A novel discrete PSO algorithm for solving job shop scheduling problem to minimize makespan

1Author: Mubeen Shaikh, a research Scholar and 2Author: Dr. Dhananjay Yadav, Professor

At Department of Mechanical Engineering at SSSUTMS, Sehore-MP

Abstract
A discrete variant of the PSO algorithm is suggested in this paper to reduce the makespan of a job-shop. To create active timetables, an unique schedule builder was used. The discrete PSO is put to the test with well-known benchmark problems from the literature. The proposed algorithms' solution is compared to the best known solution in the literature, as well as the hybrid particle swarm method and variable neighbourhood search PSO algorithm. The methodology used in this study was found to be effective in delivering high-quality solutions for a variety of benchmark job-shop scheduling challenges.

1. Introduction

The most difficult combinatorial optimization issue is job-shop scheduling [1]. Scheduling entails allocating jobs to available machines on the shop floor. A typical schedule contains information about the jobs that will be loaded onto the machines and when they will be loaded. As the number of jobs and machines grows, so does the difficulty of fixing the problem. Jobs are scheduled on the machines in accordance with a certain goal, which can be considered independently or in combination. Researchers and practical engineers created schedules based on some of the most significant objectives, such as decreasing make-span, minimising flow-time, and minimising tardiness.

The make-span criterion is used to build schedules in this research. of all the jobs in all the machines in the shop-floor. Throughput is improved when the make-span is reduced. For a 'n'job'm' machine problem, the make-span (C max) is calculated using the equation 1 expression [2]. Each work 'ji' on the shop floor must go through 'O' operations on'm' machines before being completed. Jobs must be scheduled in accordance with the process plans, taking into account prioritisation and resource restrictions. The algorithm receives as input the processing time 'tik' of job 'j' in machine 'm'. The job's start time, 'tik', is calculated 3. To whom should all correspondence be sent? Following priority and resource capacity limitations, with the goal of reducing the make-span criteria.

It is also believed that jobs would not be revisited and that each work will be processed just once in the machine.

Various optimization strategies have been used to solve the shop scheduling problem over the years. Exact methods, branch and bound algorithms, heuristics based on dispatching rules, Tabu search, Simulated annealing algorithm, Genetic algorithm, Local search techniques, Particle swarm algorithms, Differential evolution algorithms, and hybrid techniques are some of the techniques available. The researchers conducted a comprehensive survey on job-shop scheduling. McKay et al. [3], Holthaus and Rajendran [4], Yamada et al. [5], Baewicz et al. [6,7], Jain and Meeran [2], Jones et al. [8], and Wang and Zou [9] conducted an extensive survey on job-shop optimization processes of mathematical formulations, and future directions. Simulation based approaches for minimizing makespan in a job-shop has been studied by Thenarasu et. al. [10].

Kennedy and Eberhart [11] introduced particle swarm optimization, which is based on the intelligent behaviour of swarms and has been applied to a number of issues in science and engineering including continuous and discrete variables. For travelling salesman situations, Clerc [12] implemented a discrete version of PSO. Tasgetiren et al. [13] developed a continuous version of PSO for permutation flow shop scheduling problems. Tasgetiren et al. [14] and Xia and Wu [15] attempted to handle job shop scheduling difficulties with the PSO algorithm with the goal of lowering the makespan. By translating position values to its permutation of operations, Tasgetiren et al. [13,14] presented a lowest position value rule (SPV) to enable the continuous particle-swarm optimization algorithm to tackle permutation flowshop scheduling problems and job shop scheduling problems. A new hybrid particle swarm optimization approach suggested by Xia and Wu [15] combines a continuous particle swarm algorithm with a simulated annealing algorithm. For the flow-shop scheduling problem, Ramesh kumar et al. [16, 17, 18] suggested a PSO methodology for data clustering challenges, Karthi et al. [19] created discrete and continuous version PSO algorithms. For the challenge of supply chain network optimization, Kadadevaramath et al.[20] suggested an intelligent PSO model. PSO was implemented by Sun. For better performance, Meng et al. [23] merged GA and SA in PSO. The proposed novel discrete version of the PSO algorithm is used to tackle job-shop scheduling problems in this paper. By minimising the makespan criterion, optimal / near-optimal schedules were constructed. The suggested algorithm's solution quality is assessed by solving a benchmark issue and comparing it to the best-known solution in the literature.

