

Minimization of Total Weighted Delay in Flowshop Sequencing Problem Using Particle Swarm Optimization

S.V.R.Arun, S.Porselvi, DR.A.N.Balaji

ME (CAD/CAM), K.L.N college of Engineering, Tamilnadu, india

Assitant professor, Department of mechanical engg, K.L.N college of Engineering, Tamilnadu, india

professor, Department of mechanical engg, K.L.N college of Engineering, Tamilnadu, India

Abstract: Scheduling problems have played a vital role in recent years due to the fast growing of consumer demand for variety, quality, reduced product life cycles, competing with global competition and rapid development of new technologies. The Flow Shop Scheduling Problem (FSP) is one of the most popular scheduling models existing in practice, which is among the hardest combinatorial optimization problems. Particle Swarm Optimization (PSO) is a population based metaheuristic that takes advantage of individual and social cooperation in a Swarm. It has been applied to variety of optimization problems because of its simplicity and fast convergence. In this paper PSO is presented to solve the flowshop sequencing problem with an objective of minimizing the Total weighted delay of jobs. For this purpose a heuristic rule called the smallest position value (SPV) is used to enable the continuous PSO to be applied in sequencing problems.

Keywords: Particle Swarm Optimization, Shortest position value rule, Total weighted delay

I. INTRODUCTION

Metaheuristics are general frameworks commonly used for solving hard optimization problems. To examine a search space systematically, they employ intelligent and powerful heuristic mechanisms for both exploration and exploitation ([2] Mostafa Akhshabi, Javad Haddadnia). Amongst several metaheuristics using various search philosophies, ones providing dynamic balance between exploration and exploitation, usually deliver better performance. If exploration is the governing feature of a metaheuristic, it may perform a vague examination on the

search space and generally misses good solutions. On the contrary, if exploitation is dominant, the entire search space may not be explored and premature convergence to a local optimum is possible. Particle Swarm Optimization (PSO) is a metaheuristic introduced by Kennedy and Eberhart.

PSO is inspired of the behaviors of social models like bird flocking or fish schooling and is based on individual improvement and social cooperation [J. Jerald · P. Asokan [4]]. In PSO, a population of particles (swarm) flies over a multidimensional search space representing candidate solutions. During their search journey determined by their current tendency, personal experience and swarm's experience, particles are expected to explore the whole search space and to tend towards the optimal solution. PSO is initially proposed for continuous nonlinear optimization problems ([8] Ken Van den Eenden2 and Thomas Stutzle). Later it becomes quite popular and is applied to a wide range of optimization problems due to rapid swarm convergence and simplicity in its conceptual features and implementation. Although rapid convergence is a desirable advantage, it may result in a severe premature convergence problem, as well. The swarm may converge to a local solution and may not escape from it to explore the search space thoroughly. Moreover, PSO has a shortcoming in finding the best solution around a local neighborhood. Several modifications on the standard PSO algorithm are proposed to rectify above mentioned drawbacks. Most of these modifications concern only on either exploration or exploitation mechanism and merely correct the deficiencies partially. They locate good solutions for

some of the problem types but fail to produce competitive results for others. For instance, while the methods concentrated on exploration mechanism improve the solution quality for multimodal continuous function optimization problems, they typically generate inferior solutions for unimodal problems. In this paper, a PSO algorithm with improved exploration and exploitation mechanisms is proposed. In PSO, the general flow of the PSO algorithm is preserved. However, among the PSO iterations, the search strategy is temporarily modified only for the particle (the pioneering-particle) which achieves or enhances the best solution. For the pioneering-particles, Shortest position value (SPV) is employed. The solutions obtained by SPV do not affect swarm's experience in order to find promising search areas afterwards and not to hinder exploration. To improve exploration further and to prevent premature convergence, the pioneering-particles continue their search journey with random velocities and their previous experience. The primary advantage of PSO is its applicability to many different types of optimization problems with satisfactory performance. To support this claim, we experimented PSO on both continuous and discrete problems. Results for the continuous problems indicate that our approach improves the search capability of the standard PSO algorithm. PSO produces comparable solutions for both unimodal and multimodal functions while other similar PSO variants deliver poor solutions for either unimodal or multimodal functions. Moreover, our approach achieves the best known solutions for almost all instances of the Orienteering Problem (OP) which is a discrete NP-hard problem. Besides proposing a promising and valuable tool for solving different types of problems (continuous or discrete / unimodal or multimodal) satisfactorily, there are two auxiliary contributions of this paper; Firstly, we present the first successful PSO based algorithm for OP. IS-PSO which is derived from PSO, reproduces the best known solutions for all benchmark problems. Secondly, we propose that SPV with dynamic neighborhoods is a good alternative for a subsidiary local search ([5] M. Faith Tasgetiren et al).

