International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
***Article Received**: 15 June 2022 **Revised**: 26 August 2022 **Accepted**: 30 August 2022 **Publication**: 30 September 2022*

Numerical Simulation and Assessment of Meta Heuristic Optimization Based Multi Objective Dynamic Job Shop Scheduling System

Bhumika Singh Rajput[a], Anshul Gangele[b], Satish Kumar Alaria[c]
a Student,, Department of Computer Science &Engineering, Medi Caps University, Indore, India
b Professor, Department of Mechanical Engineering, Adina Institute of Science and Technology, Sagar , India,
c Assistant Professor, Department of Computer Science & Engineering, AIET, Jaipur

Abstract— In today's world of manufacturing, cost reduction becomes one of the most important issues. A successful business needs to reduce its cost to be competitive. The programming of the machine is playing an important role in production planning and control as a tool to help manufacturers reduce their costs maximizing the use of their resources. The programming problem is not only limited to the programming of the machine, but also covers many other areas such such as computer and information technology and communication. From the definition, programming is an art that involves allocating, planning the allocation and utilization of resources to achieve a goal. The aim of the program is complete tasks in a reasonable amount of time. This reasonableness is a performance measure of how well the resources are allocated to tasks. Time or time-dependent functions are always it used as performance measures. The objectives of this research are to develop Intelligent Search Heuristic algorithms (ISHA) for equal and variable size sub lot for m machine flow shop problems, to Implement Particle Swarm Optimization algorithm (PSO) in matlab, to develop PSO based Optimization program for efficient job shop scheduling problem. The work also address solution to observe and verify results of PSO based Job Shop Scheduling with help of graft chart.

Index Terms— Job Shop, PSO, Optimization, Dynamic Work Span, Scheduling

I. INTRODUCTION

Cost cutting has emerged as one of the most crucial concerns in the modern manufacturing industry. A profitable company must lower its costs to remain competitive. The machine's programming is a crucial component of production planning and control since it enables manufacturers to cut costs while making the best possible use of their resources. The programming issue is not just restricted to machine programming; it also affects a wide range of other fields, including computer, information, and communication technologies. According to the definition, programming is an art that entails allocating, organizing, and utilizing resources in order to accomplish a goal. The program's goal is to finish tasks within a fair period of time. A performance indicator of how well resources are allocated to tasks is reasonableness. It always uses time or time-dependent functions as performance indicators.

The majority of research on genetic programming algorithm optimization uses an evolutionary and genetic "living ofthe strongest "rule. According to Jain and Meeran [11], this strategy has been effective for evolutionary computation using JSP, but it hasn't been demonstrated to be better than other machine intelligence methods.

A form of computational intelligence known as particle swarm optimization has demonstrated promise for resolving additional optimization issues (PSO). The flight patterns of a flock of birds served as inspiration for the PSO Algorithm. It is regarded by Song and Gu [19] as one of the quicker convergent computational algorithms for intelligence and a prime candidate as an algorithm for multimodal functions. Since the PSO optimization approach has been used to solve other scheduling issues, such as the flow line problem (FSP) and the versatile work shop problem (FJSP), which are related issues that will be discussed later, see [12], [21], [22], and [25]. One benefit of the PSO algorithm is its speed. It is praised as a very quick convergence algorithm. The problem is only suitable to try to make this relation using the PSO algorithm JSP given the performance of strategy optimization in other areas and the ongoing need for improved heuristic methods to solve Workshop problems. Since creating a feasible schedule can

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
***Article Received**: 15 June 2022 **Revised**: 26 August 2022 **Accepted**: 30 August 2022 **Publication**: 30 September 2022*

be very computationally expensive. Therefore, any method that can prove its value would be more beneficial.

