International Journal of Advanced Statistics and Probability, 11 (2) (2024) 45-51

SPC

International Journal of Advanced Statistics and Probability

Website: www.sciencepubco.com/index.php/IJASP

Research paper

Optimal solution of printing press-flexible job-shop scheduling problem (PP-FJSSP) with exact and heuristic algorithms

Nse Udoh [1] *, Nkereuwem Okonna [2], Iniobong Uko [2]

[1] *Senior Academics, Department of Statistics, University of Uyo, Nigeria*
[2] *Research Scholars, Department of Statistics, University of Uyo, Nigeria*
**Corresponding author E-mail: nsesudoh@uniuyo.edu.ng*

Abstract

In production related industries, optimizing scare resources and machine operations is one of the main goals toward improving operational efficiency and productivity while reducing cost. The Job shop scheduling problem (JSSP) model where one job can be process on one type of machine at a particular time is key to achieving this. However, Flexible job shop scheduling problem (FJSSP) extends JSSP by allowing machine operation to be processed by any qualified alternative machines. On this premise, this work seeks to optimize the operation of printing press system with j-jobs, k-tasks and m-machines using two optimization models viz: Integer Linear Programing (ILP) and Alldifferent Constraints Programing (ACP) models with a view to identifying the best model for this class of problem in terms of minimum makespan, optimality gap and computational time criteria as performance indicators. The ACP model yielded a comparative better result with makepan value of 233 minutes and optimal solution time of 0.21 seconds as against the ILP model with makespan value of 233 minutes and optimal solution time of 1.87 seconds. Hence, the ACP model is recommended for optimal operation in the printing company.

Keywords: *Scheduling; Flexible Job-Shop; Alldiferent Constraint Programming; Makespan; Optimality Gap; Integer Linear Programming.*

1. Introduction

Flexible job shop scheduling problem (FJSSP) is an extension of JSSP that deals with more complex and difficult combinatorial problems in manufacturing system. A traditional JSSP consists of a group of jobs with various tasks that may be carried out by several eligible machines. Each machine in this system is capable of doing one operation at a time, uninterrupted, so that the operation sequence for handling the tasks is formed. In contrast to JSSP, tasks in flexible job shop scheduling problem (FJSSP) can be completed on several eligible machines from a pool of available machines. When scheduling a work, FJSSP enables the assignment of certain operations to alternative machines that are qualified. Here the solution space is more complex when compare to JSSP due to the flexibility of routing. Also, in FJSSP, some machines may have common functionalities among multiple routes and the sequence of operations on a machine can vary from one job to another. In this study, the Job scheduling problem in printing press is considered a FJSSP for optimization. Since according to [1], the availability of parallel resources is prevalent in real-life settings, making the study of this type of environment crucial from both a theoretical and practical standpoint.

Some benefits of job flexibility, which allows for seamless transitions between various jobs in the production system with no additional expense and danger of loss was highlighted by [2]. And [3] examined machine scheduling for similar parallel machine that is functioning with comparable speed and machine processing time for each task or job. Since the output rate and costs of a manufacturing facility depend on the schedules used to regulate the activity in the plant, hence, finding excellent scheduling solutions for FJSSP is of utmost significance to job-shop manufacturing environment, [4]. The FJSSP can be extended to consider various realistic factors, such as machine availability, order of acceptance, transportation costs, setup times, due dates, precedence constraints, resource constraints, etc. These extensions increase the complexity and diversity of the problem and require more sophisticated solution methods. Scheduling, according to [5] is a practice used in the sectors of the economy that include manufacturing which entails allocation of production resources that are available overtime to complete a set of job. Meanwhile, efficient and effective planning and management of FJSSP manufacturing process and resource management is one of the key factors that have been shown to boost productivity. Because of this, proper operation scheduling is essential for both the producer and the consumer in terms of lead time. This is due to the fact that smart scheduling enables business owners to fully use their varied supply networks, [6]. Moreso, the FJSSP can be integrated with other planning and control activities, such as material requirements planning, enterprise resource planning, just-in-time manufacturing, etc., to achieve a more comprehensive and adaptive production system, [7]. As earlier postulated, most industrial and production systems, as well as most information processing settings, rely heavily on scheduling as a decision-making process.