1.1 PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a population-based method in which a swarm includes n individuals called particles. Each particle has a d -dimensional position vector representing a candidate solution and a d -dimensional velocity vector expressing the current tendency of the particle during its search journey. Initial swarm can be constructed randomly or by using some predetermined values. At each step, the velocity of each particle is re-evaluated based on the particle's inertia as well as the social interaction (swarm's experience) and personal experience of the particle. The experience of each particle is usually captured by its local best position ($pbest$). The experience of the swarm is captured by the global best position ($gbest$). In the course of several iterations, particles make use of this experience and are supposed to move towards the optimum position.

1.2 PSEUDO CODE

1. Procedure of PSO
 - Initialize parameters
 - Initialize population
2. For each particle
 - Evaluate
 - Update Local best
 - Update global best
 - Update velocity and position
 - While (not terminated)
 - End procedure

Pseudo-code of the standard PSO algorithm is illustrated above. Optimization is achieved in the course of several iterations of update-evaluate steps. During the update step at iteration t , the velocity and the position vector of each particle are calculated by using equations . In these equations are the velocity and the position values of the j^{th} dimension ($1 \leq j \leq d$) of the i^{th} particle ($1 \leq i \leq n$) at iteration . The stochastic behavior of PSO is achieved by random numbers $rand_1$ and $rand_2$ which are positive numbers generally uniformly distributed in $[0, 1]$.

After the update step, the fitness function value is calculated for each particle based on its position (the candidate solution represented by the particle.) The local best position, $pbest$ of each particle and the global best position, $gbest$ of the swarm are updated if the candidate solution is better than $gbest$ is considered as the best solution. PSO algorithm for FSP The $gbest$ model of Kennedy et al. is followed in this study. according to the $gbest$ model, each particle moves towards its best previous position and towards the best particle moves towards the best particle in the whole swarm. In the PSO algorithm for fsp, parameters were initialized and the initial population was generated randomly since evaluation of each particle in the swarm requires the determination of the permutations of the job for fsp, the smallest position value (SPV) rule applies to each particle corresponding permutation thus each particle will be evaluated by using the permutation to compute the fitness function value. the objective function for the fsp with total weighted delay. after evaluation, the PSO algorithm repeats the following steps iteratively. With its position, velocity, and fitness value, each particle updates its personal best (best value of each individual so far) if an improved fitness value was found. on the other hand the best particle in the whole swarm with its fitness and position was used to update the global best. Then the velocity of the particle is updated by using its previous velocity, the experiences of the personal best, and global best in order to determine the position of each particle. The permutation is determined through the SPV rule so that evaluation is again performed to compute the fitness of the particles in the swarm. this process is terminated with a pre-determined stopping criterion.

II.EXPERIMENTAL PROCEDURE

2.1.NUMERICAL ILLUSTRATION

Table 1

Job	M1 P.T	M2 P.T	M3 P.T
1	8	7	12
2	11	9	20
3	13	6	15
4	14	6	21
5	23	9	10

The processing time of the given problem is shown in table1

The maximum delay and the weight is Shown in table 2

Table 2

Job	D(i)	W
1	8	1
2	16	2
3	32	2
4	24	1
5	28	2

- M1=machine 1
- M2=machine 2
- M3=machine 3
- D(i) = maximum delay
- W(i)=weight
- P.T=processing time
- Scheduling type (a)
- 1 single machine
- M m identical parallel machines
- P Performance criteria
- D(i) total/mean (unweighted) delay time
- W(i) total/mean weighted delay time

$$x(i)=x_{min}+(x_{max}-x_{min}) * rand$$

$$v(i)=v_{min}+(v_{max}-v_{min}) * rand$$

SOLUTION REPRESENTATION:

one of the most important issues when designing the PSO algorithm lies on its solution representation.inorder to construct a direct relationship between the problem domain and the PSO particles for th PFSP, n numbers of dimension are presented for m number of jobs. Each dimension represents a typical job and the particle $X_j^1 = [X_{j1}, X_{j2}, X_{j3}, \dots, X_{jn}]$ corresponds to

the continuous position values for n number of jobs in the PFSP. The particle itself does not present a permutation, instead the SPV rule is used to determine the permutation implied by the position values x_{ij} of paticle X_i . The table 3 illustrates the solution representation of particle X_i of the PSO algorithm for the PFSP together with its corresponding velocity and permutation

	1	2	3	4	5
X_{ij}	3.71	1.94	1.78	3.20	2.03
V_{ij}	3.89	0.87	0.20	2.94	3.16
π_{ij}	3	2	4	5	1

Table 3

This representation is unique in terms of finding new solutions since position s of each particle are updated at each iteration in the PSO algorithm,thus resulting in different permutation at each iteration to be evaluated.