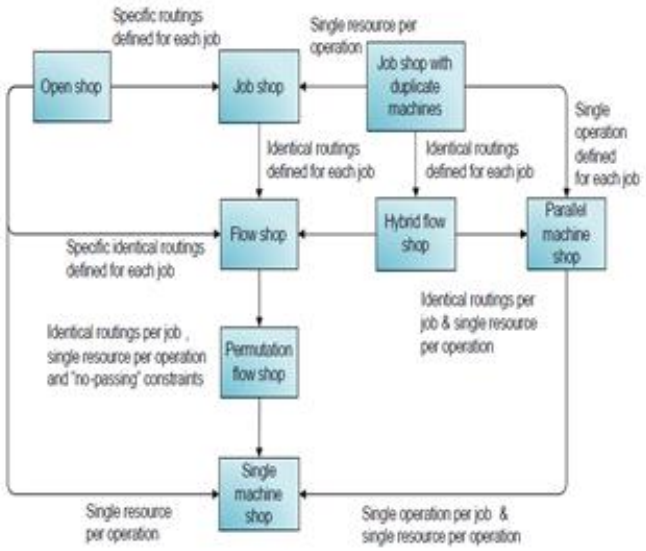


Fig. 1. Job-Shop Problem

Of course, each new method should be at least briefly explored to determine whether it may eventually result in more intriguing research and, in turn, better optimization techniques. One strategy has been used to make this optimization achievable because of the continuous nature of the PSO and a JSP permutation.

II. JOB SHOP SCHEDULING

A (n m) Shop job problem has a set of jobs, n, each with an order of operations, m, that is equal to the number of machines or resources specified in the problem. As a result, a work Ji is a predetermined sequence of operations. oi (Oi, 1, Oi, 2, ..., Oi, m). Oh, no. Oij, each service has a running period, also known as working hours.

The following are the rules for traditional JSP:

• Each job can only be handled by one computer at a time.
• Each machine can only do one job at a time.
• With the use of m computers, all jobs must be processed in the same order.
• At time zero, all jobs and machines are operational.
• Following the completion of the previous operation, a new operation begins.
• Each job's processing and setup times are known.
• Each work is self-contained, and there are no restrictions on priority.
• Machine breakdowns and interruptions are not taken into account when processing.
• Computers may not be idle during the execution of a task, but they may be idle during the transition between jobs.
• Sublot sizes should be understood ahead of time and remain consistent.
• It is not permitted to interrupt the processing of a sub lot.
• Setup time for sub lots belonging to different batches can be needed.

It's easier to imagine what a semi-active programme isn't to comprehend this idea.

Table 3.1: Simple JSP Example

Job	Machine Sequence (Processing Time)		
Job 1	1 (3)	2 (5)	3 (2)
Job 2	1 (5)	3 (1)	2 (4)
Job 3	2 (4)	1 (2)	3 (1)

Figure 3.2 shows a Gantt chart depicting one non-semi-active schedule for this simple (3 3) JSP.

Fig 3.2. Gantt chart of schedule for (3× 3) JSP

Keep in mind that the second and third procedures on machine 1 were not scheduled for the first available time window. The red boxes indicate this unnecessary extended wait. The times are obviously not ideal. Simply make the second calendar third actions on machine 1 semi-active by moving from left to the earliest permissible time. In Figure 3.2, this new semi-active schedule is displayed. It's important to note that none of the aforementioned activities may be begun without changing the order in which the machine performs its actions. [27]

A direct depiction of a schedule is any design that explicitly states the time when each operation will start running on the machines. The scheduling space is directly optimized, potentially via left-shifting. Another example would be to immediately create optimization values based on creating makespan and (N m) set start timings for a (N m) Workshop issues. A meta-heuristic algorithm may be used to learn or develop a set of start timings for this kind of connection based merely on a goal value as the

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
***Article Received**: 15 June 2022 **Revised**: 26 August 2022 **Accepted**: 30 August 2022 **Publication**: 30 September 2022*

makespan. It is evident that when direct rendering is employed, no workable schedules can be generated since the problem's prior constraints are ignored. If start timings for each operation were utilized as a direct representation, the final solution may resemble this if tij is the operation's start time.

$$t_{ij} = \begin{bmatrix} 8.0 & 0.0 & 5.0 \\ 3.0 & 1.0 & 9.0 \\ 5.0 & 1.0 & 0.0 \end{bmatrix}$$

One common method for avoiding precedent constraints in JSP programming is to use an indirect representation and then use a scheduling algorithm to convert the indirect representation into a feasible schedule. Permutation with repetition - In this representation, a scheduling algorithm decodes a permutation of job numbers (n m) into a feasible schedule. Each digit represents a job, which is repeated m times. The operation of order k to be processed by job j is represented by the repeated work order k j. This semi-scheduling algorithm uses active default schedules, such as asset class schedules and no delay. Figure 3.2 shows an example of a permutation with repetition and a scheduler.

