PARTICLE SWARM OPTIMIZATION TO SOLVE LEVEL SCHEDULES IN MIXED MODEL ASSEMBLY LINE

R. Senthilkumar[1], P. Dhiravidamani[2***], A.S. Ramkumar[3] and M.Chandrasekaran[4]

[1.]*Maharani Polytechnic College, Dharapuram, TN, India*

[2.]*KSR College of Engineering, Thiruchengodu, TN, India*

[3]*Amrita College of Engineering and Technology, Nagercoil, TN, India*

[4] *Vels Institute of Science, Technology and Advanced Studies, Chennai*

E-mail:pdhiravidamani1975@gmail.com

Abstract

In this paper presents level schedules in mixed model assembly line is solved by the application of particle swarm optimization algorithm. The Production sequence is solve through optimize the level schedule by minimize the material usage rate and number of setups .By using simulated annealing this seed sequence is optimal[2016] and this paper results obtained from simulated annealing [2016] are compared with results of the particle swarm optimization algorithm. The investigation of SA-PSO is use to find the priorities the utility of the assembly line. The Meta heuristic is promising to guide the search space into a feasible solution space for the sequence optimization. The stochastic based optimization method PSO is come from the principles of birds and fish. Its function is based optional solutions on population of particle solution, randomly initialized and freely flight across the three dimension search space, during the flight every particles are update their position and velocity based by their own experience and neighbors experience. The PSO approach results are better than previous results obtained by Miltenburg algorithm and simulated annealing [2016] for problems set considerable.

Keywords: Mixed-Model Assembly Line, Level Schedule, Just-in-Time Manufacturing, Simulated Annealing and Particle Swarm Optimization.

1. Introduction

Now-a-days many industries are using mixed model assembly line [MMAL] due to variety of product models are to be assembled. In this production line for maximizing the line utilization of MMAL, determining the sequence of the product model is necessary. In a JIT production system necessary product are to be produce at necessary time with minimum stock is handled. The MMAL is a small level sequence problem in JIT production line. There are two types of problems identified in MMAL are line balancing problems, level scheduling. In line balancing problem is long term decision problems. The reasonable distribution of the operating units is the balancing problem. Level schedule is the products are to be sequence. It is short term decision problem. For effective utilization of MMAL, implementations of the objective functions are to be solved.

1. Determining the cycle time.
2. Determining the number of sequence of station on the line.
3. Line balancing.
4. Determining the sequence for producing different product on the line.

These objectives are solved by various optimization methods like traditional optimization methods and heuristic methods. In traditional optimization methods are branch and boundary algorithm, goal chasing algorithm etc. Due to large number of possible solutions, it is difficult to find an optimal solution by this method .It is stock with local optimization. Global optimization will get by the heuristic methods like Genetic algorithm (GA), Simulated annealing(SA) and Particle swarm optimization algorithm (PSO), Tab search (TS) and Ant colony algorithm(ACO).

The multivariate combinatorial optimizing problems are difficult to solve. These are solved by Heuristics methods. A Genetic algorithm is a search method based on genetic behavior of chromosomes. Initial GA solution is making an initial population of Chromosomes. The basic Operations of GA are selection, recombination and mutations used to improve the search Function. PSO is improve from the habit of birds and fish when it searching food. After finding the food, they informed to others. Ant colony optimization is collecting food by the behavior of ant. A phenomenon is used to communicate for each other. When ant moves through the phenomena and find the food. Tabu search is short memory of previous solutions. In the Tabu search the result of local optima is recorded in the Tabu list. Searching move is forbidden unless its objective function is satisfied. The procedure is repeated until it reaches the stopping criteria.

It is proposed that the sequence obtained from Miltenburg algorithm is considered as seed. From this seed, the sequence has altered randomly. The utility and setup has been taken as objective function value to optimize the level schedules in JIT production sequence. For this objective function value by giving weightage for each utility and setup as $E=W_u$ U+ W_s s. The weights $W_s$and W_u used for the objective function values are to emphasis the importance the setup and utility. The weights are determines as W_s=R/number of setups from initial solution,W_u=R/numberof utility from initial solution=constant=1000. Three types of heuristics used in this paper as per the varying the importance of setup and utility. The objective function value for different weightage has been found from heuristic 1 to 3.

The simulated annealing algorithm and particle swarm optimization algorithm has utilized to find the optimum function value and optimum sequence. In section 2, the literature survey of mixed model assembly line and particle swarm optimization algorithm has been presented. In section 3, the detailed description of the mixed model assembly line model and Miltenburg algorithm has been discussed. In section 4, discussed the simulated annealing algorithm(2016),in section 5 discussed about particle swarm optimization, Similarly in section 6,discussed the experimental results and compared with results obtain by simulated annealing(2016) .

2. Literature Survey

Monden (1983) discusses the leveling and balancing schedule in Toyota production system.He obtainsthe sequence of models in mixed model assembly line for different level of load and usage of parts.Miltenburg (1989) find the best production schedule by algorithm1.He obtained algorithm 2,algorithm3, heuristics 1,2,finaly obtained low variation optimal scheduling algorithm Senthilkumar&Ramkumar(2016) presents simulated annealing algorithm for solving the production sequence in level schedule optimization problem . McMullen (1998) find a sequence when minimization of both setup and utility by uses techniques Tabu search.McMullen (2000) shows that genetic algorithm gives best solution than simulated annealing and Tabu search.Ponnambalam(2003) investigated the performance of genetic algorithm in MMAL sequencing problem.He uses the parents-stratum niche cubicle performs better than genetic algorithm .

Muhammad ZainiMatondang (2010) studied about the advantage ofgenetic algorithm and it increases the multi objective assembly line problems by soft computing and hybrid systems. ParamansChutima, PalidaChimkleix(2012) solved multi-objective two sided MMAL Problems byPSO algorithm with negative knowledge. This knowledge of poor solutions is part or new solutions in the next generation. Alirezaalfi,(2012) obtained novel PSO, dynamic inertia weight

PSO to cope with the online system parameter problems to be identified . In this algorithm convergence speed of PSO, improved the efficiency ,every particle inertia weight is updated dynamically based on the feedback taken from the fitness of the best particle of previous position.

Qiao Ying Dong,Jiansha Lu, Yuankungui,(2012) solved the scheduling the separation in the traditional serial production planning by modified discrete particle swarm optimization. The model and algorithms are verified by experiment. Qiaoying dong, ling qin,(2001) used the PSO to solve the MMAL sequencing adoptive escape scheme to enhance the diversity of particles.

Xiao Fengxie,Werjuwzhang,(2012) developed the self organization of dissipate structure by dissipative PSO. Introduced of the negative entropy to construct an opening dissipative system that is far from equilibrium with better fitness by irreversible evolution process.

Xiaolongxu, HanzhangRong, Marcello Travalt (2016) proposed a CS-PSO algorithm for solving combinatorial optimization problems by combine chaotic search method with PSO. Lukasz Strak (2017) applied discrete PSO to solve the dynamic travelling salesman problem in planning, logistics and chip manufacturing. The newly generate DTSP have differ in the number and size of the changes is tested by the DPSO algorithm.The convergence of the DPSO is investigated and compared with version without a pheromone memory.

3. Mixed Model Assembly Line

In mixed model assembly line scheduling of parts are difficult due to large lot of parts assembled. When demand rate of products are increase certainly, it can achieve by minimizing the variation in the usage of each parts and also it is important of minimizing setup. A setup is required each time of two consecutive items in the production sequences are different. An objective function value is obtained by composite measure of utility and setup. The various weightage rates are applied to utility and setup then obtain the optimum weightage of function value and sequence.

3.1 Miltenburg Algorithm

N products with demand $d_1, d_2, \dots, d_n$

$D_T = \sum_{i=1}^{n} d_i$unitsare to be produced.

$r_1 = \frac{d_i}{D_T}$,is theproposition of product ‘i’ demand to the total demand.

The objective is to schedule the assembly liner_i is closed to product i is proportion to total production.