The FJSSP can be formulated as an integer linear programming (ILP) model, but solving it exactly is often intractable for large-scale instances. Therefore, various heuristic and metaheuristic algorithms have been proposed to find approximate solutions in reasonable time. Some of the most popular methods include genetic algorithms, particle swarm optimization, simulated annealing, tabu search, ant colony optimization, and quantum annealing, [8]. These methods use different strategies to explore the search space and exploit the information gathered during the search process. They also differ in their performance, robustness, scalability, and applicability to different types of FJSSPs. Exact methods such as ILP which generate optimal solutions have been known to be computationally intractable for large instances. Due to computational complexities and demands from exact methods in solving hard problems, specifically FJSSP with various heuristic and meta heuristic algorithms have been utilized to find near-optimal solutions within reasonable amount of time. Hence, this research is a contribution to the search for optimal solution method for solving FJSSP - utilizing both exact method (ILP) of solving the printing press Flexible Job-Shop Scheduling Problem and heuristic method (Alldiferent constraints programming) with a view to identifying the best method for solving the printing press-FJSSP.

Exact optimization methods are techniques that aim to find the optimal solution to a scheduling problem by systematically exploring the solution space. These methods guarantee finding the best possible solution within a finite amount of time, but they can be computationally intensive and may be limited to small or moderately sized instances of scheduling problems. Some common examples of exact optimization methods include; Integer Linear Programming (ILP) where the decisions variables represent task, assignments, start times, completion times, etc., and the constraints capture resource and precedence relationships. Commercial solvers like Constraint Programming Linear with Extension (CPLEX), Gurobi, and Solving Constraint Integer Programs (SCIP) can be used to solve scheduling problems which are formulated as ILP, which is widely applicable but may become computationally intractable for large instances, [9].

Alldifferent Constraint Programming (ACP) is a declarative programming paradigm that has emerged as a powerful technique for solving combinatorial and optimization problems across diverse domains. It is rooted in artificial intelligence and operations research, providing a flexible and expressive approach to modeling and solving problems with complex constraints. In ACP, problems are represented by variables with defined domains, and constraints express relationships between these variables. The major distinction between ILP and ACP is the interval and sequence variables. ACP uses interval to denote the duration required to process a job. A sequence variable on the other hand is defined as set of permutated interval variables, [10].

Job shop production problem (JSPP) according to [11] is essential in a production system for customized batch jobs tailored towards specific requirements by clients and in a small-scale production. This kind of manufacturing system allows for the small-scale production of a wide range of customized goods. Most items manufactured require a unique setup and a job shop production flow that sequences each stage of the operation in turn. Also, [12] presented machine scheduling as the process of optimally assigning the processing time intervals of jobs on one or more machines. In a later development, [13] proposed a viable scheduling theory which serves as the foundation for machine scheduling, which has been successfully used in a variety of industries, including transportation, healthcare and agriculture.

Flexibility in the FJSSP can either be partial or absolute. According to [14], partial flexibility is when just some processes can be done by the machines that are available, and absolute flexibility is when all operations can be handled by any machine. This makes FJSSP the chosen job shop environment considered in this research. It gained recognition in recent times as a production system that can effectively handle the current competitive environment, [7]. Given its resemblance to actual production systems, flexible job shop scheduling can be applied to flexible manufacturing systems that include multitasking machines or multimachine environments.

Makespan is the total amount of time needed to complete all of the tasks in the system, according to [15]. He further asserted that makespan is the difference between the end-time of the final operation of the last job and the start-time of the first operation of the first job. In this research, makespan is used as one of the optimality criteria for comparing the strength and weakness of the exact and heuristic optimization models alongside optimality gap, which is a gap between best possible objectives and best-found objectives. An optimization solution is considered optimal when the upper and lower bounds values are equal, [6].

2. Methodology

2.1. ILP mathematical model for FJSSP

The problem considered in this study is a job shop consisting of M-machines where some of the m-machines have similar or common functionalities. The schedule of the production system comprises of a set of J independent jobs denoted by j and each of the j jobs is associated with several tasks denoted by K_j defined in a manner where each task k of job j can be implemented on a set of available and qualified machines. The total time it takes to complete task k of job j on machine m is known as the processing time designated as $P_{j,k,m}$. A task is setup in a way that it can start when the machine assigned completes its previous task.