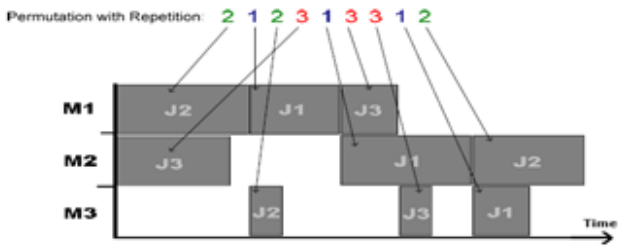


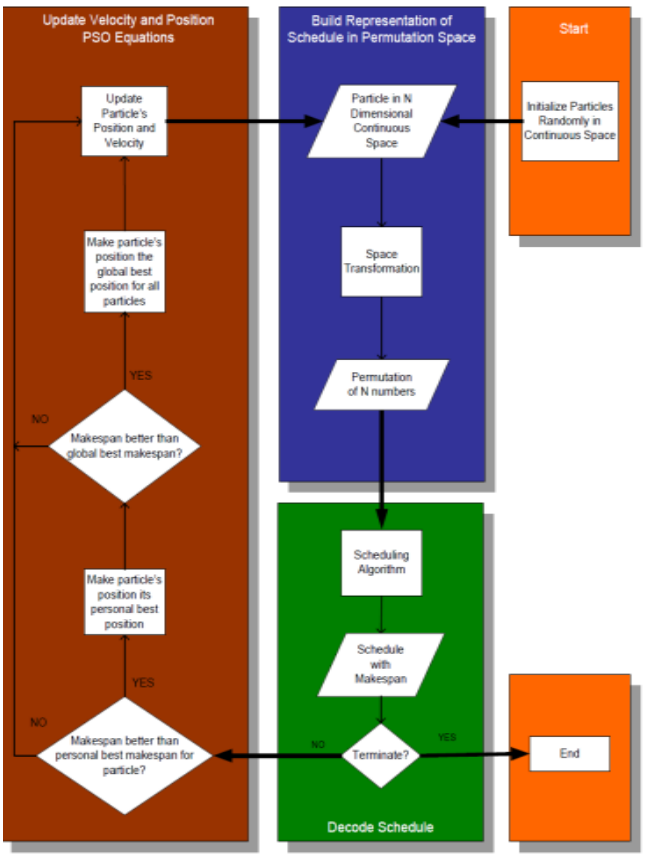
Figure 3.12: Permutation with repetition

It's simple to see how an indirect representation combined with a corresponding scheduling algorithm can circumvent the limitations of search decision space (the (n m) permutation of numbers in the example above). Because, before scheduling, a scheduling algorithm will consult the preceding limitations. The repeat permutation algorithm is a straightforward and efficient programming method.

III. PSO BASED JOB=SHOP

The With a number of permutations, the JSP/PSO method was devised to describe an indirect solution to a JSP. There are two obstacles in the way of PSO's immediate use in JSP. The governing equations PSO cannot be used to directly optimize a permutation of the integers that would indicate a solution to the JSP (in the context of a scheduling algorithm), as illustrated in Figure 4.1. There must be a technique to continuously permute space. The choice of what kind of scheduling algorithm to utilize, or how to turn these permutations of integers one at a time, is the second roadblock. Figure 5.1 on the following page of the complete clarifying process illustrates these two independent stages.

The permutation decoding process is shown in green space, while the transformation process is shown in blue space. The continuum of particles in the space permutation were altered using the greatest value prior rule, or GVP [15]. (shown in blue in Figure 5.1). The GVP rule includes giving each dimension or component of a particle continuous space its complete index. The sequence in which these distributions are formed determines their permutation. The n permutation values will be high if there is an n-dimensional continuous PSO space. The permutation of a particle is determined by assigning a value of 1 to the dimension with the largest magnitude. Then, a value of two is assigned to the particle's dimension with the next-largest magnitude. For every aspect of the issue, this procedure is repeated. The foregoing method converts a particle's three-dimensional space into a permutation, as shown in Figure 5.2. [22]



International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
***Article Received**: 15 June 2022 **Revised**: 26 August 2022 **Accepted**: 30 August 2022 **Publication**: 30 September 2022*