$S_{i,k}, i = 1,2, \dots, D_T, k = 1,2, \dots, D_T$,where$S_{i,k}$ is either 0 or 1 be a production schedule.

If $S_{i,k} = 1$ then product i will be produced during stage k.

$\sum_{i=1}^{n} S_{i,k} = 1$, for all k, because only one product can be produced during each stage.

$x_{i,k} = \sum_{i=1}^{k} S_{i,k}$be the total production of product i over stages 1 to k.

Clearly, $x_{i,k}$is an integer of non-negative and$\sum_{i=1}^{n} x_{i,k} = k$, for all k. The objective should one of the following,

Minimize $\sum_{k=1}^{D_T} \sum_{i=1}^{n} \left(x_{i,k} - r_i\right)^2$ (1)

$\sum_{i=1}^{n} x_{i,k} = k$, $k = 1,2, \dots, D_T$, (2)

$\sum_{i=1}^{n} x_{i,k} = k\, r_i$ (3)

The constraints should besatisfy and the objective function is equal to zero.

$\sum_{i=1}^{n} x_i$, k=$\sum_{i=1}^{n} k r_i = k, \sum_{i=1}^{n} r_i = k$ (4)

Algorithm

Find the point X to nearest point M, where $\sum_{i=1}^{n} m_i = \sum_{i=1}^{n} x_i = k$

- Calculate $k = \sum_{i=1}^{n} x_i$
- Find the nearest non –negative integer m_i to each coordinate x_i,
- That is find m_i and $m_i - x_i \leq \frac{1}{2}$, $i = 1,2, \dots, n.$
- Calculate $k_m = \sum_{i=1}^{n} m_i$
- (a) If $k - k_m = 0$ stop. The nearest integer point is $M = (m_1, m_2, \dots, m_n)$
- (b) If $k - k_m > 0$ go to step 5
- (c) If $k - k_m < 0$ go to step 6
- Find the coordinate x_i with the smallest $m_i - x_i$ increment the value of this $m_i; m_i \rightarrow m_i + 1$ Go to step 3
- Find the coordinate x_i with largest m_i-x_i decrement the value of this $m_i; m_i \rightarrow m_i - 1$ Go to step3

3.2 Objective Function

3.2.1. Minimizing the setups

The number required setup $S = \sum_{k=1}^{DT} S_k$ (5)

Where, k= sequence position index. The setup is required when k position is varying from position k-1. And $S_k = 1$, otherwise 0.The setups are assumed to sequence independent, so that the machine does not depend on which other product preceded it on that machine.

3.2.2 Minimizing the utility

$$U = \sum_{k=1}^{DT} \sum_{i=1}^{n} \left(x_{i,k} - k.\frac{d_i}{D_T} \right)^2 \quad (6)$$

In the production sequence usage rate of material is sensitive. In the assembly line different products are made, it is important that usage of material is constant.

3.3 Composite Objective Function Value

In the production sequence, the objective function value is then determined with utility and setup where W_U is the weightage of usage rate and W_s is the weightage of number of setups. The composite functions

Min $E = W_s S + W_U U$ (7)

3.3.1. Sequencing heuristics used:

AsMcmullen(2000) uses the various heuristics according to different weightage of utility and setup.The different objective functions are

Heuristic A

Min E=S (8)

This heuristic gives that required number of set ups is minimized in the product sequence. It does not require minimum setup,it does not require heuristics, it get from inspection.

Heuristic B

Min E=U (9)

This heuristic gives that minimizes the material usage rate. This objective is addressed by Ding and Cheng's heuristic (1993), then it is simplified by Miltenburg sequence(1989).

Heuristic 1

$$\text{Min } E = 14.2755S + 1U \quad (10)$$

From the sampling the objective function values coefficients (14.2755 and one) are obtained. The both utility and set up are gives equal contribution from the sampling. The coefficients are derived by McMullen (1998).

Heuristic 2

$$Min\ E = 3\ X\ 14.2755S + 1S \quad (11)$$

In this objective function value, setup is three times importance than utility. The setup is minimizing by three times.

Heuristic 3

$$Min\ E = 14.2755S + 3U \quad (12)$$

The utility is minimizing three times than setup. The utility is three times importance than setup. the first two heuristics the objective function value does not require to minimize, because it directly comes from minimized by their own setup and utility. Three types of heuristics are used as

$$E_1 = U + 3S, E_2 = U + 3 * 14.27Sand\ E_3 = 3U + S$$

.

4. Simulated Annealing

Simulated annealing is a search technique that exploits an analogy between the ways in which a Meta cools into minimum energy crystalline structure. Simulated annealing is a method for obtaining good solutions to difficult optimization problems. The concept of annealing is introduced by KirkPatrick.S.and Gelatt.C.D.Vecchi. In simulated annealing, solid is melt by increasing the temperature then slowly cool it so it crystallizes into perfect lattice. Solution – states of the physical system, objective function –energy control parameter- temperature.

In simulated annealing the initial state of a thermodynamic system energy E and temperature T the change in energy ΔE. If the change in energy is negative then the new state is accepted. If the change in energy is positive then it is accepted with$e^{-\frac{\Delta E}{T}}$. The processes are repeated for sufficient times go to give good sampling statistics of current temperature. Temperature is decremented until the final temperature or number of iteration or sufficient computational time is attained. In this parameter selection is very important. The initial temperature, final temperature, number of iterations is three important parameters for the success of working the simulated annealing procedure. If high initial temperature is chosen, it takes number of iterations for convergence. If a small initial temperature, the search is not adequate to the search space before finding the time optimum. The advantage of simulated annealing is the ability to move from local optimization thus the ability to find the global optimum is not related to the initial condition. The disadvantage of simulated annealing is the subjective nature of choosing the configuration parameters.

4.1 Annealing Procedure

4.1 .1 Initial Temperature

In Connolly's method (1990) the initial and final temperature were determined by information obtained the trial to annealing process. In this trial, a certain number of random moves were performed to record the results change in the objective function. From this results, the minimum temperature

$$T_0 = \Delta E_{min} + \frac{1}{10}(\Delta E_{max} - \Delta E_{min}) \quad (13)$$

$$T_0 = \Delta E_{min} \quad (14)$$

4.1.2 Decrementing Temperature

One of the major issues is related to the annealing schedule to cool the temperature during the annealing process. Varies methods are used to reduce the temperature as follows in table 1

Table 1 Temperature set

Wilhelm & War method(1987)	Golden &Skiscin method (1986)	Connolly method (1990)	Vilarinho&Simaria method(1992)
$T_{i+1} = \alpha T_i,$ $0 < \alpha < 1$	$T_{i+1} = T_i - \frac{T_0}{25}$ $i = 0, \ldots .25$	$T_{i+1} = \frac{T_i}{1+\beta T_i},$ $\beta = \frac{T_0 - T_f}{M T_0 T_f}$	$T_{k+1} = T_k - T$

We select Connolly method because it is clear that when the temperature is too high a lot of uphill moves are accepted, when the temperature is too low, the probability is falling into a local minimum. The temperature should be between these two extreme that the temperature is high but cooling rate is low. The Connolly were designed based on this idea so we adept this method.In Connolly method, during these trials the initial temperature T_o and final temperature T_f are determined and M refers to the number of pair wise exchanges examined

$$T_{i+1} = \frac{T_i}{1+\beta T_i}, \quad \beta = \frac{T_0 - T_f}{M T_0 T_f} \quad (15)$$

$$M = \frac{n(n-1)}{2} \quad (16)$$

In this equation parameter β usually has a small value and there after the temperature reduction proceeds slowly, n is the number of demand. The Algorithm perform depends upon the cooling rate than individual temperature for better result, reduction rate should be slower in middle temperature range.

4.1.3 Random number generation

A significant component of an SA code is the random number generator, which is used both for generating random changes in the control variables and for the increase acceptance test. It is important, particularly when tacking large scale problems requiring thousands of iterations, so that the random numbers generator used have good spectral properties. We use the Microsoft excel VBA procedure method is inbuilt into the program for find the random number generation

4.1.4 Number of iteration

A constant number of iteration at each temperature is generally employed. Another method is only one iteration at each temperature but decrease the temperature very slowly. The iterations at each temperature is proportional to $n = t/(t+1)$. An alternative is dynamically changing the

number of iterations as the algorithm progress. In this paper, total number productsareused as the number of iteration.