In general, FJSSP consist of two combinatorial problems: task sequencing (TaS) and machine selection (MaS), on the assumptions that a machine can only work on one task alone at a given period. This constraint is known as the machine capacity constraint; processing of a task cannot be interrupted until completion. This is a case of preemption which is not allowed; each task associated with a job can only be assigned to only a single machine; when the sequence of task is defined, modification is not allowed. This is known as the precedence constraint and that all jobs and machines are available from time, 0.

The input of the model with j =1, 2, …, 8; m = 1, 2, …, 10 and k = 1, 2, …, 36 is defined as:

i) A set of jobs is defined and is denoted by $J = \{J_1, J_2, J_3, \dots, J_8\}$ each with specific number of tasks.

ii) A set of machines is defined as $M = \{M_1, M_2, M_3, \dots, M_{10}\}$ each with specific processing time for each task of each job.

In addition, other parameters and notations are presented with ILP model formulation. Some of the additional parameters are:

i) For each job j_i, and each task k, there is set of machines $M_{ij} = \{M_{ij1}, M_{ij2}, M_{ij3}, \dots, M_{ij10}\}$. These machines represent the alternative routes to performed task k_{ij}.

$$x_{jkj'k'} = \begin{cases} 1, & \text{if task k of job j can be performed after } k^{/} \text{ task of job } j^{/} \\ 0, & \text{otherwise} \end{cases}$$

ii) Ω: Denotes a very large number.

Some continuous variables are denoted as follows:

i) C_{jk}: Completion time of task k of job j.

ii) C_{max}: Total completion times for all jobs also known as the makespan

Some binary variables are denoted as follows:

- $$z_{j,k,m} = \begin{cases} 1; & \text{if task k of job j is processed by machine} \\ 0, & \text{otherwise} \end{cases}$$

The objective function of the model is as follows:

$$\text{Minimize } C_{max} \quad (1)$$

Subject to:

$$C_{max} \geq C_{jk};\ \forall j \in J, k \in K_j \quad (2)$$

$$C_{jk} \geq C_{jk-1} + \sum_{m \in M_{jk}} P_{jkm} z_{jkm};\ \forall j \in J, k \in K_j \quad (3)$$

$$\sum_{m \in M_{jk}} z_{jkm} = 1;\ \forall j \in J, k \in K_j \quad (4)$$

$$C_{j'k'} \geq C_{jk} + P_{j'k'm} - \Omega\left(2 + x_{jkj'k'} - z_{jkm} - z_{j'k'm'}\right); \quad (5)$$

$$\forall j > j' \in J, k \in K_j, k' \in K'_j, m \in M_{jk} \cap M_{j'k'}$$

$$C_{jk} \geq 0;\ \forall j \in J, k \in M_j \quad (6)$$

$$x_{jkj^ik^i}, z_{jkm} \in \{0,1\};\ \forall j > j' \in J, k \in K_j, k' \in K_j, m \in M \quad (7)$$

Equation 1 is the objective function of the ILP model which is the minimization of the makespan. The objective function and Equation 2 defined the makespan of the job schedule. Constraint in Equation 3 ensure that there is no overlap of tasks of a particular job. In other words, the completion time of task k of job j must be ≥ the completion time of the previous task $k - 1$ of the same job plus the processing time of task k. Constraint in Equation 4 is used to assigned each k task to eligible machines. Equation 5 ensures non overlapping of different tasks of different jobs assigned to the same machine. Constraint in Equations 6 and 7 defined the type of decision variables (continuous and binary).