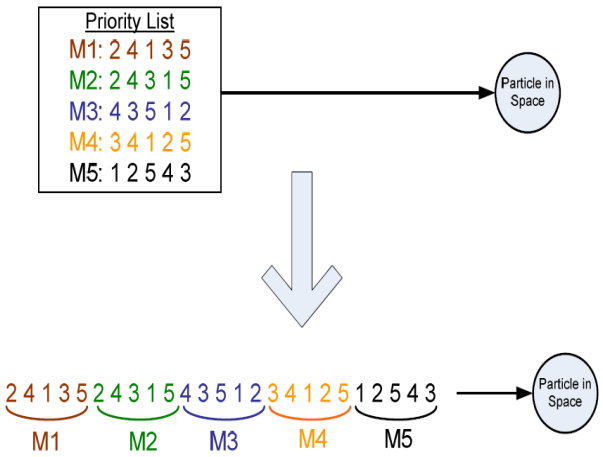


Figure 5.3: List of priorities for a particle (a)

The arrows technically should point from the particle to the permutation as permutations are created from the locations of particles in space, however the goal of this example is to show how space is divided. Another illustration of the best-of-all principle is shown in Figure 5.6.

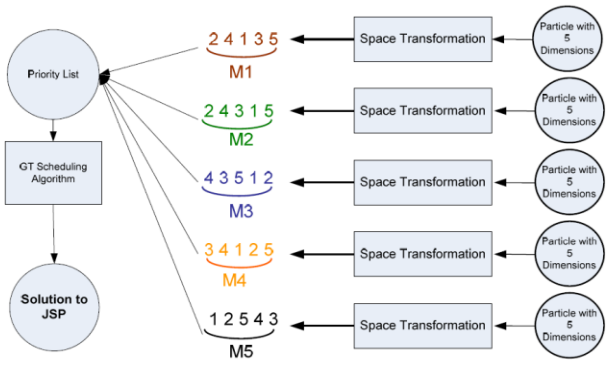


Figure 5.5: JSP / PSO Block Diagram

Using this technique, the search space in the JSP / PSO has effectively become m number of swarms, one for each machine. Figure 5.7 shows this clearly other "related" particles in other swarms. The space division principle does not initially appear to be effective as an optimization tool, and it does not combine well with other meta-heuristics, but the PSO may be used in this case because it enables the particles to store information about their best individual and overall positions in space. There are m better separated places, one for each of the m distinct swarms, when a collection of connected particles considers a better makespan. Each swarm will record the position reached by the particle in the swarm as the best in the world if this occurs and makespan is the best to date.

IV. SIMULATION & RESULT

In the MATLAB programming language, the JSP/PSO was written and evaluated. This algorithm is divided into two parts: scheduling and optimization, also known as PSO. Many of the scheduling and optimization (PSO) aspects have been discussed in general. However, in the following two pages, more information on how this software worked in these two areas is revealed.

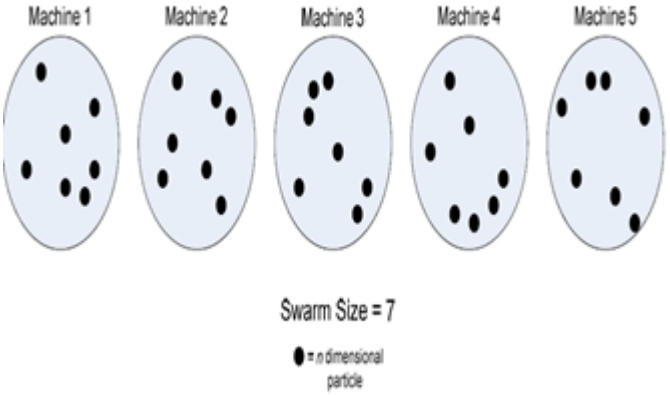


Figure 5.6: JSP / PSO Swarm for a n x 5 JSP

Each JSP machine has its own swarm, which gives particles a goal-oriented ideal world to work toward. Which makes sense because the priority list for each machine must differ. There are particles, however the swarm size is just 7. In the JSP/PSO method, each swarm particle is "linked" to another swarm particle and is maintained as "connected" during optimization. The pairing will soon be established and will be chosen at random. The identical particles from each swarm will always be combined to build a solution because there is, of course, no genuine real link (which does not share information). In other words, the "matching" particles have the same "fitness" throughout each cycle. Figure 5.8 in particular shows it graphically.