4.1.5 Stopping criteria

Stopping criteria is total number of iterations have been completed or fixed amount of execution time. The stopping criteria can either be a suitably low temperature or when the system is "frozen" at the current temperature (i.e. no better or worse moves are being accepted). In our paper after reach the final temperature, the process will stop.

4.2 Parameter Set

The initial and final temperaturesare determined by information obtained in trail to the annealing process. In this trail certain number of random moves is performed to record the resulting changes in the objective function. From this result, the minimum value of ΔE_{min} and maximum value of $\Delta E_{max}$are to be final. The initial temperature is set as

$$\mathrm{T}_0 = \Delta E_{min} + \frac{1}{10}(\Delta E_{max} - \Delta E_{min})\text{and} \quad (17)$$

final temperature $\mathrm{T}_0 = \Delta E_{min}$

Another annealing schedule is how to cool the temperature during the annealing process

$$T_{i+1} = \frac{\mathrm{T_i}}{1+\beta \mathrm{T_i}}, \quad \beta = \frac{\mathrm{T_0}-\mathrm{T_f}}{M_0 \mathrm{T_0} \mathrm{T_f}}, M_0 = \frac{\mathrm{n(n-1)}}{2} \quad (18)$$

where, M refers to the number of pair wise exchange examined

$$T_{i+1} = \text{next temperature to be set, n} = \text{no of demand.}$$

4.3. Proposed Simulated Annealing Algorithm

The flow chart for proposed simulated algorithm as shown in figure 1. Initial sequence is generated by Miltenburg Algorithm. Swap this sequence by random algorithm and its objective function value. And choose initial temperature, temperature reduction factor, and final temperature and select the number of iteration. Select the objective function is initial Miltenburg Algorithm sequence as initial energy state (E_O). Find the randomizing select the another energy state (E_1)

Find the difference between the two energy states $\Delta E = E_1 - E_0$.Check whether$\Delta E < 0$. If yes store the energy and find the randomly energy state. If no generate randomly X ϵ U (0, 1) and Check, whether $X < e^{-\frac{\Delta E}{T}}$.If yes store the energy state otherwise go for iteration. The above function is repeating do until the all the iteration and reduce the temperature according to reduction factor. Continue the above procedure until reach up to final temperature.

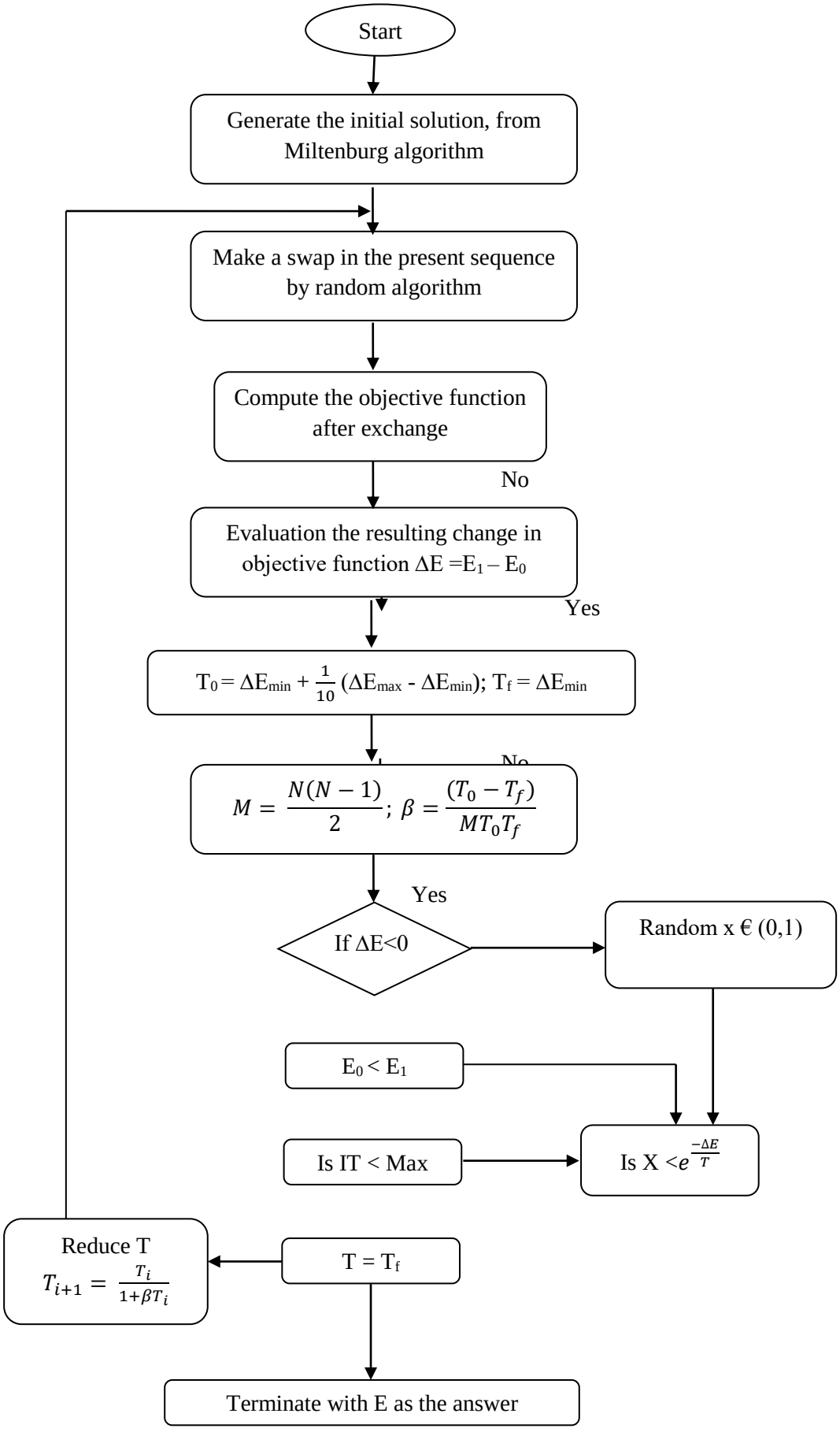

Fig.1 Proposed Simulated annealing Algorithm

5. PSO Algorithm

Swarm intelligence is find solution from explore search space that inspiration from biological like swarming, flocking, herding.Birds and fishes adjust their position according to their own experience and their neighbors while searching food, Particle swarm optimization (PSO) is a heuristic optimization method introduce by James Kennedy and Russell C. Eberhart in 1995.Particle swarm optimization is based on the metaphor of social interaction such as bird flocking and fish schooling. This algorithm is fast convergence and easy to implement and it is population based optimization tool. It can easily introduce and implement with few parameters to optimization problems. In this method each member called particle, each particle has their own position, velocity. While moving the particles update their position and velocity in the search space by their best experience and neighbors are memorized. Each particle flies the search space to find the near optimization solution and its position represents a potential solution for the problem.

Bird flocking optimizes a certain objective function. Each function value is called particle. Each particle knows its best value called p_{best} value and position. This information is analogy of personal experience of each particle. The best particle among the group of p_{best} is called g_{best}. This information is analogy of knowledge of around the position of other particles. The particles change its position according to the principles of keep its inertia, to change its most optimist position, change to the swarm most optimist position.