2. Particle representation and proposed schedule builder

In this paper, the suggested PSO method for the job shop scheduling problem is implemented using the operation-based representation provided by Gen et al. [24]. All operations for a given job are represented as discrete values, which are subsequently inferred according to the sequence of occurrence. A particle in PSO is represented by 'nm' integers for a 'nm' job-shop problem. For a three-job (n), three-machine (m) problem, the representation would be as illustrated in figure 1, where 1 denotes job 1, 2 denotes job 2, and 3 denotes job 3. In the PSO algorithm, a particle will be represented by...
nine integers.

| Particle representation | 3   | 2   | 2   | 1   | 2   | 3   | 3 |
|--------------------------|-----|-----|-----|-----|-----|-----|---|
| First Operation | job3 | First Operation | job2 | Second Operation | job2 | First Operation | job1 | Second Operation | job1 | Third Operation | forjob2 | Second Operation | forjob | Third Operation | forjob3 |

Figure 1. Particle representation

2.1. Schedule builder

The schedule builder is a component of the evaluation technique that should be selected based on the optimization performance measure. The majority of significant work shop scheduling performance indicators are regular measurements, implying that best solutions are always semi-active [25]. The particle representation approach is used to provide an optimal solution for a performance objective such as minimising the makespan using semi-active scheduling methods. Computational experiments revealed that using a more powerful schedule builder, specifically an active scheduler, improves the minimal makespan job shop scheduling problem. To improve the quality of the solution, an active schedule builder performs a kind of local-search. Forcing is a tactic that was first utilised by Nakano and Yamada [26]. The strategy of left-shifting is utilised in the schedule builder to generate the schedule. To design the solution for the given sequence of operations, we explored a semi active schedule builder as well as a novel active schedule builder in this work.

2.2. Active schedule builder (proposed)

Left shifts and the forcing operation are performed by an active scheduler. The technique for creating the active schedules is based on a time-incrementing scheduling generating scheme. Figure 2 shows the algorithmic description of the schedule creation scheme that was utilised to construct the active schedules.

Sett=0,RT$_{w}=$0(Release time of machine(m))

Phase I:
For every machine(m): do
If f (RT) = Release the job on that machine(m);
Then
Activate the next operation of that job; Put it in the corresponding machine queue;
Phase II:
For every machine(m): do

If $e$ (RT) and machine is free and queue exist
Then
Choose one from queue and proceed
r: Load the operation on machine m;
Update (RT = t + processing time)
Goto
Phase until all jobs are scheduled.

Figure 2. Schedule generations scheme

3. Benchmark problems

Bench-mark problems are used test the performance of proposed algorithms. These bench-mark problems were proposed by many researchers. These bench-mark problems were solved by many researchers by various approaches and techniques and reported their results in the literature. The bench-mark problems proposed by researchers are of wide range of sizes and also their difficulty level of solving also varies. There are many problems in the bench-mark where the optimal solutions are not found due to its combinatorial nature. These benchmark problems are formulated by various authors. To illustrate the effectiveness and performance of the proposed algorithm, we have considered 40 problem instances (LA01 ~ LA40) of eight different sizes due to Lawrence [27], 5 problems (ABZ5 ~ ABZ9) of 2 different sizes due to Adams et al. [28], three problem instances called FT06, FT10 and FT20 due to Fisher and Thompson [29] and 10 problems (ORB1 ~ ORB10) used by Applegate and Cook [30]. We have solved all the 58-benchmark problems using the proposed PSO algorithm and the results were compared with the PSO algorithms proposed by Tasgetiren [13,14] and Xia and Wu [15].

4. Proposed discrete version PSO for job-shop scheduling

4.1. Structure of PSO algorithm for Job-shop scheduling problem

The main goal is to establish the job-shop problem's minimum makespan by determining the sequence of operations done on the jobs in the respective machines. Make-span is the total time it takes for all operations on all jobs in all machines to be completed. A series made up of m integers. The number of jobs is represented as 'n,' and the number of machines is represented as 'm.' Each integer in the series represents the order in which the jobs in the machines were completed. A particle in PSO is a series.