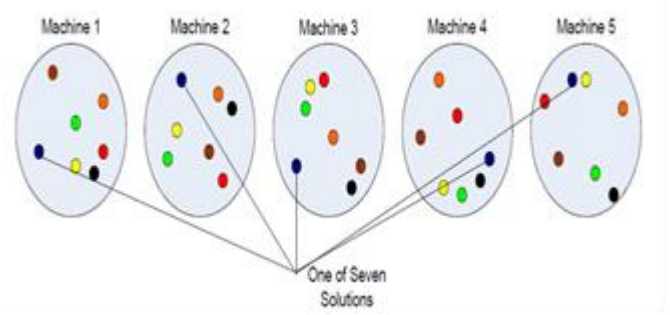


Figure 5.7: JSP / PSO Bonding

This is a vital idea to grasp. Essentially, these particles travel independently of other swarms, but their "fitness" or

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
***Article Received**: 15 June 2022 **Revised**: 26 August 2022 **Accepted**: 30 August 2022 **Publication**: 30 September 2022*

better location in space is inevitably influenced by the position of

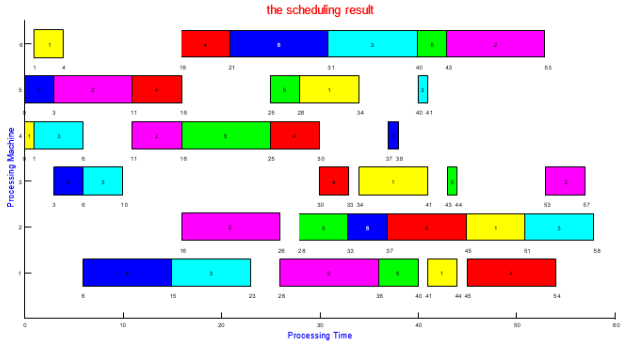


Fig 6.1 Sample Output of 6 X6 PSO based JSP

Case-1 - JSP/PSO Results(6 X 6 Problem)

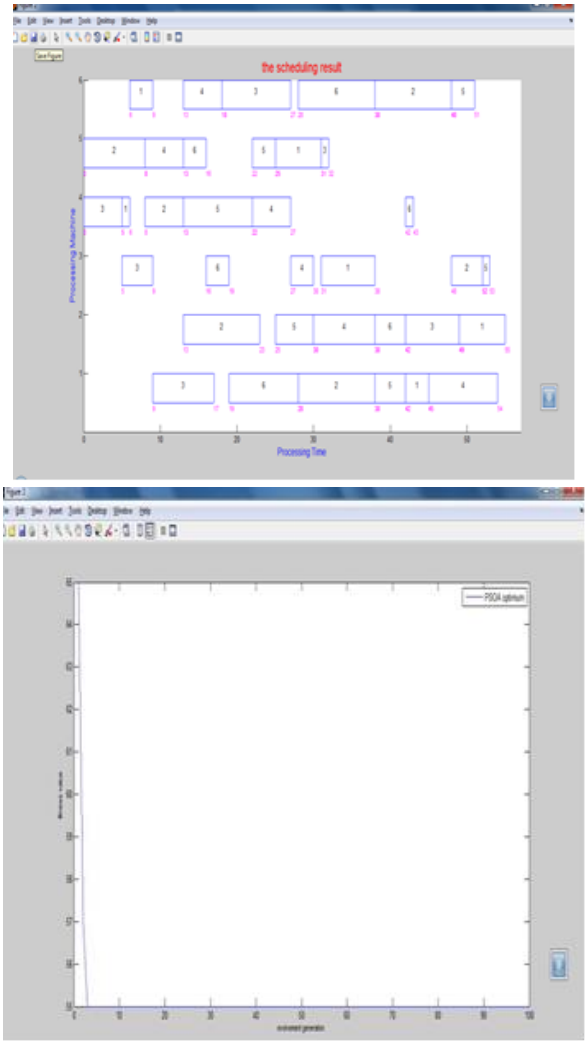
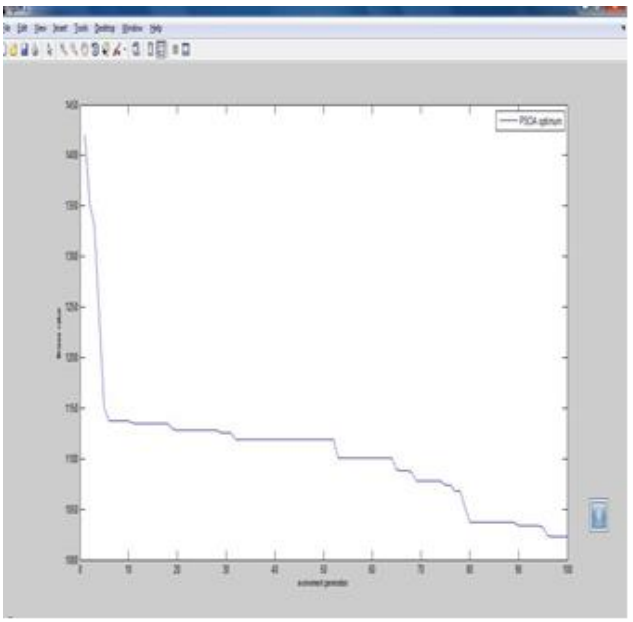


