The Job Scheduling Problem

Muddassir Ahmed
School of Computing
National College of Ireland
Dublin, Ireland
x23138688@student.ncirl.ie

***Abstract*— The report evaluates the performance for optimization algorithms—Integer Linear Programming (ILP), Tabu Search, and Particle Swarm Optimization (PSO), it is used for solving the Job-Scheduling Problem (JSP). The goal is to minimize the processing time completion on multiple machines. The algorithms are compared based on the time required to find solution and the final completion time for the last machine. Results indicate that ILP, while time-consuming, provides ideal solutions. Tabu Search shows IP solution time 0.05 seconds and tabu search solution time 0.01 seconds and final cost showing 75 result, while Particle swarm optimization give best solution that is found [2,5,1,6,4,3,0]. It gives IP solution time 0.03 seconds, PSO solution time 0.01 seconds and final cost is 75. This demonstrates the trade-offs between computational efficiency, solution quality, and heuristic nature of optimization algorithms.**

Keywords—Job Scheduling Problem, Linear Programming, Tabu, Particle Swarm Optimization (PSO).

I. INTRODUCTION

Efficient resource allocation is important in different industries. The Job-Scheduling Problem (JSP) highlight the need by assigning jobs to resources while optimizing performance metrics. This paper explores the essence, historical roots, and motivations behind studying JSP, and introduces the algorithms used for its optimization. In this new era allocation of the resources becomes very difficult component for success and productivity in the vast range of the field. A question always arrives of a manufacturing plant for attempting and reducing downtime to meet its productivity output. Data centres managing thousands of tasks in seconds with the ability of schedule.

Job scheduling problem is one of the most challenging problem. In this scheduling problem the optimization theory has been studied extensively by different researches and practitioners for many years. In board of the review, it will consider its presence history and areas in this problem. In this report an introduction is for such comprehensive exploration and explaining that what job scheduling problem is used and why it is important. It is presenting how following papers are structured.

A. Job-Scheduling Problem

In job scheduling problem some jobs were given to machine for processing with the specified processing level and time periods and it may have some precedence of the relationship and some set of resources such as processors and different machines and provide schedule for processing. Some instance and assignment of each job for processor and then follows some acceptable timetable to minimize or maximize as certain of performance measures.

Different techniques were developed for individual dispatcher for each machine [1]. Scheduling performance can be measured which can be include the minimization of total completion of the time and it is maximization of the throughput. Scheduling of the job can be meet deadlines while minimizing resource utilization or some jobs processing cost [2]. This problem of the optimization is much complex in multiple reasons which is including the combinatorial nature of the problem. It is presence of requisites, resource limitations, and balancing multiple objectives. The optimal solution is to the job scheduling problem is difficult to find, and it is often even intractable, especially for large instances. Adaptive machines were also developed for optimizing job scheduling problems [3]. Some categories are used in recent literature review those focus in different designing models which can help to choose optimal design for JSP [4].

In the availability of scheduling algorithms in different practitioners uses own expertise for making decision in different critical conditions. These expertise comes from their experience and literature support to understand scheduling problem in meaningful and easy way. Data mining can be use for understanding the relationship between travelling salesman problem [5]. Machine learning can be used to understand why prediction and job scheduling problem is easy or difficult in JSP.

B. Seminal Contributions

In the early time of the computer science and computational science, Job-Scheduling Problem has its roots in the early days of operations research and computer science. Different researchers conducted many experiments on JSP problems, including problem with or without precedence constraints, its resource constraints, and machine dependent and machine independent problems. This is due to the importance of job scheduling across all industries [6].

C. Motivation for Job-Scheduling Problem

JSP findings on optimal schedules and efficient algorithms for its determination them is a highly complicated and it is on some machine impossible task. However, Job scheduling problem is ubiquitous in the real life and its findings solutions to it could yield high dividends. In different manufacturing units, some job scheduling can minimize its production time and cost due to scheduler and it can increase customer satisfaction. In the machine computing, scheduling of the computing tasks can increase system overall performance and resource can be utilize very efficiently and optimally. Moreover, job scheduling problem sometimes used as a heuristic problem. Heuristic techniques can be used for these types of the problem and it optimize algorithm test problem. When solving these types of the problem, one can compare the

effectiveness of the algorithm and its heuristic to optimize the problem.

The processing times per job and machine:

[[1, 2, 2, 6],
[3, 5, 5, 4],
[1, 3, 7, 7],
[9, 6, 9, 8],
[9, 5, 1, 1],
[6, 8, 6, 7],
[7, 9, 3, 9]]

The optimal job schedule: Sequence [0, 2, 1, 4, 6, 5, 3]. The processing time of the optimal job schedule is 52.