$X_{i,k=}$position of particle

$V_{i,k=}$velocity of P_{best} position of particle

$P_{best} =$ P_{best} position of particle

$G_{best} =$ g_{best} position of particle

$$X_{i+1} = X_i + V_{i+1} \tag{19}$$

$$V_{i+1} = \omega V_i + c_1 r_1 (p_{best} - X_{i,k}) + c_2 r_2 (g_{best} - X_{i,k}) \tag{20}$$

ω =inertial coefficient is between 0.8 to 1.2

c_1=cognitive coefficient is between 0 to 2

c_2=social coefficient is between 0 to 2

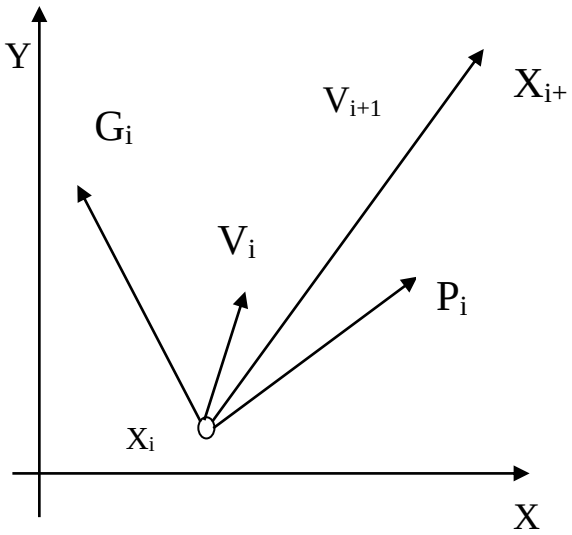

Fig.2 Explore search space

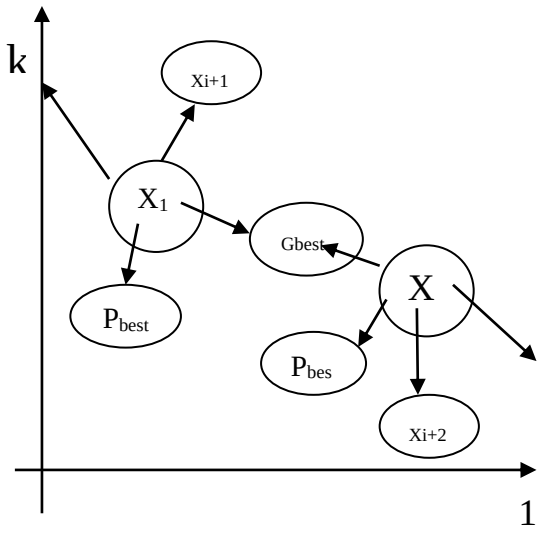

Fig.3 Exploit search space

From the figure 2, the search space is X,Y, the particle X_i is at a point, while its velocity is V_i when its velocity is changed to V_{i+1},then its position is change to X_{i+1},its best position is P_i and its global

best position is G_i ,then it adjust its velocity as its position, it check its best and global best position and updated. From the figure 3, Exploit - Individual particles (1and 2) are accelerated toward the location of the global best solution (G best) & location of their own personal best (P best) in the k dimensional space.(ability to perform more refined local search)-Explore the unknown space(ability to explore regions in global search space)

5.1 PSO parameters

The parameter setup is important to performance of the algorithm. The important parameters are velocity components, number iteration, swarm size, acceleration coefficients

5.1.1 Velocity components

The first term ωVi is the inertia compound, it can accelerate to keep the particle moving same direction as original, lower value is speedup the convergence of the swarm to optimum and higher value is encourage exploration of the entire search space. The second term $c_1 r_1 (p_{best} - X_{i,k})$ is called cognitive compound, which measures the performance of the particles relative to past performances. This component looks like an individual memory of the position that was the best for the particle. The effect of the cognitive component represents the tendency of individuals to return to positions that satisfied them most in the past. The third term $c_2 r_2 (g_{best} - X_{i,k})$ is called social component which measures the performance of the particles relative to a group of particles or neighbors. The social component's effect is that each particle flies towards the best position found by the particle's neighborhood. In this paper ω value taken is 0.8.

5.1.2Iteration numbers

The number of iterations to obtain a good result is also problem-dependent. A too low number of iterations may stop the search process prematurely, while too large iterations has the consequence of unnecessary added computational complexity and more time needed. In this paper total number product is used as the number of iteration.

5.1.3Swarm size

Swarm size is the population size, if the swarm size high, then number of iteration is low, if the swarm size is low then number of iteration high, generally swarm size is 5,10,15,20 as like. The large number iteration need more computational time and complexity, less swarm size does not give good optimum result, so swarm size should fix by trial run.

5.1.4Accelerationcoefficients

The acceleration coefficients are c_1,c_2 along with random number r_1,r_2 maintain the influence in cognitive and social components, The constant c_1 expresses how much confidence a particle has in itself, while c_2 expresses how much confidence a particle has in its neighbors, When , $c_1 = c_2$ then all particles continue flying at their current speed, and all particles are attracted towards the average of p_{best} and g_{best} . When $c_1 > 0$ and $c_2 = 0$, all particles are independent (p_{best}).when $c_2 > 0$ and $c_1 = 0$, all particles are attracted to a single point in the entire swarm(g_{best}). $c_1$and c_2 are the static behavior and wrong selection of c_1, c_2 cause divergent and cyclic behavior. In this paper 0.8 are used for $c_1$and c_2

5.2Algorithm for Basic PSO

Assuming n particles in the swarm in the k dimensional search space, the i-thparticle is represented as :

$$X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,k})$$

The best previous position giving the best fitness value is given by

$P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,k})$

The rate of change (velocity) of the particle is given by

$$V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,k})$$

Step 1: Initialization - positions & velocities of particles are randomly generated

Step 2: Velocity updating - Update the velocities of all particles at time *k* (current iteration) using the particles objective or fitness values.

$$V_{i+1} = \omega V_i + c_1 r_1 (p_{best} - X_{i,k}) + c_2 r_2 (g_{best} - X_{i,k})$$

Step 3: Position updating - The Position of each particle is updated using its velocity vector as follows: $X_{i+1} = X_i + V_{i+1}$

Step 4: Memory updation – Update $P_{i,best}$ and $g_{i,best}$ when condition is met,

$P_{i,best} = p_i, if\ f(p_i) > f(p_{i,best})$

$g_{i,best} = p_i, \quad if\ f(g_i) > f(p_{i,best})$

Step 5: Stopping Criteria – The algorithm repeats steps 2 to 4 until certain stopping conditions are met, such as a pre-defined number of iterations. Once stopped, the algorithm reports the values of$g_{i,best}$ and $f(g_{i,best})$as its solution.

Pseudo code

Generate initial sequence

Calculate objective function P_1, P_2

Initialize parameters, τ, C_1, C_2, p_i

Set $v_1 = 0, p_i = P_1, P_{best} = P_1, G_{best} = P_{best},$

$N = N_{Max}$

Check $P_{best} < p_2$

Check $G_{best} < P_{best}$

Randomly generate sequence

Calculate objective function P_3,

Update velocity

$V_2 = \tau V_1 + c_1 r_1 (P_{best} - P_1) + c_2 r_2 (G_{best} - P_1)$

Update position $P_2 = P_1 + V_1$

$P_{best} = p_2$, if $f(P_2) > f(P_1)$

$G_{best} = p_2$, if $f(P_2) > f(P_1)$

Check $N = N_{Max}$

Randomly generateanother sequence

Calculate objective function P_4,

Update velocity

$V_3 = \tau V_2 + c_1 r_1 (P_{best} - P_2) + c_2 r_2 (G_{best} - P_2)$

Update position $P_3 = P_2 + V_2$

Check P_{best}<p_4, if $f(P_2) > f(P_1)$

Check G_{best}<P_{best},if $f(P_2) > f(P_1)$

Stop with $G_{best} = p_4$ with corresponding sequence

Figure 4 Proposed Pseudo code

From the figure 4,shows proposed Pseudo code in which explains the steps of PSO algorithm. Generate the initial solution from Miltenburg algorithm and find the objective function value(z),set the values of ω,c_1,c_2,N,n from the initial trials. initial velocity is equal to zero and initial g_{best} is equal to initial objective function value(z), g_{best} is equal to $P_{best}$and set initial iteration , generate initial swarms, find its sequences and its objective function values and find P_{best}, g_{best} and compare the P_{best}, g_{best} and find whether it is minimum or not, if it is less than previous then set new z is new P_{best}, g_{best} and find V_{x+1} and X_{i+1} go to next iteration, repeat the steps up to maximum iteration, stop the process up to last iteration.