Fig 6.2 PSO based JSP for 6 X 6 Problem (b) Minimization of Objective (Make span)

We discovered that using the PSO Optimization process, the job's make time was drastically reduced.With the help of the following table, we have represented the reduction of make span (Objective function). The table below shows a comparison of make span reduction as we perform iterations in the PSO optimization process.

Case-2 - JSP/PSO Results(10 X 10 Problem)



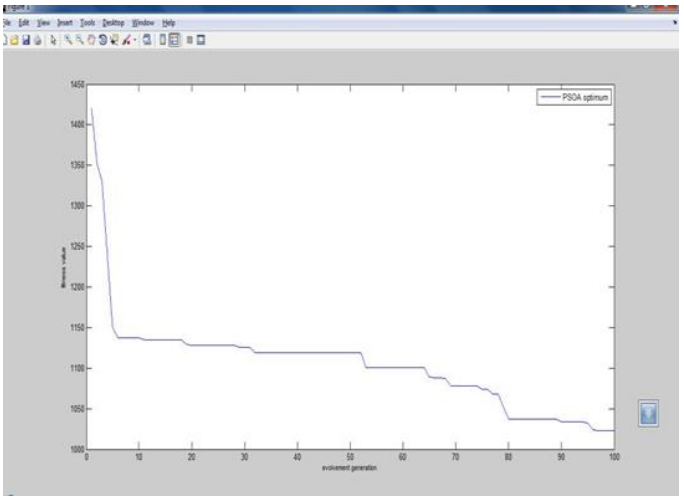
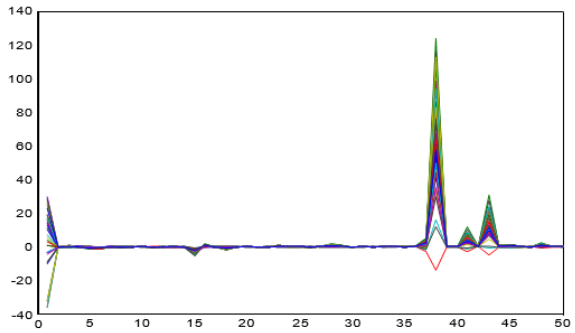


Fig 6.3 PSO based JSP for 10 X 10 Problem (b) Minimization of Objective (Make span)

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
Article Received: 15 June 2022 Revised: 26 August 2022 Accepted: 30 August 2022 Publication: 30 September 2022



Plot of Particle Position in PSO

Case-3 - JSP/PSO Results(20 X 5 Problem)

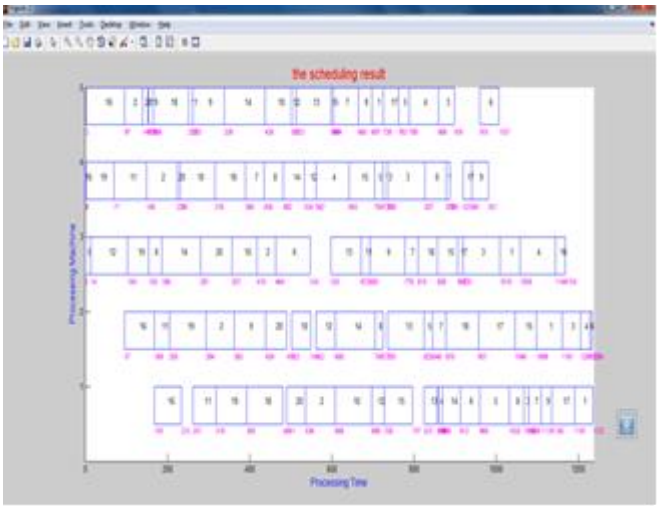


Fig 6.3 PSO based JSP for 10 X 10 Problem (b) Minimization of Objective (Make span)