Optimal job schedule in detail:

Diagram 1 showing overall job scheduling details of each machine. There are total 4 machines and different jobs.

```
print(schedule())
```

		Machine: 0		Machine: 1		Machine: 2		Machine: 3
		Idle: 0		Idle: 1		Idle: 3		Idle: 5
		Start: 0		Start: 1		Start: 3		Start: 5
Job: 0	Wait: 0	Proc: 1	Wait: 0	Proc: 2	Wait: 0	Proc: 2	Wait: 0	Proc: 6
		Stop: 1		Stop: 3		Stop: 5		Stop: 11
		Idle: 0		Idle: 0		Idle: 1		Idle: 2
		Start: 1		Start: 3		Start: 6		Start: 13
Job: 2	Wait: 1	Proc: 1	Wait: 1	Proc: 3	Wait: 0	Proc: 7	Wait: 0	Proc: 7
		Stop: 2		Stop: 6		Stop: 13		Stop: 20
		Idle: 0		Idle: 1		Idle: 5		Idle: 3
		Start: 2		Start: 7		Start: 18		Start: 23
Job: 1	Wait: 2	Proc: 3	Wait: 2	Proc: 5	Wait: 6	Proc: 5	Wait: 0	Proc: 4
		Stop: 5		Stop: 12		Stop: 23		Stop: 27
		Idle: 0		Idle: 0		Idle: 1		Idle: 0
		Start: 5		Start: 12		Start: 24		Start: 27
Job: 4	Wait: 5	Proc: 7	Wait: 0	Proc: 9	Wait: 3	Proc: 3	Wait: 0	Proc: 9
		Stop: 12		Stop: 21		Stop: 27		Stop: 36
		Idle: 0		Idle: 0		Idle: 0		Idle: 0
		Start: 12		Start: 21		Start: 27		Start: 36
Job: 6	Wait: 12	Proc: 9	Wait: 0	Proc: 6	Wait: 0	Proc: 9	Wait: 0	Proc: 8
		Stop: 21		Stop: 27		Stop: 36		Stop: 44
		Idle: 0		Idle: 0		Idle: 2		Idle: 0
		Start: 21		Start: 27		Start: 38		Start: 44
Job: 5	Wait: 21	Proc: 6	Wait: 0	Proc: 8	Wait: 3	Proc: 6	Wait: 0	Proc: 7
		Stop: 27		Stop: 35		Stop: 44		Stop: 51
		Idle: 0		Idle: 1		Idle: 0		Idle: 0
		Start: 27		Start: 36		Start: 44		Start: 51
Job: 3	Wait: 27	Proc: 9	Wait: 0	Proc: 5	Wait: 3	Proc: 1	Wait: 6	Proc: 1
		Stop: 36		Stop: 41		Stop: 45		Stop: 52

Diagram 1

II. ALGORITHMS

A. *Tabu Search*

It is uses a local or neighborhood search procedure while avoiding previously visited states. It mitigates local optima and explores new solution areas. It is use for local search model and it is used to prevent cycling to explore decision space. Measure can be use with the help of tabu that track the solution recently visited to avoid any further visits. Moreover, tabu can help to discover new areas for the solution space.

Diagram 2 shows schedule of tabu search. It has four columns, job colums shows job number and second column showing machine number that which job is allocated to particular machine. Column number third and fourth showing job start time and stop time of the job.

Tabu Search Schedule:

Job	Machine	Start Time	Stop Time
0	0	0	1
0	1	0	2
0	2	0	2
0	3	0	6
2	0	1	2
2	1	2	5
2	2	2	9
2	3	6	13
5	0	2	8
5	1	5	13
5	2	9	15
5	3	13	20
6	0	8	15
6	1	13	22
6	2	15	18
6	3	20	29
1	0	15	18
1	1	22	27
1	2	22	27
1	3	29	33
3	0	18	27
3	1	27	33
3	2	27	36
3	3	33	41
4	0	27	36
4	1	33	38
4	2	36	37
4	3	41	42

Diagram 2

Derivation:

Tabu Search iteratively improves solutions while preventing cycling through a tabu list. It balances exploration and exploitation. It is used for search operate in different iterations for improving solutions and previous visited states as avoided. In every iteration neighbour are identified and best one always selected. In tabu aspiration condition and all other possible accepting moves are allowed for solution of next accepted state.

Performance Metrics:

```
IP Solution Time: 0.05 seconds, Final Time: 0
Tabu Search Solution Time: 0.01 seconds, Final Cost: 75
```

Performance metrics gives IP solution and its time in our experiment is 0.05 with final 0 time.