2.2. Alldifferent constraint programming model

The two main distinctions between ILP model and ACP Optimizer model are interval and sequencing variables. For each task, an interval variable in the ACP Optimizer is establish which represents the amount of time that a task is processed. Decisions for the model are the interval variable's start and end points that are inside a broader interval $[\alpha, \beta)$. Formally, an interval variable x may be defined as a decision variable whose domain $(s, e) \subset \{[\alpha, \beta) | \alpha, \beta \in Z; \alpha \leq \beta\}$ where s and e denote the interval's start and end points respectively, and $l = e - s$ represents its length.
According to [18] Naderi and Roshanaei (2021), the ACP model for the FJSSP is given as follows:

$$\text{Minimize } C_{max} \quad (8)$$

Subject to:

$$\text{Task}_{jkm} = \text{IntervalVar}\left(P_{jkm}, \text{Optional}\right);\ \forall j \in J, k \in K, m \in M_{jk} \quad (9)$$

$$\text{Alternative}\left(\text{Task}^*_{jk}, \left\{\text{Task}_{jkm}: m \in M_{jk}\right\}\right);\ \forall j \in J, k \in K_j \quad (10)$$

$$\text{EndBeforeStart}\left(\text{Task}_{jk-1}, \text{Task}_{jk}\right);\ \forall j \in J, k \in K_j \quad (11)$$

$$\text{NoOverlap}\left(\text{Task}_{jki}: j \in J, k \in K_J \middle| m \in M_{jk}\right);\ \forall m \in M \quad (12)$$

$$C_{max} = \max_{j \in J}(EndOf(Task^*_{j|K_j|})) \quad (13)$$

Equation 8 defined the objective of the model as C_{max} while Equation 9 defined the interval variable for task k of job j processed on machine m. Equation 10 assign machine m for processing task k of job j given alternative and eligible machines. Equation 11 enforced a sequence constraint which ensures that task $k > 1$ can only begin after the completion of task $k - 1$. Equation 12 constraint ensures that machine m can only process one task at a time. while Equation 13 defined C_{max} as the maximum completion time of task k of job j. The ILP and ACP algorithms were developed using CPLEX Ilog Studio 22.12.1 and python 3.9.

3. Application

The wonder world printing press (WWPP) produces Text books, Magazines, Photo cards, etc. on commercial basis. Its production process consists of 8 jobs with 36 operations on 10 machines. The eight different jobs can be assigned to any of the ten printing machines. Hence, it qualifies as FJSSP. The Company wants to effectively schedule its production among the available machines to save time and cost.
The network diagram for the WWPP having 8 jobs, 10 machines with a total number of 36 operations is shown in Figure 1. The lines from left hand side of Figure 1 show how the different tasks; $k_1, k_2, ...$ of Jobs; J_1, J_2, … J_8 can be handled by any of the machines; M_1, M_2, M_{10} on the righthand side. For instance, Job_8 which consists of 2 tasks, k_1 and k_2 can be processed by 3 machines; M_6, M_9 and M_{10} where the order of operation depends on the machine availability and hence, qualifies for flexibility.

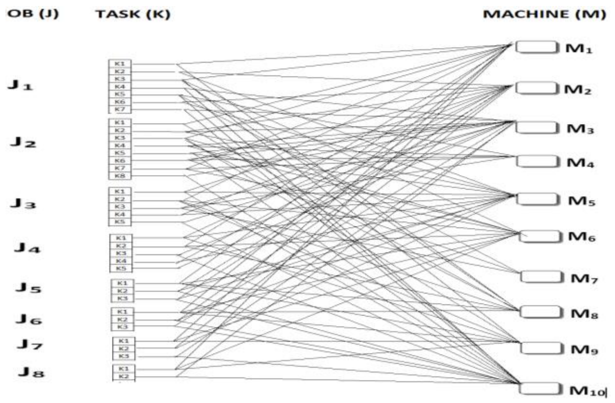

Fig. 1: Network Diagram of WWPP scheduling problem.

The respective ILP and ACP model formulation of the WWPP problem are as follows:

3.1. Mathematical Structure of ILP Problem

$Min\ Z\ =\ p_1k_{1,1}+p_2k_{1,2}+p_3k_{1,3}+p_4k_{1,4}+p_5k_{1,5}+p_6k_{1,6}+p_7k_{1,7}$

$s.t.\ t_{1,1}k_{1,1}+t_{1,2}k_{1,2}\ +t_{1,3}k_{1,3}\ +\ t_{1,4}k_{1,4}\ +\ t_{1,5}k_{1,5}\ +\ t_{1,6}k_{1,6}\ +t_{1,7}k_{1,7}\ \geq q_1$