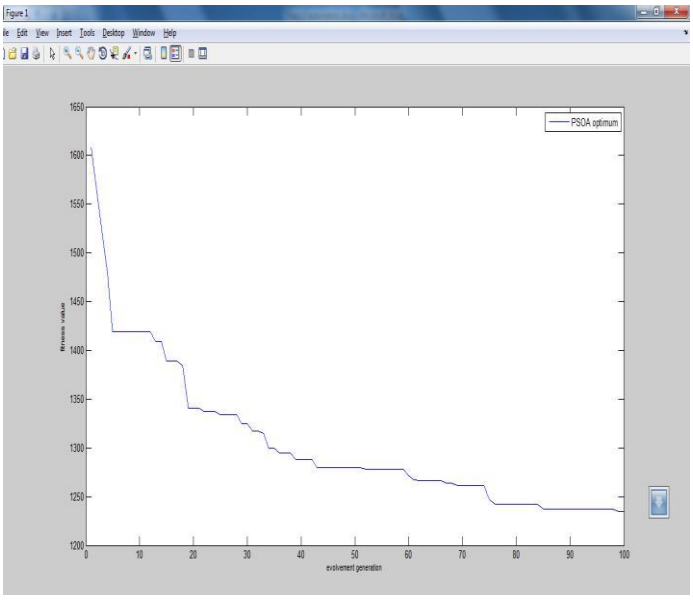
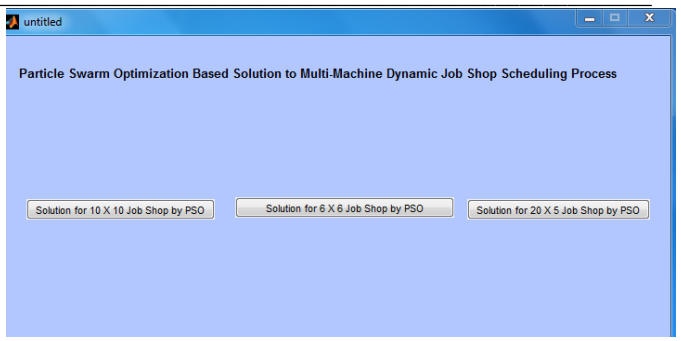


Fig 6.3 PSO based JSP for 10 X 10 Problem (b) Minimization of Objective (Make span)



V. CONCLUSION

The JSP/PSO Algorithm that has been presented can be used to tackle Job Shop Problems. This is significant because of how the optimization process operated, partitioning the search space across the participating machines. This space division approach shows that the existence of such a "collective effort" as of this writing need not be known to individual swarms or communities cooperating towards a shared aim. Although this may have been acknowledged in the meta-heuristic optimization community, to the best of my knowledge, it has never been used to solve the Work Shop Problem. A more important finding is that an algorithm may have a greater likelihood of success if it employs the same search space split by machines but promotes information exchange amongst various swarms or populations.

REFERENCES

[1]. Bierwirth, C. A Generalized Permutation Approach to Job Shop Scheduling with Genetic Algorithms. OR Spektrum, **17**:87-92. 1995.

[2]. Chang, Y. L, Sueyoshi, T. and Sullivan, R. S. Ranking Dispatching Rules by Data Envelopment Analysis in a Job- Shop Environment, *IIE Transactions*, **28**(8): 631-642. 1996.

[3]. Davis, L., Job Shop Scheduling with Genetic Algorithms, *Proceedings of the 1st International Conference on Genetic Algorithms*, Pittsburgh, PA, 136-140. 1985.

[4]. Dimopoulos, C., Zalzala, A., Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions, and Comparisons, *Transactions on Evolutionary Computation*, **4**(2): 93-113. 2000.

[5]. Eberhard, R.C., and Kennedy, J., A New Optimizer Using Particle Swarm Theory, *Proceedings. of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 39-43. 1995.

[6]. Fenton, P., Walsh, P., Improving the Performance of the Repeating Permutation Representation Using

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2105
Article Received: 15 June 2022 Revised: 26 August 2022 Accepted: 30 August 2022 Publication: 30 September 2022

Morphogenic Computation and Generalized Modified Order Crossover, *The 2005 IEEE Congress on Evolutionary Computation*, **2**: 1372-1379. 2005.