Runtime: Tabu Search shows moderate runtime efficiency, with a runtime of 0.01 seconds in our experiment. In our problem runtime of tabu is moderate for the solution to explore and it execute its decision for analysing capabilities.

Accuracy: In this part we will navigate the solution and try to acquire the improvement for compare to local search. It did not provide optimum solution. In our findings the final cost is 75 and Tabu search solution time is 0.01 seconds. It gives overview that our model can cover optimal solution if process are limited.

B. *Particle Swarm Optimization (PSO)*

In computational science, PSO is a computational method. It is used to optimizing a problem by iteratively trying to improve a candidate solution, with the regard to given measure of quality. It gives solution of the problem having a population of the candidate solutions. Where the dubbed particles, moving these particles around in the search space. According to the simple and mathematical formula over the particle's position and velocity.

PSO job scheduling is showing in diagram 3. Job number and machine number is showing in first and second columns and job start and stop also showing in third and fourth columns.

PSO Schedule:

Job	Machine	Start Time	Stop Time
2	0	0	1
2	1	0	3
2	2	0	7
2	3	0	7
5	0	1	7
5	1	3	11
5	2	7	13
5	3	7	14
1	0	7	10
1	1	11	16
1	2	13	18
1	3	14	18
6	0	10	17
6	1	16	25
6	2	18	21
6	3	18	27
4	0	17	26
4	1	25	30
4	2	25	26
4	3	27	28
3	0	26	35
3	1	30	36
3	2	30	39
3	3	28	36
0	0	35	36
0	1	36	38
0	2	39	41
0	3	39	45

Diagram 3

Derivation: Particle Swarm Optimization (PSO) maintains a swarm of particles that update their positions and velocities which is based on personal and global best positions. This facilitates effective search for solutions.

PSO always starts with initial population then randomly spread over possible solution space. Velocity include inertia, cognitive and social components. These elements oversee speed and moment of particles towards favourable regions and it determine best trade-off between the execution and exploration.

Performance Metrics:

```
IP Solution Time: 0.03 seconds, Final Time: 0
PSO Solution Time: 0.01 seconds, Final Cost: 75
```

Particle Swarm Optimization IP solution time is calculating in seconds and it is observed 0.03 seconds.

Runtime: PSO exhibits efficient and parallel search capabilities with high dimensional various objective problems. This algorithm take 0.01 seconds runtime and in our experiment PSO is comparatively fast with execution time

Accuracy: PSO may converge to suboptimal solutions and it is seen in our experiment's final cost of 75. In Tabu search PSO could not give guarantee for global optimality in the case of high dimension and solution space. Its final cost calculates as 75 in experiment which suggest PSO can cover suboptimal solution under different constraints of algorithm.

III. Results and Evaluations

Two algorithms were implemented for Job Scheduling problem. Tabu search and particle swarm optimization (PSO), these are used for performance comparison with integer programming solution that is offer context for the effectiveness of the algorithms. Results indicate that ILP, while time-consuming, provides ideal solutions. Tabu Search shows IP solution time 0.05 seconds and tabu search solution time 0.01 seconds and final cost showing 75 result, while Particle swarm optimization give best solution that is found [2,5,1,6,4,3,0]. It gives IP solution time 0.03 seconds, PSO solution time 0.01 seconds and final cost is 75. This demonstrates the trade-offs between computational efficiency, solution quality, and heuristic nature of optimization algorithms.

References

[1] Zhang, Tao, and Oliver Rose. "Scheduling in a flexible job shop with continuous operations at the last stage." Journal of Simulation 10 (2016): 80-88..

[2] Karger, David R., Cliff Stein, and Joel Wein. "Scheduling algorithms." Algorithms and theory of computation handbook 1 (1999): 20-20

[3] Li, Li, et al. "Data-based scheduling framework and adaptive dispatching rule of complex manufacturing systems." The International Journal of Advanced Manufacturing Technology 66 (2013): 1891-1905.

[4] Branke, Jürgen, et al. "Automated design of production scheduling heuristics: A review." IEEE Transactions on Evolutionary Computation 20.1 (2015): 110-124.

[5] Smith-Miles, Kate, Jano Van Hemert, and Xin Yu Lim. "Understanding TSP difficulty by learning from evolved instances.". Springer Berlin Heidelberg, 2010.

[6] Lee, Yun-Han, Seiven Leu, and Ruay-Shiung Chang. "Improving job scheduling algorithms in a grid environment." Future generation computer systems 27.8 (2011): 991-998.