In PSO, the solution is built by taking into account the particle's present position ($C$, $n$), the best position obtained by the particle at a specific point in time throughout the evolution ($B$, $n$), and the overall best position among all the particles ($G$). $C$, $n$, $B$, $n$, and $G$ are updated after a new sequence is created by computing the sequence's makespan. The letter 's' stands for the sequence, and 'n' stands for the number of dimensions. The proposed form of PSO is not the same as the general PSO that is utilised to solve problems involving continuous function optimization. The structure of the proposed discrete PSO is given as follows:

\[\text{Step 1: } \text{Generate the sequences randomly. No. of sequences(SwarmSize))say} S \text{; i.e., (C, n), (B, n), (G) are } 1, 2, \ldots, S.\]

\[\text{Step 2: Generate the feasible sequence proposed by Genetial[24].}\]

\[\text{Step 3: For all } \{C, n\}, \text{ find the make-span of the sequence (objective function value) } \]

\[\text{i.e., } f_i(C, n) \text{. }\]

\[\{\text{Active schedule builder and Semi active schedule builder are used for construction the schedule}\}

\[\text{Step 4: Initialize (Bs, n) and (Gn) \}

\[\text{Step 5: While (No of iterations/ No of solutions generated is not reached) do for each particle }\]

\[\text{Build a new sequence(} \text{particle) ;}\]

\[\text{Step 6: Update (Cs, n), (Bs, n) and (Gn) \}

\[\text{Step 6: Report the best sequence/solution found.}\]

4.2. Building a new sequence

A new sequence of the particle ‘s’ is built using the current position of the particle is [$C$, $n$] best position reached by the particle say at an iteration ‘t’ is [$B$, $n$] and over-all best position among all the particles is [$G$]. [$B$, $n$] and [$G$] are identified based on the make-span value arrived by evaluating the sequences(particles). A random number in the range U[0,1] is generated and compared with the set weights $w_c$, $w_h$ and $w_g$. The set weights $w_c$, $w_h$ and $w_g$ corresponds to the sequences (particle) [$C$, $n$], [$B$, $n$] and [$G$] for building the new sequence. The weights are generated in such a way that $w_c + w_h + w_g = 1$.

To demonstrate the sequence building procedure, It is assumed that...
that there is a sequence of four operations for a 3-job-3-machine problem (i.e., particles) in the swarm as given below:

\[ \{C_n\} = \{1, 2, 1, 3, 2, 1, 3\} \]

\[ \{B_{n}\} = \{2, 1, 2, 1, 3, 2, 1, 3\} \]

\[ \{G_{n}\} = \{3, 2, 1, 3, 2, 1, 3\} \]

Let \( w_c, w_b, w_d \) and \( w_a \) are the relative importance, which is similar to the coefficient \( c_1 \) and \( c_2 \) used in the context of the PSO equation for calculating velocity as shown in equation 1. Velocity of the particle is denoted by \( v \), \( t \) is the iteration counter, \( c_1 \) and \( c_2 \) are the relative significance of the ‘social’ and ‘cognitive’ coefficients which is used to find the velocity update the new position. \( X(t) \) is the current position of the particle and \( X_{best} \) and ‘G-best’ are particle’s best and global best positions. Two uniformly distributed random numbers \( r_1 \) and \( r_2 \) are in the range \([0, 1]\).

New position is updated after calculating the velocity. New position of the particle is given in the equation 2.