6. Experimental Design and Discussion of Results

The three problem sets are taken from Mcmullen (2001) for analysis. The problem set 1 contains 7 types of problems with total demand is 10 each. The solution obtained is 2524 to 4104. The problem set 2 contains 9 types of problems with total demand is 12 each. The solution obtained is 4300 to 5680.The problem set 3 contains9 types of problems with total demand is 15 each. The solution obtained is 8524 to 11104.The Miltenburg algorithm; Random generation algorithm and simulated annealing algorithm arecoded inMicrosoft Excel in macro and executed on anIntel processor at 2 GB under windows XP using 250 MB of RAM.The problem set 1 has solved with 7 types problem for all problems to all demand of product is equal but demand of each product is different for each problem. Thethree types of heuristicshave been solved. The Heuristics1 have minimum weightage for both setup a usage rate. Heuristics 2 have a composite value of 14.755 for setup. The coefficient used from the sampling, so the setup and material usage made equal contribution to the objective function. Heuristics 2 is weighted three times important of setup than minimum usage rate. Heuristics 3 is three times of usage rate than setup. So the importance of usage rate is three times than minimum setup. The all the problems with heuristics 1 to 3 are solved by Miltenburgalgorithm and simulated annealing algorithm. The two methods are compared by RPD.RPD $= \frac{\text{PSOA (OR) SA solution} - \text{MA solution}}{\text{MA solution}} * 100$.In Heuristics1($W_u$=1,$W_s$=14.27) the weightages of setupis 14.27and utility are 1, denote as w1.Heuristics2(W_u=1,W_s=42.81) the weightages of setup is 42.81and utility are 1, denote as w2. Heuristics3(W_u=3,W_s=14.27) the weightages of setup is 14.27and utility are 3 denote, as w3.All the results obtained by heuristics 1 to 3 are tabulated in table 2 to4and comparisons is tabulated in table 5to 8. For all the problem sets the solutions obtained by PSO shows minimum compared with Miltenburg Algorithm and Simulated Annealing. The number of setup is equal or small reduction. In heuristics 1 shows the RPDis 19.58 to 30.82% inPSO and RPD of 24 to 38.47% and number or setup is low in SA compare with Miltenburg. In PSO number of setup is requires is 5 to 9 andin SA number or setup is requires is 4 to 7 where as in Miltenburgare9. The objective function inPSO is 98.50 to 80.35 andSA is 85 to 80.35 where as in Miltenburg are 136.47 to 106.89. In this number or setup and usage rates is balance inPSO and SA compare with Miltenburg. In heuristics 2the RPD is 35.38 to 35.19 inPSO and in SA the RPD is 49.35 to 35.45, Miltenburg the number of setup is 9. The objective function value in PSO is 227.10 to 198.24 and in the objective function in SA is 202.2 to 198.30 whereas in Miltenburg are 391.23 to 306.7. So the setup and usage rate is much balance inPSO and simulated annealing. but in heuristics 3, the RPD is

38.22 to 38.13 in pso andin the SA RPD is 27.85 to 13.75 and The objective function value in PSO is 107.59 to 88.45 and the objective function value in SA is 127.65 to 98.67 but in Miltenburg is 152.47 to 104.37.In problem set 1, heuristic 1 shows high RPD. But the utility is more and setup is less. In heuristic 1 and 3 shows low utility and setup is high. In heuristic 2, balancing the setup and utility as well as the RPD is high. The Problem type G (6 1 1 1 1) shows in PSO is utility 9 and setup 5,minimum objective function value is 80.35,high RPD is 24.83 andthe optimum sequence is C,A,A,A,A,D,B,E,A,A. In problem set 2 the heuristic 3 shows high RPD. But the utility is more and setup is less. In heuristic 1 and 4 show low utility and setup is high but RPD is low. But in heuristic 2 shows the RPD is moderate as well as balancing the utility and setup. In problem type I (8 1 1 1 1) shows utility by PSO is13.11 and setup is5 and objective function value is 84.46 and RPD is 30.82. So the obtained optimum sequence is D,A,A,A,A,A,A,C,B,E,A,A. In problem set 3 the heuristic 1 show high RPD of 35.16. But the utility is more and setupis less. In heuristic 1 and 3 show low utility and setup is high RPD is low. But in heuristic 2 shows the RPD is moderate as well as balancing the utility and setup. In problem type I (11 1111) shows by PSO is the utility12.88 and setup 6 and objective function value is 98.50and RPD is 20.80. So the obtained optimum sequence is C,A,A,A,A,A,B,D.A,A,A,A,E,A,A. So it has been concluded that from the above problem sets analysis, the heuristic 2 is work well than other heuristic by optimize the optimum function RPD and balancing the utility setup of PSO with compare with Miltenburg algorithm and SA.

Table 2 Solution for Problem Set 1

Miltenburg Algorithm						Simulated Annealing Algorithm					Pso Algorithm				
Problem set	Scheme	u	S	z	Sequence	u	s	z	Sequence	RPD	u	s	z	Sequence	RPD
A (22222)	W1	7.99	9	136.47	12345 12345	14.79	7	85.10	33244 51215	37.64	9.99	8	85.08	5 3 4 2 2 1 3 1 4 5	37.66
	W2	7.99	9	393.43	12345 12345	28.02	4	199.30	44331 12255	49.35	28.00	4	199.24	3 3 2 2 4 4 1 1 5 5	49.36
	W3	7.99	9	152.47	12345 12345	14	6	127.65	11223344 55	16.27	14	6	127.62	3 5 4 4 2 2 1 1 3 5	16.29
B (32221)	W1	5.69	9	134.17	1234152 341	29.52	4	86.60	52233 44111	35.45	17.5	5	85.5	5 1 1 4 4 2 2 3 3 1	36.27
	W2	5.69	9	391.13	1234152 341	30.92	4	202.20	33522 44111	48.30	29.5	4	200.74	5 4 4 3 3 2 2 1 1 1	48.67
	W3	5.69	9	145.57	1234152 341	13.31	6	125.55	13544 22311	13.75	12.3	6	122.52	4 1 1 2 2 3 3 4 5 1	15.83
C (33211)	W1	5.4	9	133.07	1234125 312	14.95	5	86.30	52443311 12	35.14	29.2	4	86.20	4 3 3 1 1 1 5 2 2 2	35.22
	W2	5.4	9	434.08	1234125 312	24.95	5	239.05	52443311 12	44.92	31.00	4	202.24	5 1 1 1 3 3 4 2 2 2	53.40
	W3	5.4	9	144.67	1234125 312	10.81	6	118.05	12233541 12	18.40	10.6	6	117.42	1 5 2 2 3 3 1 1 4 2	18.83

D (42211)	W1	4.9	9	133.37	1231451231	13.72	5	85.07	1142233511	36.21	13.5	5	85.02	4 1 1 5 2 2 3 3 1 1	36.25
	W2	4.9	9	390.33	1231451231	30.92	4	202.2	5422331111	48.19	30.9	4	202.14	4 5 2 2 3 3 1 1 1 1	48.21
	W3	4.9	9	143.179	1231451231	13.70	4	112.47	1143322511	21.14	17.1	5	88.45	5 1 1 1 4 2 2 3 3 1	38.22
E (43111)	W1	5.8	9	134.27	1231241521	11.72	5	84.37	3115222411	37.16	13.2	5	84.55	5 1 1 2 2 2 3 4 1 1	37.02
	W2	5.8	9	391.23	1231241521	30.22	4	201.5	3452221111	48.49	30.2	4	201.44	3 5 4 2 2 2 1 1 1 1	48.51
	W3	5.8	9	145.87	1231241521	13.20	5	110.97	5112233411	23.92	13.2	5	110.90	3 1 1 5 2 2 2 4 1 1	23.97
F (52111)	W1	5.4	9	133.87	1231241521	11.02	5	82.37	4111225311	38.4	11	5	82.30	3 1 1 1 2 2 5 4 1 1	38.52
	W2	5.4	9	390.80	1231241521	30.42	4	201.7	4352211111	48.38	30.4	4	201.60	4 5 3 2 2 1 1 1 1 1	48.41
	W3	5.4	9	144.67	1231241521	11	5	104.37	4111225311	27.85	11	5	104.30	3 1 1 1 2 2 4 5 1 1	27.90
G (61111)	W1	7.00	7	106.89	1121341151	8.99	5	80.35	2111534111	24.80	9	5	80.35	3 1 1 1 1 4 2 5 1 1	24.80
	W2	7.00	7	306.78	1121341151	27.02	4	198.30	4523111111	35.30	27	4	198.24	4 2 3 5 1 1 1 1 1 1	35.38
	W3	7.00	7	120.37	1121341151	9	5	98.37	3111145211	18.27	9	5	98.35	4 1 1 1 5 2 3 1 1 1	18.29