[7]. Giffler, B. and Thompson, G.L. Algorithms for Solving Production Scheduling Problems, *Operations Research*, **8**(4): 487-503. 1960.

[8]. Goncalves, J. F, Mendes, J.J, and Resende, M.C.G., A Hybrid Algorithm for the Job Shop Scheduling Problem, AT&T Technical Labs Report TD-5EAL6J. 2002.

[9]. Jackson, J. R., Scheduling a Production Line to Minimize Maximum Tardiness, Research Report 43, Management Science Research Projects, University of California, Los Angeles, USA. 1955.

[10]. Jackson, J. R. Simulation Research on Job-Shop Production, *Naval Research Logistics Quarterly*, **4**: 287-295. 1957.

[11]. Jain, A.S. and Meeran, S., A State-of-the-Art Review of Job-Shop Scheduling Techniques, European Journal of Operations Research, **113**: 390-434. 1999.109

[12]. Lian, Z., Gu, X., Jiao, B., A Similar Particle Swarm Optimization Algorithm for Permutation Flowshop Scheduling to Minimize Makespan, *Applied Mathematics and Computation*, 2005.

[13]. Liu, J., Zhong, W., Jiao, L., A Multiagent Evolutionary Algorithm for Constraint Satisfaction Problems. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, **36**: (1): 54-73.2006.

[14]. Muth J.F., Thompson G.L., Industrial Scheduling, Englewood Cliffs, New Jersey, Prentice Hall. 1963.

[15]. Pang, W., Wang, K., Zhou, C., Dong, L., Liu, M., Zhang, H., Wang, J., Modified Particle Swarm Optimization Based on Space Transformation for Solving Traveling Salesman Problem. *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, **4**: 26-29. 2004.

[16]. Pirlot, M., General Local Search Methods, *European Journal of Operational Research*, **92**: 493-511. 1996.

[17]. Rowe, A. J., Jackson, J. R., Research Problems in Production Routing and Scheduling, *Journal of Industrial Engineering.*, **7**: 116-121. 1956.

[18]. Smith, W. E., Various Optimizers for Single Stage Production, *Naval Research Logistics Quarterly*, **3**: 59-66. 1956.

[19]. Song, M., Gu, G., Research on Particle Swarm Optimization: A Review, *Proceedings of the 2004 Iternational Conference on Machine Learning and Cybernetics*,**4**: 26-29. 2004.

[20]. Lawrence, S., Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement), *Graduate School Ind. Adm.*. Pittsburgh, PA, Carnegie Mellon Univ., 1984.

[21]. Tasgetiren M. F., Yun-Chia Liang, Sevkli M., Gencyilmaz G., Particle Swarm Optimization Algorithm for Makespan and Maximum Lateness Minimization in Permutation Flowshop Problem, *4th International Symposium on Intelligent Manufacturing Systems*, Sakarya, Turkey, 431-441. 2004.

[22]. Ta[getiren M. F., Sevkli M., Yun-Chia Liang, Gençy]lmaz G., Particle Swarm Optimization Algorithm for Single-machine Total Weighted Tardiness Problem, *Congress on Evolutionary Computation*, Portland, OR, **2**: 19- 23. 2004.

[23]. Wang, K., Huang, L., Zhou, C., Pang, W., Particle Swarm Optimization for Traveling Salesman Problem. *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, **3**: 2-5. 2003.

[24]. Weijun, X., Zhiming, W., Wei, Z., and Genke, Y., A New Hybrid Optimization Algorithm for the Job Shop Scheduling Problem. *Proceeding of the 204 American Control Conference*. Boston, MA, **6**: 5552-5557. 2004.

[25]. Xia, W., Wu, Z., An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems. *Computers and Industrial Engineering*, **48**: 409-425. 2005.

[26]. Engin and Alper Doyen, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", Future generation computer systems, Vol. 20, pp. 1083- 1095, 2004.

[27]. Fantahun M. Defersha and Mingyuan Chen, "A hybrid genetic algorithm for flow shop lot streaming with setups and variable sub lots", International Journal of Production Research, Vol. 48, No. 6, pp. 1705-1726, 2010.

[28]. Fantahun M. Defersha and Mingyuan Chen, "A genetic algorithm for one-job m-machine flow shop lot streaming with variable sub lots", International Journal of Operational Research, Vol. 10, No.4, pp. 458 – 468, 2011.