$t_{2,1}k_{1,1}+t_{2,2}k_{1,2}\ +t_{2,3}k_{1,3}\ +\ t_{2,4}k_{1,4}\ +\ t_{2,5}k_{1,5}\ +\ t_{2,6}k_{1,6}\ +t_{2,7}k_{1,7}\ \geq q_2$

$t_{3,1}k_{1,1}+t_{3,2}k_{1,2}\ +t_{3,3}k_{1,3}\ +\ t_{3,4}k_{1,4}\ +\ t_{3,5}k_{1,5}\ +\ t_{3,6}k_{1,6}\ +t_{3,7}k_{1,7}\ \geq q_3$

$t_{4,1}k_{1,1}+t_{4,2}k_{1,2}\ +t_{4,3}k_{1,3}\ +\ t_{4,4}k_{1,4}\ +\ t_{4,5}k_{1,5}\ +\ t_{4,6}k_{1,6}\ +t_{4,7}k_{1,7}\ \geq q_4$

$t_{5,1}k_{1,1}+t_{5,2}k_{1,2}\ +t_{5,3}k_{1,3}\ +\ t_{5,4}k_{1,4}\ +\ t_{5,5}k_{1,5}\ +\ t_{5,6}k_{1,6}\ +t_{5,7}k_{1,7}\ \geq q_5$

$t_{6,1}k_{1,1}+t_{6,2}k_{1,2}\ +t_{6,3}k_{1,3}\ +\ t_{6,4}k_{1,4}\ +\ t_{6,5}k_{1,5}\ +\ t_{6,6}k_{1,6}\ +t_{6,7}k_{1,7}\ \geq q_6$

$t_{7,1}k_{1,1}+t_{7,2}k_{1,2}\ +t_{7,3}k_{1,3}\ +\ t_{7,4}k_{1,4}\ +\ t_{7,5}k_{1,5}\ +\ t_{7,6}k_{1,6}\ +t_{7,7}k_{1,7}\ \geq q_7$

$t_{8,1}k_{1,1}+t_{8,2}k_{1,2}\ +t_{8,3}k_{1,3}\ +\ t_{8,4}k_{1,4}\ +\ t_{8,5}k_{1,5}\ +\ t_{8,6}k_{1,6}\ +t_{8,7}k_{1,7}\ \geq q_8$

$t_{9,1}k_{1,1}+t_{9,2}k_{1,2}\ +t_{9,3}k_{1,3}\ +\ t_{9,4}k_{1,4}\ +\ t_{9,5}k_{1,5}\ +\ t_{9,6}k_{1,6}\ +t_{9,7}k_{1,7}\ \geq q_9$

$t_{10,1}k_{1,1}+t_{10,2}k_{1,2}\ +t_{10,3}k_{1,3}\ +\ t_{10,4}k_{1,4}\ +\ t_{10,5}k_{1,5}\ +\ t_{10,6}k_{1,6}\ +t_{10,7}k_{1,7}\geq q_{10}$

$K_{1j}\geq 0;\ j=1,2,\dots,7$

Z = Total processing time of the various tasks of wonder world printing press
$p_{i's}$ = Time (min) contribution coefficients. That is, the numerical values that express the per minute contribution to the time equation.
t_{ij} = The time (min) put in by the respective machine at each task.
$q_{i's}$ = The time (min) that we seek to utilize.
The decision variables are:
$K_{1,1}$ = Unit of colour separation
$K_{1,2}$ = Unit of paper cutting
$K_{1,3}$ = Unit of printing
$K_{1,4}$ = Unit of folding
$K_{1,5}$ = Unit of stapling
$K_{1,6}$ = Unit of trimming
$K_{1,7}$ = Unit of packaging

Based on the information provided, the ILP formulation of Wonder World Printing Press Scheduling Problem is as follows;

$Min\ Z\ =\ 15K_{1,1}+12K_{1,2}+148K_{1,3}+20K_{1,4}+55K_{1,5}+10K_{1,6}+20K_{1,7}$

$s.t.\ 45K_{1,1}+0K_{1,2}\ +100K_{1,3}+\ 0K_{1,4}\ +\ 0K_{1,5}\ +\ 0K_{1,6}\ +\ 0K_{1,7}\ \geq 146$