INITIAL POPULATION:

A population of a particle is constructed randomly for the PSO algorithm of the FSP.the continuous values of position are established randomly. The following formula is used to construct the intial continuous position values of particle uniformly.

$$x_{ij}=x_{min}+(x_{max}-x_{min}) * r1$$

where $x_{min}=0.0, x_{max}=4.0$ and r1 is a uniform random number between 0 and 1

initial velocities are generated by similar formula

$$v_{ij}=v_{min}+(v_{max}-v_{min}) * r2$$

$v_{min}=-4.0, v_{max}=4.0$ r2 is a random number between 0 and 1.the continous velocity are restricted to some range,namely

$$v_{ij}=[v_{min}, v_{max}]=[-4.0, 4.0]$$

where $v_{min}=-v_{max}$

population size is twice the number of dimensiond .the objective is to minimize the total weight delay

2.2 COMPUTATIONAL PROCEDURE

The complete computational procedure for PSO algorithm for the FSP can be summarized as follows.

Step1:Initialization

- Set $t=0, NP=$ twice the number of dimensions.
- Generate NP particles randomly as explained before
- Generate initial velocities for each particle randomly
- Apply SPV rule to find the permutation
- Evaluate each particle in the swarm using objective function
- Find the best fitness value among the whole swarm

Step2:Update iterations counter

- $t=t+1$

Step3 :Update inertia weight

$$w^t=w^{t-1} * \beta$$

where β is a decrement factor

Step4:Update velocity

$$V_{ij}=w^{t-1} v_{ij}^{t-1} + c_1 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2 (g_j^{t-1} - x_{ij}^{t-1})$$

Minimization of Total weighted delay in flowshop sequencing problem using PSO

Step 5: Update position

After calculation for total weighted delay the change in processing time are shown in table 4

Job	M1	M 2	M3
2	11	20	40
1	19	27	52
3	32	38	67
5	55	64	77
4	69	75	98

Table 4

The total processing time of each job and the delay calculated are shown in table 5.

P.T	D(i)	Delay	w
40	16	24	2
52	8	44	1
67	32	35	2
77	28	49	2
98	24	74	1

Table 5

- Delay = $P.T < D(i) = 0; P.T > D(i) = P.T - D(i)$
- The total weighted delay = $Delay1 * W1 + Delay2 * W2 + Delay3 * W3 + Delay4 * W4 + Delay5 * W5$

The total weighted delay for the sequence 2-1-3-5-4=334

III EXPERIMENTAL RESULT AND DISCUSSION

The proposed PSO algorithm for the FSP was coded in MATLAB R2009B and run on a intelcore i3 laptop pc with 1GB memory Initially the velocity and population is calculated and the result got after executing the problem in matlab by using PSO algorithm It is found the minimization of Total weighted delay is less the global best delay.

The personal best delay got is in sequence 3-2-4-5-1=250

IV. CONCLUSION

To the best of my knowledge, this is the first reported application of the particle swarm optimization algorithm to solve the FSP with total weighted delay of jobs in the literature. Here SPV rule is used to enable the continuous PSO algorithm to be applied to all classes of sequencing problems and the result got is better than the global best solution

REFERENCES

- [1] Alebachew D. Yimer, Kudret Demirli (Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines) International Journal of Approximate Reasoning 50 (2009) 117–137
- [2] Rasaratnam Logendran, Paula deSzoek, Faith Barnard (sequence-dependent group scheduling problems in flexible flow shops) Int. J. Production Economics 102 (2006) 66–86
- [3] Ruben Ruiz a,*, Jose Antonio Vazquez-Rodriguez The hybrid flow shop scheduling problem
- [4] J. Jerald · P. Asokan · G. Prabaharan · R. Saravanan Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm
- [5] M. Faith Tasgetiren, Yun-chia Liang (A particle swarm optimization algorithm for makespan and flowtime minimization in the permutation flowshops sequencing problem) european journal of operational research 177(2007)1930-1947
- [6] Bo Liua,b,c,*, Ling Wang, Ying Liud, Bin Qiane, Yi-Hui Jin (An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes)
- [7] Junqi Zhang Kun Liu Ying Tan* Xingui He (Allocation of Local and Global Search Capabilities of Particle in Canonical PSO)
- [8] A. Montes de Oca1, Ken Van den Enden2 and Thomas Stutzle (Incremental Particle Swarm-Guided Local Search for Continuous Optimization.)
- [9] Suk-Hun Yoon, Jose A. Ventura, (Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling), Computers & Operations Research ,29, (2002) 1301-1315
- [10] Buthainah F Al-Dulaimi and Hamza A. Ali, (A Novel Genetic Algorithm Approach for Solving Flow Shop Problem), IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.9, September 2008.
- [11] Mostafa Akhshabi, Javad Haddadnia, Mohammad Akhshabi, (Solving flow shop scheduling problem using a parallel genetic Algorithm), Procedia Technology 1 (2012) 351 – 355.
- [12] Pempera Jaroslaw, Smutnicki Czeslaw, Zelazny Dominik, (Optimizing bicriteria flow shop scheduling problem by simulated annealing algorithm), Procedia Computer Science 18 (2013) 936 – 945.
- [13] F. Jolai, H. Asefi, M. Rabiee, P. Ramezani, (Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem) Scientia Iranica E (2013) 20 (3), 861–87