Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

Parallel Particle Swarm Optimization for Job Shop Scheduling

Sathis Kumar K[1], Janani T[2], Raihana A[3], Guru Vimal Kumar M[4]
[1,2,4] Bannari Amman Institute of Technology, Erode – 638 401, India
[3] Sri Krishna College of Engineering and Technology, Coimbatore – 641008, India
sathiskumark@bitsathy.ac.in

ABSTRACT

The job-shop scheduling problem is one of the existing combinatorial optimization problems and it is also an NP-hard problem. A schedule of operations has to be found in order to reduce the maximum completion time is one of the main intention in performing job-shop scheduling. The completed time of completing all the jobs as specified in the given schedule for the given n * m; n – jobs and m – machines. Particle Swarm Optimization (PSO) is a prototype for planning metaheuristic calculations for combinatorial improvement issues. This decides the sequencing of creation in a Job shop system, which comprises of a change of assignments to be distributed on the machines with the end goal that limit the production time (or makespan). This methodology gives great arrangements in a short execution time, permitting the examination of huge situations in qualified occasions. Grid environment simultaneously applies the assets of numerous PCs in an organization to handle a solitary issue. The proposed work adopts the problem in grid environment to reduce the makespan.

Keywords
Job Shop Scheduling, Particle Swarm Optimization, Makespan, Metaheuristic.

Introduction

Job shop booking is an optimization problem in Computer Science in which ideal positions are allocated to assets at specific occasions. In its overall structure, it is NP-complete implying that there is presumably no proficient methodology for precisely discovering shortest best schedule for self-assertive occurrences of the issue. Job shop Scheduling is generally done utilizing heuristic algorithms that exploit exceptional properties of every instance.

Scheduling is the process of deciding how to commit resources between varieties of possible tasks. The fundamental Performance models for scheduling are machine usage, fabricating lead times, stock expenses, meeting due dates, consumer loyalty, and nature of items which all ward depends in how effectively the submitted jobs is to be scheduled and executed in the given system. The Job Shops are delegated as single-stage, single machine, parallel machine, multi-stage stream shop and multi-stage work shop as proposed by Lawler et al. 1993. Delegating a shop with single-stage designs requires just a single activity to be carried out as specified in the given machines. In a multi-stage stream work shop, a few assignments were carried out for each job and there subsist a typical course for each work. In job shops with multi-stage, a choice of choosing elective asset sets and courses for the given positions are specified. Consequently the job shop permits adaptability in creating an assortment of components.

Juang (2004) proposed a crossover of two algorithms namely genetic and particle swam optimization for repetitive system plan. Sivanandam et al. (2008) proposed a plan in utilizing the Genetic Algorithm (GA) to address the problem of Job Shop Scheduling in a productive way. Lin et al. (2009) planned a different algorithm based on hybrid swarm intelligence which comprises three techniques namely optimization using particle swarm, procedures of simulated annealing and multi-type singular improvement to take care of the Job shop scheduling issue. Bozejko et al.

Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

(2009) built up a Job Shop Scheduling Problem (JSSP) by employing Parallel Simulated Annealing Technique. Ge et al. (2007) introduced another cross breed calculation for Job Shop Scheduling Problem dependent on swarm of particles enhancement and simulated annealing. Tamilarasi and Anantha kumar (2010) proposed hybrid methodology of genetic algorithm with simulated annealing to tackle job shop booking issues.

This work focuses on utilizing Particle Swarm Optimization to resolve of the Job Scheduling problem. Particle Swarm Optimization is a population based on the strategy of stochastic optimization motivated by friendly conduct of bird flocking or fish schooling. Particle Swarm Optimization is utilized for approaches that can be utilized across a broad scope of applications and for explicit purpose zeroed on a particular necessity.

Job Shop Scheduling Problem

Let m be the machines and n be the jobs that constitute JSSP. In order to finish its work, each and every job should pass through the available m machines. There are m operations in each job. Each and every operation employs one of the machines in order to perform the work in a predetermined amount of time. If one task is completed on a machine, it cannot be stopped until the job is completed. A job's sequence of operations should be predetermined and can vary from job to job. Oji is referred as a single operation which is the output of one single job executed on the machine. Oji is specified as ith machine process jth job, $1 \leq j \leq n, 1 \leq i' \leq m$. This was given by Garey et al. 1976 and Lawler et al. 1993. During the time interval, each machine can only execute one operation. The aim of JSSP is to uncover a suitable function permutation for all the available jobs which minimizes makespan represented as Cmax, which is the cumulative completion time of the closing operation in a nm operation agenda.