$0K_{1,1}+20K_{1,2}\ +0K_{1,3}\ +\ 0K_{1,4}\ +\ 0K_{1,5}\ +\ 30K_{1,6}\ +0K_{1,7}\ \geq 52$

$0K_{1,1}+\ 0K_{1,2}\ +0K_{1,3}\ +20K_{1,4}+\ 0K_{1,5}\ +\ 0K_{1,6}\ +\ 60K_{1,7}\ \geq 82$
$0K_{1,1}+\ 0K_{1,2}\ +0K_{1,3}\ +20K_{1,4}+\ 50K_{1,5}+\ 0K_{1,6}\ +\ 0K_{1,7}\ \geq 51$

$0K_{1,1} + 35K_{1,2} + 0K_{1,3} + 0K_{1,4} + 0K_{1,5} + 30K_{1,6} + 0K_{1,7} \geq 62$

$0K_{1,1} + 0K_{1,2} + 40K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 41$

$0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 0K_{1,4} + 45K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 46$

$60K_{1,1} + 0K_{1,2} + 90K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 152$

$0K_{1,1} + 0K_{1,2} + 200K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 202$

$0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 40K_{1,4} + 60K_{1,5} + 0K_{1,6} + 20K_{1,7} \geq 121$

$K_{1j} \geq 0;\ j = 1, 2, \dots, 7$

3.2. Mathematical structure of all different constraint programming model

The constraint of alldifferent model $(k_{1,1}, k_{1,2}, \dots, k_{1,n})$ is satisfied if variables $k_{1,1}, k_{1,2}, \dots, k_{1,n}$ have different values. According to [19] Refalo (2000), a sharp formulation of this constraint is given as;
Let $M = \cup_{i=1}^{n} D_i$ be the union of domains. The linearization of this constraint implies that each value of M can be given at most once to any of the variables $k_{11}, k_{12}, \dots, k_{1n}$. Hence, we have;

$$\partial\left(alldifferent(k_{1,1}, \dots, k_{1,n})\right) = \{\textstyle\sum_{i=1}^{n} \gamma_{ij} \leq 1\}$$

For the minimization problem considered in this work, we have;

$$\partial\left(alldifferent(k_{1,1}, \dots, k_{1,n})\right) = \{\textstyle\sum_{i=1}^{n} \gamma_{ij} \geq 1\}$$

Where $\gamma_{ij} = the\ product\ of\ minute\ input\ by\ each\ machine, and\ the\ unit\ of\ each\ task$

$$\partial(alldifferent(k_{11}, \dots, k_{17})) = \left\{\begin{array}{c} \gamma_{1,1} + \gamma_{1,2} + \gamma_{1,3} + \gamma_{1,4} + \gamma_{1,5} + \gamma_{1,6} + \gamma_{1,7} \geq 1 \\ \gamma_{2,1} + \gamma_{2,2} + \gamma_{2,3} + \gamma_{2,4} + \gamma_{2,5} + \gamma_{2,6} + \gamma_{2,7} \geq 1 \\ \vdots + + + + + \vdots \\ \gamma_{10,1} + \gamma_{10,2} + \gamma_{10,3} + \gamma_{10,4} + \gamma_{10,5} + \gamma_{10,6} + \gamma_{10,7} \geq 1 \end{array}\right\}$$

Then, the wonder world printing press scheduling information can be translated into the above model as follows;

$$\partial(alldifferent(k_{11}, \dots, k_{17})) = \left\{\begin{array}{c} 45K_{1,1} + 0K_{1,2} + 100K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 146 \\ 0K_{1,1} + 20K_{1,2} + 0K_{1,3} + 0K_{1,4} + 0K_{1,5} + 30K_{1,6} + 0K_{1,7} \geq 52 \\ 0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 20K_{1,4} + 0K_{1,5} + 0K_{1,6} + 60K_{1,7} \geq 82 \\ 0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 20K_{1,4} + 50K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 51 \\ 0K_{1,1} + 35K_{1,2} + 0K_{13} + 0K_{1,4} + 0K_{1,5} + 30K_{1,6} + 0K_{1,7} \geq 62 \\ 0K_{1,1} + 0K_{1,2} + 40K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 41 \\ 0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 0K_{1,4} + 45K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 46 \\ 60K_{1,1} + 0K_{1,2} + 90K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 152 \\ 0K_{1,1} + 0K_{1,2} + 200K_{1,3} + 0K_{1,4} + 0K_{1,5} + 0K_{1,6} + 0K_{1,7} \geq 202 \\ 0K_{1,1} + 0K_{1,2} + 0K_{1,3} + 40K_{1,4} + 60K_{1,5} + 0K_{1,6} + 20K_{1,7} \geq 121 \end{array}\right\}$$