For the demonstration of constructing the new solution, Let us consider \( w_c=0.2, w_b=0.3 \) and \( w_d=0.5 \). \( w_c, w_b \) and \( w_d \) are sampled in the range \([0, 1]\) in such a way that \( w_c+w_b+w_d=1 \). The building of a new sequence starts with a null set \( \{C\} \). Corresponding to the sequence of operations, nine random numbers are generated in the range \([0, 1]\) for each element \( \{C\} \). Let the random numbers be \( 0.98, 0.45, 0.35, 0.24, 0.64, 0.04, 0.95, 0.17 \) and \( 0.10 \). Corresponding the random numbers, by following the construction procedure, the new sequence

\[ \{C_n\} = \{1, 2, 1, 3, 2, 1, 3\} \]

\[ \{B_{n}\} = \{2, 1, 2, 1, 3, 2, 1, 3\} \]

\[ \{G_{n}\} = \{3, 2, 1, 3, 2, 1, 3\} \]

5. Performance analysis of discrete PSO algorithm

The bench-mark problems are solved by the proposed discrete version of the PSO algorithm. Swarm size considered in this study is 20 and max number of iterations allowed to report the results using the proposed algorithm is 50. The make-span reported by the algorithm is indicated in the Table 1. The make-span of the proposed algorithm is compared with best known solution available in the literature and two PSO algorithms proposed by Tasgetiren et al. [13] and Xia and Wu [14]. Tasgetiren et al. [13] hybridized their PSO algorithm with local search with a variable neighbourhood search method. Xia and Wu [14] hybridized their PSO with simulated annealing technique. The relative performance increase (RPI) over the best known solution is also presented in the Table 1.

The relative percent increase in makespan over the best-known solution is calculated using the equation 3 for the proposed algorithm.

\[ \text{Relative percent increase in makespan} = \frac{\text{BKS}-\text{Make span}_{\text{proposed}}}{\text{BKS}} \times 100 \]

It is observed from the results that for 37 bench-mark problems PSO

| S.No Problem (n,m) | BKS H-PSO PSO-VNS D-PSO PSO % |
|-------------------|---------------------------------|
| 01 abz5 10x10    | 1234 1234 1234 1238 0.32        |
| 02 abz6 10x10    | 943 943 943 945 0.21            |
| 03 abz7 20x15    | 656 666 659 688 4.88            |
| 04 abz8 20x15    | 646 681 674 708 9.60            |

was able to achieve the best known solution available in the literature. This shows that the proposed discrete version PSO is able to produce good quality solutions for the job-shop scheduling problems. Swarm size considered in this study is 20. Max number of iterations allowed to report the results using the proposed algorithm is 50.

Table1. Performance of algorithms- Problems provided by Adamsetal.[27]

Conclusions

In this paper, a novel discrete version of particle swarm algorithm (PSO) is presented. The algorithm is tested using well-known benchmark problem available in the literature. The solution obtained from the proposed algorithm is compared with best-known solution published in the literature. The performance of the algorithm is found to be good and able to achieve the best-known solution to 37 problems among 38 problems considered in this study. The success of this proposed algorithm is due to the novel solution construction procedure employed in this study. The weights assigned to the current particle’s best and global best particles is equivalent to the social and cognitive coefficients used in the conventional PSO algorithms used for solving the continuous function optimization problems. Further, the algorithm proposed in this study will be modified to suit the multi-objective optimization of the job-shops.

References

[1] Garey MR, and Johnson D 1979 Computers and Intractability: A Guide to Theory of NP-Completeness. Freeman, San Francisco.

[2] Jain AS, and Meerran S 1999 A state of the art review of job shop scheduling techniques European Journal of Operational Research. 113: 390-434.

[3] McKay KN, Safayeni FR and Buzacott JA 1998 Job shop scheduling theory: What is relevant? Interfaces. 18(4) 84-90.

[4] Holthaus JR, and Rajendran C 1997 New dispatching rules for scheduling in a job shop—An experimental study The International Journal of Advanced Manufacturing Technology. 13(2) 148-153.

[5] Yamada Tand Nakano R 1997 Job shop scheduling IEE control Engineering series. 134-134.

[6] Blazewicz J, Domschke W, and Pesch, E., 1996 The job shop scheduling problem: Conventional and new solution techniques Europeanjournal of operational research. 93(11): 33.

[7] Blazewicz J, Eckert K, H.Pesch E, Schmidt G, and Weglarz J 2007 Handbook on scheduling: from theory to applications. Springer Science & Business Media.

[8] Jones A, Rabelo L, C, and Sharawi AT 1999 Survey of job shop scheduling techniques Wiley encyclopedia of electrical and electronics engineering.