Considering a JSSP with n×m, we can model with a set of machines m which is symbolized as $M = \{1,2,\ldots,m\}$ in order to process the set of operations given as n×m which is symbolized as $O = \{0,1,2,\ldots,(n\times m)-1\}$

The following are the notations:

n	: list of all jobs
m	: list of the operations for a single job
O_i	: Operation completion period i
t_i	: processing time on a given machine for operation i
C_{max}	: Makespan
O_{ji}	: A single job processed on a single machine is measured as a single operation.
T_{ij}	: processing start time
Tf_{ij}	: processing completion time

The following hypothesis is formed:

- A machine can process at most single job in a given time
- A job can be processed by at most single machine at a given time
- The order of machines m which the job visits during its execution has to be fully specified
- Time taken to process is to be recognized
- All the jobs have to be processed only once in each of the m machines.

Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

Particle Swarm Optimization

PSO has a lot in common with evolutionary computing methods like the available Genetic Algorithms. The system is initiated by initializing random population and then updates generations to look for optima. In contrast to GA, however, PSO lacks development operators such as crossover and mutation. Particles, which are possible solutions in PSO, obey the existing optimum particles across the problem space. Every particle in the search space is considered as a position in the given N-dimensional space that changes the characteristic of "flying" based on its own flying experience as well as that of other particles.

The following are the guiding principles:

Evaluate : the current present position
Compare : with the best value of previous and neighborhood
Imitate : oneself and others

Personal Best is a concept used to describe how each particle keeps track of the coordinates components in the solution space that are correlated with the available best solution (fitness) that that particle has accomplished so far which is represented as Pbest. Gbest – global best of the particle is the best value acquired so far by any particle in the vicinity of that particle. The basic idea behind PSO is to use a random weighted acceleration to accelerate every particle toward its personal and global best positions at every time point.

The x-vector, p-vector, and v-vector are the three vectors that make up a particle (individual). The x-vector represents the present position (location) of the particle in the given search space, the p-vector represents the location of particle's best solution achieved so far, v-vector represents the gradient (direction) in which the particle will move if it is not disturbed. A particle also has two fitness values: x-fitness, which records the x-fitness, vector's and p-fitness, which records the p-fitness vector's. The swarm keeps the global best.

Algorithm

I. Create the array of particle population in the problem space with dimension d by assigning the locations and velocities with random values.
II. Calculate the optimal optimization fitness function in d variables for each particle.
III. Compare the particle's fitness assessment to the particle's pbest. pbest is set to the current value if the current value is better than pbest. The pbest location is assigned with the current location in the given d-dimensional space.
IV. Contrast the fitness evaluation to the overall previous best in the population. gbest is reset to the array index and value of the current particle when the current value is better than gbest.
V. Adjust the particle's velocity and position according to equations (1) and (2), respectively:
    1. $v_{id} = w * v_{id} + c1 * rand( ) * (p_{id} – x_{id}) + c2 * Rand( ) * (p_{gd} – x_{id})$ (1)
    2. $x_{id} = x_{id} + v_{id}$ (2)
VI. Repeat steps II – VI until a requirement is reached, normally a high level of fitness or a maximum number of iterations (generations)

Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

Proposed System

The jobs and its operations are represented as shown below.

Table 1. Representation of jobs and operations

Job:1	Operation:1	Operation:2	Operation:3
Job:2	Operation:4	Operation:5	Operation:6
Job:3	Operation:7	Operation:8	Operation:9

The operations on each machine are scheduled before the execution. This can be represented as

Table 2. Representation of operations on each machine

Machine:1	Machine:2	Machine:3
Operation:1	Operation:2	Operation:3
Operation:4	Operation:5	Operation:6
Operation:7	Operation:8	Operation:9

Operational time for each operation is 1, 2, 1, 1, 2, 2, 1, 2, and 1. The feasible schedule for the above is Operation 7, Operation 8, Operation 4, Operation 5, Operation 1, Operation 2, Operation 6, Operation 9 and Operation 3. The operations that are executed on the machine based on the above given schedule yields a value of 8 as the makespan. This makes the above given schedule to be feasible. The gantt chart for the above schedule is given in the figure below.

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

Figure 1 Gantt Chat

Particles are corresponding to the solutions of job shop scheduling. A permutation sequence is an arranged rundown of tasks that addresses an arbitrarily created gathering of particles (arrangements). These particles are encoded by generation of random numbers and performing a ranking among them to form an operational sequence which a particle has to solve the problem. Every particle is assigned with a arbitrary random number (π_k) by positioning the genes (real numbers) in rising order. Every random number π_k is assigned to carry out (π_k mod No. of Jobs) +1 operation in order to obtain the equivalent operation order of a chromosome. Representation of particles can be illustrated as follows (Anantha kumar and Tamilarasi, 2010).