$k_i \in \forall\ 1 \geq i \geq 0$

4. Results

The results of the ILP and ACP models' performances are presented in Table 1 and the makespan of ILP and ACP models in Figs 2 and 3 respectively.

Table 1: Optimal Solution for Printing Press (PP) Scheduling Problem with ILP and ACP

Model	Lower Bound	Upper Bound (Objective Value)	Iteration	Optimality Gap	Solved Time (s)
ILP	233	233 (optimal)	5	0	1.872
ACP	233	233 (optimal)	5	0	0.21

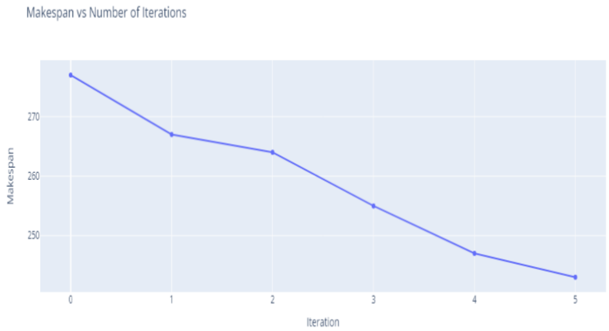

Fig. 2: Makespan Vs Number of Iteration for ILP Model.

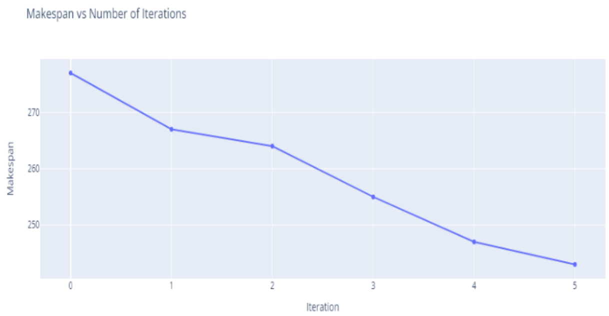

Fig. 3: Makespan vs. Number of Iterations for ACP Model.

3.1. Gantt chart schedule completion timeline

The Gantt chart of the solution schedule for ILP and ACP models are respectively presented in Figs 4 and 5. The timeline of the chart is expressed in date format.

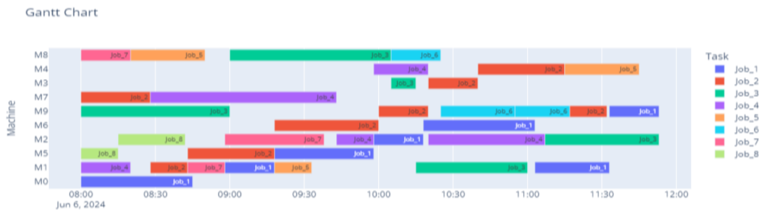

Fig. 4: Gantt Chart of ILP Model.

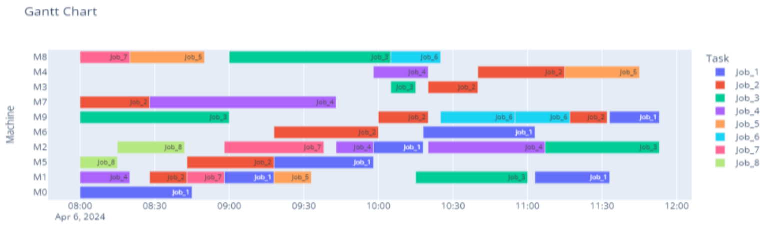

Fig. 5: Gantt Chart for ACP Model.