Sequence (A-1,B-2,C-3,D-4,E-5)

Table 3Solution for Problem Set 2

Miltenburg Algorithm						Simulated Annealing Algorithm					Pso Algorithm				
Problem set	Scheme	u	s	Z	Sequence	u	s	z	Sequence	RPD	u	s	z	Sequence	RPT
A(33222)	W1	7.08	11	164.05	123451234512	22.58	6	108.08	2211334455122	34.11	22.58	6	108.2	1122334455122	34.04
	W2	7.08	11	478.10	123451234512	25.24	6	282.16	2244113355122	40.98	25.25	6	282.10	2 2 3 3 1 1 4 4 4 5 5 1 2	40.99
	W3	7.08	11	178.22	123451234512	12.58	8	151.910	5422115345122	14.76	12.58	8	151.900	3 5 2 2 1 1 4 4 3 5 1 2	14.77
B (44211)	W1	6.19	11	163.16	123412451231 2	23.19	6	108.81	5111222433122	33.31	23.10	6	108.80	5 1 1 1 2 2 2 4 3 3 1 2	33.32
	W2	6.19	11	477.21	123412451231 2	23.19	6	280.11	4222111533122	41.30	23.10	6	280.05	5 1 1 1 2 2 2 4 3 3 1 2	41.32

	W3	6.19	11	175.553	1234124512312	10.69	8	146.2433	311222451312	16.69	10.60	8	146.20	3 1 1 2 2 2 4 5 1 3 1 2	16.72
C(43221)	W1	5.80	11	162.77	123412513421	21.80	6	107.42	2211133544 21	34.00	21.80	6	107.40	2 2 1 1 1 3 3 5 4 4 2 1	34.02
	W2	5.80	11	476.82	123412513421	21.80	6	278.72	2211133544 21	41.54	21.80	6	278.66	2 2 1 1 1 3 3 5 4 4 2 1	41.56
	W3	5.80	11	174.41	123412513421	20.97	6	148.554	5221113344 21	14.82	21.97	6	107.59	2 2 5 1 1 1 3 3 4 4 2 1	38.31
D(52221)	W1	6.86	10	149.56	123411521341	20.69	6	106.31	4511112233 41	28.91	20.60	6	106.10	4 5 1 1 1 1 2 2 3 3 4 1	29.05
	W2	6.86	10	435.06	123411521341	20.69	6	277.61	4511112233 41	36.19	20.8	6	277.60	5 4 1 1 1 1 2 2 3 3 4 1	36.19
	W3	6.86	10	163.31	123411521341	14.86	7	144.49	4311112253 41	11.52	14.86	6	144.40	3 4 1 1 1 1 2 2 5 3 4 1	11.58
E(53211)	W1	5.88	11	162.85	123141251321	20.05	6	105.67	5221111433 21	35.11	20.05	6	105.67	5 2 2 1 1 1 1 4 3 3 2 1	35.11
	W2	5.88	11	476.90	123141251321	20.05	6	276.97	5221111433 21	41.92	20.05	6	276.90	5221111433 21	41.94
	W3	5.88	11	174.63	123141251321	8.72	8	140.32	3112241153 21	19.64	8.70	8	140.30	3 1 1 2 2 5 4 1 1 3 2 1	19.66
F(62211)	W1	6.30	10	149.00	121314511231	18.80	6	104.42	3511111422 31	29.91	18.8	6	104.40	3 4 1 1 1 1 1 5 2 2 3 1	29.93
	W2	6.30	10	434.50	121314511231	18.80	6	275.72	3411111522 31	36.54	18.80	6	275.66	3 4 1 1 1 1 1 5 2 2 3 1	36.56
	W3	6.30	10	161.61	122131451123 1	3.38	9	138.57	1214311152 31	14.24	8.13	8	138.50	2 1 1 3 4 5 1 1 1 2 3 1	14.30
G(63111)	W1	6.66	11	163.63	121312415121	11.83	7	111.72	3111225411 21	31.72	12.83	7	111.72	1 1 1 3 2 2 4 5 1 1 2 1	31.72
	W2	6.66	11	477.68	121312415121	31.49	6	288.41	5322411111 21	39.62	31.3	6	288.36	5 3 2 2 4 1 1 1 1 1 2 1	39.63
	W3	6.66	11	177.00	121312415121	11.84	7	135.41	3111225411 21	23.49	11.83	7	135.39	1 1 1 2 2 3 5 4 1 1 2 1	23.51
H(72111)	W1	6.77	9	135.20	121311451121	8.61	7	108.50	2111345111 21	19.74	8.61	7	108.50	2 1 1 1 1 5 4 3 1 1 2 1	19.74
	W2	6.77	9	392.15	121311451121	26.61	6	283.53	2435111111 21	27.6	26.61	6	283.47	2 4 5 3 1 1 1 1 1 1 2 1	27.71
	W3	6.77	9	148.76	121311451121	8.61	7	125.72	2111134511 21	15.48	8.60	7	125.70	2 1 1 1 3 5 4 1 1 1 2 1	15.50
I(81111)	W1	7.94	8	122.10	112131141511	13.11	5	84.46	3111111425 1	30.82	13.11	5	84.46	4 1 1 1 1 1 1 3 2 5 1 1	30.82
	W2	7.94	8	350.42	112131141511	13.06	5	227.16	3111114251 1	35.17	13.11	5	227.10	4 1 1 1 1 1 1 3 2 5 1 1	35.19
	W3	7.94	8	138.01	112131141511	13.11	5	110.69	3111111245 1	19.79	13.10	5	110.60	4 1 1 1 1 1 1 3 2 5 1 1	19.86

Sequence (A-1,B-2,C-3,D-4,E-5)

