A, Ahmad I and Al-Madani, S., (2002) Particle swarm optimization for task assignment problem, *Microprocess. Microsyst*. 26 363–71

Salman, A., Ahmad, I., and Al-Madani, S. (2002). Particle swarm optimization for task assignment problem, *Microprocess. Microsyst*. Vol. 26, Pages 363–71

Shi, Y., and Eberhart, R. C., (1998b) A modified particle swarm optimizer, *Proc. the IEEE Int. Conf. on Evolutionary Computation* (*Piscataway, NJ: IEEE*), Pages 69–73

Swarnkar R. and Tiwari M.K. (2004). Modeling machine loading problem of FMSs and its solution methodology using hybrid Tabu search and simulated annealing based heuristic approach, *Robotics and Computer Integrated Manufacturing*, Vol. 20, No.3, Pages 199-209

Tan, W. and Khoshnevis, B. (2004). A linearized polynomial mixed integer programming model for the integration of process planning and scheduling. *Journal of Intelligent Manufacturing*, Vol. 15, Pages 539-605

Ting, Tiew-On, Rao, M.V.C., Loo, C. K. and Ngu, S.S. (2003). Solving Unit Commitment Problem Using Hybrid Particle Swarm Optimization, *Journal of Heuristics,*Vol.9, No.6, Pages 507-520, ISSN1381-1231 (Print) 1572-9397 (Online)

Tiwari, M.K., and Vidyarthi, N.K. (2000). Solving machine loading problem in a flexible manufacturing system using a genetic algorithm based heuristic approach. *International Journal of Production Research*, Vol.38, No.14, Pages 3357-3384

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection, *Inf. Process. Lett*. Vol. 85, Pages 317–25

Weiqi, L.I., and Alidaee, B. (2002). Dynamics of Local Search Heuristics for the Traveling Salesman Problem, *IEEE Transactions on man, machine, and cybernetics- part A: Systems, Man and Cybernetics*, Vol. 32, No. 2, Pages 173- 184, ISSN: 1083-4427

Xie, Xiao-Feng, Zhang, Wen-Jun, Yang, Zhi-Lian.(2002). A Dissipative Particle Swarm Optimization, *Congress on evolutionary computing (CEC),*Hawaii, USA, Pages 1456-1461

Xing, L. (1999) A medical knowledge based system for the selection of beam orientations in intensity modulated radiation therapy (IMRT), *Int. J. Radiat. Oncol. Biol. Phys*. Vol. 45, Pages 246–247

Arena, P., Caponetto, R., Fortuna, L., Rizzo, A., and La Rosa, M. (2000), Self organization in non recurrent complex system. *Int. J. Bifurcation Chaos*, Vol.10, No.5, Pages1115–1125.

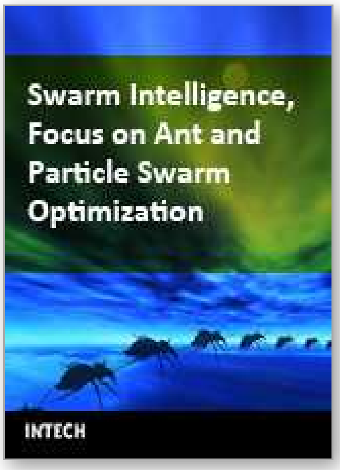
Determan, J. and Foster, A. (1999). Using chaos in genetic algorithm. *In Proceedings of the IEEE Conference on Evolutionary computation*, IEEE, Piscataway, New Jersey, Vol.3, Pages 2094–2101

Manganaro, G. and Pineda, D. G. J. (1997). DNA computing based on chaos. *In Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, New Jersey, IEEE, Pages. 255–260

Suganthan, N. (1999). Particle swarm optimizer with neighbourhood operator. *In Proceedings of the IEEE International Conference on Evolutionary computation*, IEEE, Piscataway, New Jersey, Vol. 3, Pages 1958–1962

Nozawa, H. A (1992). A neural network model as globally coupled map and application based on chaos, *Chaos*, Pages 377–386.

Wang, L. and Smith, K. (1998). On chaotic simulated annealing. *IEEE Trans. Neural Networks*, Vol.9, Pages 716–718.



Swarm Intelligence, Focus on Ant and Particle Swarm Optimization

Edited by FelixT.S.Chan and Manoj KumarTiwari

ISBN 978-3-902613-09-7

Hard cover, 532 pages

Publisher I-Tech Education and Publishing

Published online 01, December, 2007

Published in print edition December, 2007

In the era globalisation the emerging technologies are governing engineering industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from the nature and implant it in the engineering sciences. This way of thinking led to emergence of many biologically inspired algorithms that have proven to be efficient in handling the computationally complex problems with competence such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms the present book on "Swarm Intelligence: Focus on Ant and Particle Swarm Optimization" aims to present recent developments and applications concerning optimization with swarm intelligence techniques. The papers selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. In addition to the introduction of new concepts of swarm intelligence, this book also presented some selected representative case studies covering power plant maintenance scheduling; geotechnical engineering; design and machining tolerances; layout problems; manufacturing process plan; job-shop scheduling; structural design; environmental dispatching problems; wireless communication; water distribution systems; multi-plant supply chain; fault diagnosis of airplane engines; and process scheduling. I believe these 27 chapters presented in this book adequately reflect these topics.