The solution of WWPP problem from the two optimization models showed that ACP solved the problem in 0.21 seconds while ILP solve it in 1.87 seconds with equal makespan of 223 minutes. Hence, ACP with 0.21 seconds is the preferred model for the WWPP problem. Gantt chart of the solution schedules showed that both models complete all the 8 jobs in 233 minutes. The plots of makespan vs. number of iterations show that it took 5 iterations for both ILP and ACP to achieved optimal solution in 1.87 and 0.21 seconds respectively. The results further demonstrate the advantage of ACP optimization model to the ILP model.

5. Conclusion

The superior performance of heuristics algorithm over exact method is further demonstrated with the utilization of ILP and ACP in this work. The ACP model has minimum job completion time for the WWPP problem, though at par with ILP in makespan and optimality gap criteria. Hence, the ACP is adjudged a better model for the WWPP problem and could be adaptable to any size of problem with similar features.

References

[1] Senthiil, P. V., Selladurai, V., & Rajesh, R. (2007). Parallel machine scheduling (PMS) in manufacturing systems using the ant colony optimization algorithmic rule. Journal of Applied Sciences, 7(2), 208-213. https://doi.org/10.3923/jas.2007.208.213.

[2] Roy, R. N. (2007). A modern approach to operations management. New Age International (P) Limited.

[3] Chaudhry, I. A., Mahmood, S., & Ahmad, R. (2010, June). Minimizing makespan for machine scheduling and worker assignment problem in identical parallel machine models using GA. Proceedings of the World Congress on Engineering, 3.

[4] Motaghedi-Larijani, A., Sabri-laghaie, K., & Heydari, M. (2010). Solving flexible job shop scheduling with multi-objective approach. International Journal of Industrial Engineering and Production Research, 21(4), 197-209.

[5] Agrawal, R., Pattanaik, L. N., & Kumar, S. (2012). Scheduling of a flexible job-shop using a multi-objective genetic algorithm. Journal of Advances in Management Research, 9(2), 178-188. https://doi.org/10.1108/09727981211271922.

[6] 23:25, 21/09/2024] +234 811 368 7403: [6] Krajewski, L. J., Malhotra, M. K., & Ritzman, L. P. (2016). Operations Management Processes and Supply Chains. Pearson Education.

[7] Ziaee, M., Mortazavi, J., & Amra, M. (2022). Flexible job shop scheduling problem considering machine and order acceptance, transportation costs, and setup times. Soft Computing, 1-17. https://doi.org/10.1007/s11740-022-01145-8.

[8] Schworm, P., Wu, X., Glatt, M., & Aurich, J. C. (2023). Solving flexible job shop scheduling problems in manufacturing with Quantum Annealing. Production Engineering, 17(1):105-115. https://doi.org/10.1007/978-3-030-27192-3.

[9] Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., Sterna, M., & Węglarz, J. (2019). Handbook on scheduling: From theory to practice. Springer. https://doi.org/10.1111/poms.13373.

[10] Naderi, B., Roshanaei, V., Begen, M. A., Aleman, D. M., & Urbach, D. R. (2021). Increased surgical capacity without additional resources: Generalized operating room planning and scheduling. Production and Operations Management, 30(8): 2608-2635.

[11] Pinedo, M. (2012). Scheduling: theory, algorithms and systems. Springer Science & Business Media. LLC. https://doi.org/10.1007/978-1-4614-2361-4.

[12] Vaghefinezhad, S., & Wong, K. Y. (2012). A genetic algorithm approach for solving a flexible job shop scheduling problem. arXiv preprint arXiv:1207.2253.

[13] 23:26, 21/09/2024] +234 811 368 7403: [13] Anshulika & Bewoor, L. A. (2017). A genetic algorithm approach for solving a job shop scheduling problem, 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 1-6. https://doi.org/10.1109/ICCCI.2017.8117720.

[14] Vital-Soto, A., Azab, A., & Baki, M. F. (2020). Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility. Journal of Manufacturing Systems, 54: 74-93. https://doi.org/10.1016/j.jmsy.2019.12.003.

[15] Bayat Movahed, S. (2014). Linear programming assisted genetic algorithm for solving a comprehensive job shop lot streaming problem (Doctoral dissertation, University of Guelph).