Table 4 Solution for Problem Set 3

Miltenburg Algorithm						Simulated Annealing Algorithm					Pso Algorithm				
Problem set	Scheme	u	s	Z	Sequence	u	s	Z	Sequence	RPD	u	s	z	Sequence	RPD
A(33333)	W1	11.9	14	211.77	1234512345 12345	32	9	160.43	44113322551 2345	24.24	32	9	160.40	1 1 4 4 2 2 3 3 5 5 1 2 3 4 5	24.26
	W2	11.9	14	611.34	1234512345 12345	32	9	417.29	11443322551 2345	31.74	32	9	417.20	1 1 4 4 2 2 3 3 5 5 1 2 3 4 5	31.76
	W3	11.9	14	235.77	1234512345 12345	18	11	210.97	15442233151 2345	10.51.	18	11	210.90	5 2 3 3 1 1 4 4 2 5 1 2 3 4 5	10.55
B(43332)	W1	8.35	14	208.13	1234512341 52341	30.88	9	159.31	22445331115 2341	23.45	28.22	9	156.65	5 2 2 4 4 3 3 1 1 1 5 2 3 4 1	24.73
	W2	8.35	14	607.69	1234512341 52341	28.22	9	413.51	52233441115 2341	31.95	28.20	9	413.50	5 2 2 4 4 3 3 1 1 1 5 2 3 4 1	31.96
	W3	8.35	14	224.84	1234512341 52341	18.35	10	197.76	15334422115 2341	12.04	14.22	11	199.63	3 1 5 4 4 2 2 3 1 1 5 2 3 4 1	11.21
C(53331)	W1	8.35	14	208.13	1234152134 12341	28.35	9	156.78	33111225441 2341	24.672	28.30	9	156.70	2 2 1 1 1 3 3 5 4 4 1 2 3 4 1	24.71
	W2	8.35	14	607.79	1234152134 12341	28.41	9	413.7	22111335441 2341	31.92	30.35	9	415.64	1 1 1 2 2 3 3 5 4 4 1 2 3 4 1	31.61
	W3	8.35	14	224.88	1234152134 12341	13.96	11	198.86	41133225141 2341	11.59	14.22	11	199.64	2 4 1 1 1 3 3 2 5 4 1 2 3 4 1	11.22
D(63321)	W1	7.55	14	207.33	1231415231 41231	14.75	10	157.45	41133225114 1231	24.71.07	14.75	10	157.40	4 1 1 2 2 3 3 5 1 1 4 1 2 3 1	24.08
	W2	7.55	14	606.89	1231415231 41231	33.95	9	419.24	45332211114 1231	30.92	34.22	9	419.51	4 3 3 2 2 5 1 1 1 1 4 1 2 3 1	30.88
	W3	7.55	14	207.37	1231415231 41231	5.06	10	157.88	11422335114 1231	23.86	14.75	10	186.96	4 1 1 2 2 3 3 5 1 1 4 1 2 3 1	9.84
E(73221)	W1	7.34	14	207.15	1231415121 34121	26.71	9	155.14	34111225113 4121	25.10	12.44	10	155.14	4 1 1 1 3 2 2 5 1	25.10

														1 3 4 1 2 1	
	W2	7.34	14	606.75	1231415121 34121	29.13	9	414.42	345221111134121	31.69	29.24	9	414.53	4 3 2 2 5 1 1 1 1 1 3 4 1 2 1	31.68
	W3	7.34	14	221.92	1231415121 34121	12.44	10	180.04	411132251134121	18.87	12.44	10	180.03	4 3 1 1 1 2 2 5 1 1 3 4 1 2 1	18.88
F (75111)	W1	8.62	14	208.40	1212314125 12121	22.88	9	151.31	222111134512121	27.39	22.88	9	151.31	2 2 2 1 1 1 1 3 4 5 1 2 1 2 1	27.39
	W2	8.62	14	608.00	1212314125 12121	22.91	9	408.20	222111134512121	32.86	23.28	9	408.57	3 1 1 1 1 2 2 2 4 5 1 2 1 2 1	32.80
	W3	8.62	14	225.68	1212314125 12121	12.89	10	181.39	112221143512121	19.625	13.28	10	182.56	1 1 2 2 2 4 1 1 3 5 1 2 1 2 1	19.10
G(93111)	W1	8.26	12	179.50	1211314121 51121	13.86	8	128.02	111223411151121	28.67	13.86	8	128.02	1 1 1 2 2 3 4 1 1 1 5 1 1 2 1	28.67
	W2	8.26	12	522.02	1211314121 51121	38.00	11	356.90	111212134151121	31.63	34.80	7	334.47	4 3 2 2 1 1 1 1 1 1 5 1 1 2 1	35.93
	W3	8.26	12	196.03	1211314121 51121	9.10	9	155.75	311411221151121	20.54	13.86	8	155.75	1 1 1 2 2 3 4 1 1 1 5 1 1 2 1	20.54
H(102111)	W1	8.35	10	151.05	1121311415 11211	12.35	7	112.24	211111143511211	25.71	12.35	7	112.24	2 1 1 1 1 1 1 4 3 5 1 1 2 1 1	25.71
	W2	8.35	10	436.455	1121311415 11211	13.35	7	312.02	211111134511211	28.51	12.35	7	312.02	2 1 1 1 1 1 1 3 4 15 1 1 2 1 1	28.51
	W3	8.35	10	167.79	1121311415 11211	17.11	6	136.97	211111143511211	18.36	12.35	7	136.95	2 1 1 1 1 1 1 3 4 5 1 1 2 1 1	18.38
I (11111)	W1	10.22	8	124.38	1112113141 11511	20.88	6	98.50	432111111111511	20.80	12.88	6	98.50	3 1 1 1 1 1 2 4 1 1 1 1 5 1 1	20.80
	W2	10.22	8	352.72	1112113141 11511	32.90	5	246.95	311111241111511	29.98	32.88	5	246.93	3 2 4 1 1 1 1 1 1 1 1 1 5 1 1	29.99

	W3	10.22	8	144.85	1112113141 11511	12.83	6	124.30	2111143211 51121	14.18	12.88	6	124.28	1 1 1 1 3 4 2 1 1 1 1 1 5 1 1	14.20

7. CPU time

We find the solutions for all the three problem sets and CPU time. In the computational problems, finding the CPU time is important because CPU time should be less. In all problem sets we get average CPU time is36.38.and solutions is1194 by PSO. In problem set 1 average CPU time is 17.75.and solutions is1091by PSO where as in SA.CPU time is22.05 and solution is2729, for problem set 2 average CPU time is33.54, solution is 1210 by PSO and in a SA. CPU time is56.59 solutions is4648. For problem set 3 average CPU time is57.87 and solutions is1283 by PSO where as in SA.CPU time is 122.84 and solutions is 8979.Among the all weightages, W2 have less CPU time of 13, no of solutions is 1208 for problem set 1 by PSO and in problem set 2,w3 have less CPU time of 29.55,no of solutions is 867 by PSO and W2 have less CPU time of 53.88, no of solutions is 1204 for problem set 1 by PSO

Table 5 CPU time and number of solutions for problem set 1

	W 1, simulated annealing		W1,PSO		W2, simulated annealing		W2,PSO		W3, simulated annealing		W3PSO	
Problem	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time
22222	2494	13.36	1234	35	2734	19.10	1285	14	2624	20.10	1259	22
32221	2784	14.36	1133	18	3054	20.10	996	10	2534	14.10	1021	20
33211	3216	43.30	1493	32	3293	15.20	1599	20	2524	12.52	648	9
42211	2604	35.00	1200	28	2744	10.16	1229	12	2544	33.10	907	14
43111	2454	34.40	506	18	2494	09.39	1335	13	2687	37.55	1365	16
52111	2764	35.10	947	26	2804	09.13	1362	16	2664	21.16	1467	18
61111	3034	34.20	365	15	2914	09.01	652	6	2364	23.10	914	11
Average	2764	29.96	982	24.51	2862	13.1	1208	13	2563	23.09	1083	15.74

Average number of solutions by SA; 2729, Average CPU TIME by SA ; 22.05

Average number of solutions by PSO; 1091, Average CPU TIME by PSO ;17.75

Table 6.Comparison results of Miltenburg, simulated annealing and PSO of problem set 1,2,3 for Heuristic1to 3

	Problem set 1					**Problem set 2**					**Problem set 3**				
Heuristic	**Miltenburg**	**SA**	**RPD**	**PSO**	**RPD**	**Miltenburg**	**SA**	**RPD**	**PSO**	**RPD**	**Miltenburg**	**SA**	**RPD**	**PSO**	**RPD**
Heuristic 1	106.89	80.35	24.80	80.35	24.80	122.10	84.46	30.82	84.46	30.82	124.38	98.50	20.80	98.50	20.80
Heuristic 2	306.78	198.30	35.30	198.24	35.38	350.42	227.16	35.17	227.10	35.19	352.72	246.95	29.98	246.93	29.99
Heuristic 3	120.37	98.37	18.27	98.35	18.29	138.01	110.69	19.79	110.60	19.86	144.85	124.30	14.18	124.28	14.20