Particle

210	91	73	64	1	2	100	200	63

Integer Series (π_k)

Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

9	6	5	4	1	2	7	3	3

(π_k mod no_of_jobs) + 1

1	1	3	2	2	3	2	3	1

Operational sequence

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

Figure 4.2 Particle Representation

According to the novel PSO velocity principle, every particle travel at a speed defined by distance among its preceding location and gbest (pbest) solution. The particle velocity serves two purposes: it keeps the particle going in the direction of the solutions of gbest and pbest, and it maintains inertia to stop particles as of being stuck in value of local optima.

Random method is employed to generate the initial velocity of particles. The velocity of the particles in the PSO is influenced by its inertial weight. Random updating is made on velocities at the beginning of the iteration. Then the pervious velocity is used for the remaining iterations.

Conclusion

Making the Particle swarm streamlining equal for addressing position shop planning will limit the makespan. Grid Computing uses the appropriated heterogeneous assets to help confounded issues in computation. This is a sort of parallel and distributed framework that empowers the sharing, determination, and total of geologically conveyed self-ruling and heterogeneous assets progressively at runtime relying upon their accessibility, capacity, execution, cost, and clients' nature of-administration prerequisites. This can be accomplished through grid computing which simultaneously applies the assets of numerous PCs in an organization to handle a solitary issue. Parallel execution in grid is possible with the Message Passing Interface. Message Passing Interface is an API specification that permits computers to communicate with each other. The Message Passing Interface is intended to give fundamental virtual geography, synchronization, and correspondence usefulness between a bunch of cycles (that have been planned to hubs/workers/PC occurrences) in a language-free way, with language-explicit grammar, in addition to a couple of language-explicit highlights.

References

[1] T. Anantha kumar and A. Tamilarasi, “An Enhanced Genetic Algorithm with Simulated Annealing for Job-Shop Scheduling”, International Journal of Engineering, Science and Technology, Vol. 2, No. 1, pp. 144-15, 2010.

[2] P. Brandimarte and D. Lei, “A Pareto Archive Particle Swarm Optimization for Multi-Objective Job Shop Scheduling,” Computers & Industrial Engineering, pp. 960–971, 2008.

Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 4989 - 4994
Received 11 May 2021; Accepted 15 May 2021.

[3] F. Pezzella, G. Morganti and G. Ciaschetti, "A Genetic Algorithm for the Flexible Job-Shop Scheduling Problem". International journal of Computational Operations, vol 35, pp.3202-3212, 2008.

[4] D.Y. Sha and C.-Y Hsu, "A Hybrid Particle Swarm Optimization for Job Shop Scheduling Problem," Computers & Industrial Engineering, pp.791–808, 2006.

[5] Y.C. Liang , H.W Ge, and X.C Guo, "A Particle Swarm Optimization-Based Algorithm for Job-Shop Scheduling Problems," International Journal of Computational Methods, pp. 419–430, 2005.

[6] W. Xia and Z. Wu, "An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problem". IEEE Transactions on Computation India, vol 48, pp.409-425, 2005.

[7] K. Mesghouni, S. Hammadi and P. Borne, "Evolutionary Algorithms for Job-Shop Scheduling", International Journal of Applied Mathematics and Computer Science, Vol. 14, No. 1, pp. 91-103, 2004.

[8] X. Wang and J. Li, "Hybrid Particle Swarm Optimization with Simulated Annealing", Proceedings of Third International Conference on Machine Learning and Cybernetics, Vol.4, pp. 2402-2405, 2004.

[9] R.C. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", Proceedings of Congress on Evolutionary Computation, pp.105-110, 1998.

[10] E.L.Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B Shmoys, "Sequencing and Scheduling: Algorithms and Complexity", Handbooks in operations research and management science, Logistics of Production and Inventory, Vol. 4, pp. 445-552, 1993.

[11] M. Singer, "A Shifting Bottleneck Heuristic for Minimizing the Total Weighted Tardiness in a Job Shop", Naval Research Logistics, Vol. 46, No. 1, pp. 1–17, 1999.

[12] Y.C. Liang, M. Sevkli, and G. Gencyilmaz, "Particle Swarm Optimization Algorithm for Single Machine Total Weighted Tardiness Problem", IEEE Proceedings of Evolutionary Computation, pp. 1412-1419, 2004.