[9] Wang S F, and Zou Y R 2003 Techniques for the job shop scheduling problem: a survey Systems Engineering Theory & Practice. 23(1) 49-55.

[10] Thenarasu M, Ramesh Kumar K, and Marimuthu P (in press) Simulation Modeling and Development of Analytic Hierarchy Process (AHP) based Priority
Dispatching Rule (DPR) for a Dynamic PressShop International Journal of Industrial and Systems Engineering.

[11] Kennedy J, and Eberhart RC 1995 Particles warm optimization Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ, USA, 1942-1948.

[12] Clerc M 2004 Discrete particle swarm optimization, illustrated by the Traveling Salesman Problem New Optimization Techniques in Engineering. Heidelberg, Germany, Springer 219-239.

[13] Tasgetiren F M, Liang Y-C, Sevkli M, and Gencyilmaz G 2007 Particle Swarm optimization for makespan and Total flowtime minimization in permutation flowshop sequencing problem European Journal of Operational Research, 177 1930–1947.

[14] Tasgetiren M F, Sevkli M, Liang Y-C, and Yenisey M M 2006. A particle swarm optimization and differential evolution algorithms for jobshop scheduling problem International Journal of Operations Research, 3(2) 120-135.

[15] Xia W J, and Wu Z M 2006 A hybrid particle swarm optimization approach for the job-shop scheduling problem The International Journal of Advanced Manufacturing Technology, 29(3) 360-366.

[16] Ramesh Kumar, Suresh R, and Mohana Sundaram K 2005 Discrete particle swarm optimization (DPSO) algorithm for permutation flowshop scheduling to minimize makespan. LNCS - Advances in Natural Computation, 361 257-281.

[17] Rameshkumar K, Rajendran C, and Mohana Sundaram K M 2011 Discrete particle swarm optimization algorithms for minimizing the completion-time variance of jobs in flowshops. International Journal of Industrial and Systems Engineering, 7(3) pp. 317-340.

[18] Rameshkumar K, Rajendran C, and Mohana Sundaram K M 2012. A novel particle swarm optimization algorithm for continuous function optimization. International Journal of Operational Research, 13(1) 1-21.

[19] Karthi R, Rajendran C, and Ramesh kumar K 2011 Neighbourhood Search Assisted Particle Swarm Optimization (NPSO) Algorithm for Partitional Data Clustering Problems Advances in Computing and Communications, 552-561.

[20] Kadadevaramath R S, Chen JC, Shankar B L, and Ramesh kumar K 2012 Application of particles warm intelligence algorithms in supply chain network architecture optimization Expert Systems with Applications, 39(11) 10160-10176.

[21] Sun Y, and Xiong H 2012. Job-shop scheduling problem based on particle swarm optimization algorithm Sensors & Transducers, 161 116.

[22] Gao H, Kwong S, Fan B, and Wang R 2014 A hybrid particle-swarm tabu search algorithm for solving job shop scheduling problems IEEE Transactions on Industrial Informatics, 10(4) 2044-2054.

[23] Meng Q, Zhang L, and Fan Y 2016 A Hybrid Particle Swarm Optimization Algorithm for Solving Job Shop Scheduling Problems Asian Simulation Conference, 71-78 Springer Singapore.

[24] Gen M, Tsujimura Y, and Kubota E 1994 Solving job-shop scheduling problems by genetic algorithm Systems, Man, and Cybernetics, 2157-1582.

[25] French H 1982 Sequencing and Scheduling: An introduction to the mathematics of the job shop. Ellis Horwood John Wiley & Sons New York.

[26] Nakano R, and Yamada T 1991 Conventional genetic algorithm for job shop problems ICGA, 91474-479.

[27] Lawrence S 1984 Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques GSA. Carnegie Mellon University Pittsburgh PA.

[28] Adams J, Balas E, and Zawack D 1988 The Shifting bottleneck procedure for job shop scheduling Management Science, 34, 391-401.

[29] Fisher H, and Thompson GL 1960 Probabilistic learning combinations of local jobshop scheduling rules MathJ