Table 7CPU time and number of solutions for problem set 2

	W 1 simulated annealing		W1,PSO		W2, simulated annealing		W2,PSO		W3, simulated annealing		W3PSO	
Problem	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time
33222	4144	60	1598	37	4252	55	1796	44	4329	55	910	31
44211	5284	68	1261	31	5668	64	1283	31	4600	64	561	28
43211	4884	65	1565	39	5128	60	1989	45	4492	60	432	22
52221	5140	78	1911	47	5080	45	1024	28	4360	45	1600	48
53211	4672	56	1656	42	4300	40	1250	31	4564	40	981	31
62211	4540	61	1249	32	4200	53	1340	36	4564	53	718	26
63111	4756	70	1302	34	4650	54	1280	32	4768	54	801	28
72111	4468	61	1385	38	4780	55	954	29	4648	55	786	25
81111	4432	58	1228	36	4456	50	823	28	4372	50	1020	27
Average	4702	64.11	1461	37.33	4723	52.88	1304	33.77	4521	52.88	867	29.55

Average number of solutions by SA; 4648, Average CPU TIME by SA ;56.59

Average number of solutions by PSO; 1210 , Average CPU TIME by pso ;33.54

Table 8 CPU time and number of solutions for problem set 3

	W 1, simulated		W1,PSO		W2 simulated annealing		W2,PSO		W3 simulated annealing		W3PSO	
Problem	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time	No of solution	Cpu time
33333	8610	104	1872	72	8929	143	1543	65	8705	117	1937	78
43331	9159	115	1289	64	7872	117	567	38	8449	124	1423	57
53331	9229	126	1226	61	9574	150	928	42	8629	142	1625	68
63221	8615	118	1224	62	8884	123	1877	71	8569	130	1365	62
73221	8954	76	1317	65	9484	144	1478	62	8524	120	1789	73
75111	8674	87	1215	61	9574	152	925	45	8404	110	869	40
93111	8944	129	1422	78	9154	140	1280	62	8494	115	657	42
10 2111	9525	116	742	34	10264	85	854	37	8614	125	1987	78
11 1111 1	9600	117	872	38	9994	160	1390	63	9034	132	982	45
Avarage	9034	109.77	1242	59.44	9303	134.88	1204	53.88	8602	123.88	1403	60.33

Average number of solutions by SA; 8979, Average CPU TIME by SA ; 122.84

Average number of solutions by PSO; 1283, Average CPU TIME by pso ; 57.87

8. Conclusion

In this paper, the various heuristic methods based on particle swarm optimization have been discussed for solving production sequence in level schedule optimization problem. The Miltenburg algorithm has been used to find the sequence for scheduling. The sequence has been modified to obtain another sequence and to get utility and setup estimations by giving different weightages. These weights are then adjusted to obtain the desired value. According to different weightages, the utility and number of setups are obtained for heuristics 1to3. Theses elective heuristics procedures are applied for both Miltenburg algorithm and simulated annealing algorithm and particle swarm optimization. The proposed algorithm based on particle swarm optimization technique has been applied to find production sequences, when objective function values of setup and usage rates are desired as minimum. Three problem sets up to 15 numbers of products are solved and the results obtained for all the heuristics and results shows that particle swarm algorithm is better than Miltenberg algorithm and simulated annealing algorithm (2016) is to obtain balanced setup and utility and minimize the objective function value.

References

1. Monden.y,Toyota production system , Norcross GA; Institute of industrial engineers press.(1983).
2. KirkPatrick.S.andGelatt.C.D,Vecchi, Optimization by simulated annealing, Science 220 : 671-682(1983)
3. John Miltenburg , Level schedules for mixed model assembly lines in Just- in- time production system, Management science 35(2); 192-207((1989)
4. Senthilkumar. R and Ramkumar. A.S , Solving level scheduling in mixed model assembly line by simulated annealing.circuits and systems7 ; 907-931(2016).
5. Sumichrast.R.T.andRussell.R.S , Evaluating mixed model assembly line sequencing heuristic for just in time production system.Journal of operations management 9 ; 370-390(1990)
6. McMullen.P.R , JIT sequencing for mixed model assembling lines with setups using Tabu search.Production Planning&Control 9(5); 504 – 51(1998).
7. McMullen.P.R, Peter Taransewich and Gregory V Frazier , Using genetic algorithm to solve the multi productJit sequencing problem with setups. International Journal of Production Research 38(12); 2653 – 2670(2000)
8. McMullen.P.R , A Simulated annealing approach to mixed model sequencing with multiple objective on just in time. Institution of Industrial Engineering 32; 679-686(2000)
9. Afshinemansouri .S , A multi objective GA for mixed model assembly line sequencing on JIT assembly lines. European Journal of operational research167;696-716(2005)
10. Tanka NathDhamala, WieslawKubiak, A brief survey of Just- in- time sequencing of mixed model system. International Journal of Operations research 2;38- 47(2005).
11. Mohammed ZainiMatondang and Jambak, Soft computing in optimizing assembly line balancing. Journal of computer science 6; 141- 162(2010)
12. Tavakkoli-Moghaddam.R,.andGholipour-kanani.Y , Genetic and memetic algorithms for sequencing a new JIT mixed model assembly line.International journal ofmanagement science 44; 17-28(2008)
13. SenerAkpinar and MiracBayhan and AdilBaykasoglu (2013) Hybridizing ant colony optimization via genetic algorithm for mixed model assembly line balancing problem with sequence dependent setup time between tasks. Applied Soft Computing 13;574-589(2013).
14. SenerAkpinar and AdilBaykasoglu , Modeling and Solving mixed model assembly line balancing problem with setups part11;A Multiple colony hybrid bees algorithm. Journal of Manufacturing systems 33;445-461((2014)
15. Alirezaalfi ,PSO algorithm with dynamic inertia weight for on line parameter identification applied to Lorenz chaotic system. International journal of innovative computer information and control 8;1191-1203(2012).

16. Arti. S.gonge,D.S.Ingole , PSO algorithm for line balancing. International Journal of Innovative and emerging Research in engineering 2; 26-29,(2015).
17. Lixin tang and Xianpangwang , An improved pso algorithm for the hybrid flow shop scheduling to minimize total weighted completion time in process industry. IEEE transactions on control system technology 18(6);1303-1314,(2010)
18. Dian palupiRini, Sitimarriyamshamsuddin, PSO technique, system and challenges. International Journal of computer application14(1) ; 19-28,(2011).
19. Qiaoying dong, ling qin , sequencing mixed model assembly lines based on a modified pso multi objective algorithm.Proceeding of IEEE, International conference, automation and logistics Jinan China ; 18-22(2007).
20. Kennedy and Eberhart , A discrete binary version of pso.proceeding of IEEE conference on system man cybermetrics ;4104-4188 (1997).
21. Xiao fengxie,werjuwzhang , A Dissipative PSO.Congress on evolutionary computation,HawaliUSA ;1456-1461(2002).
22. Renato A Krohling , Gaussian swarm a novel pso.IEEEconference on cybermetric and intelligent systems Singapore;1-3,(2004)
23. Xiaolongxu, HanzhangRong, Marcello Travalt,MarkLiptrott, NikBessis , Cs-pso ; chaotic pso algorithm for solving combinatorial optimization problems.Soft computing;2383-2388(2016).
24. Lukasz Strak, RafelSkinderowicz,UrszulaBoryczka , Adjustability of a discrete particle swarm optimization for the dynamic travelling sales man problem. Soft computing ; 2738-2755,(2017) .

***** Corresponding author details:**

P.Dhiravidamani

Asst.Professor

KSR College of Engineering, Tiruchengode

Problem Set: 1

B	6	1	1	1	1
C	5	2	1	1	1`
D	4	2	2	1	1
E	4	3	1	1	1
F	3	3	2	1	1
G	3	2	2	2	2
H	2	2	2	2	2

Problem Set : 2

B	8	1	1	1	1
C	7	2	1	1	1`
D	6	3	1	1	1
E	6	2	2	1	1
F	5	3	2	1	1
G	5	2	2	2	1
H	4	3	2	2	1
I	4	4	2	1	1
J	3	3	2	2	2

Problem Set: 3

B	11	1	1	1	1
C	10	2	1	1	1
D	9	3	1	1	1
E	7	5	1	2	1
F	7	3	2	2	1
G	6	3	3	2	1
H	5	3	3	3	1
I	4	3	3	3	2
J	3	3	3	3	3