An Effective Mathematical Model for Advanced Planning and Scheduling in Multi-Plant Chain

Miguel Fernández Pérez, PhD
Department of Engineering, Pontifical Catholic University of Peru, Lima 32, Peru, mfernandezp@pucp.edu.pe

***Abstract*– This work addresses the advanced planning and scheduling problem. This problem is an extension of the Flow Shop, Job Shop, Flexible Job Shop, and Integrated Resource Selection and Operation Sequences problems, being its main characteristics the precedence relationship between operations that are part of customer orders, the batch size of the orders, the flexibility of the machines, which are in different plants. In order to solve the problem, a mixed-integer linear programming model is proposed, which seeks to minimize the makespan. The computational results show that the mathematical model was able to solve the cases or instances generated in a satisfactory way.**

***Keywords*-- Advanced planning and scheduling problem; multi-plant chain; mixed-integer linear programming model.**

I. INTRODUCTION

In a globalized and highly competitive world in which companies compete for space, customers, recognition and, above all, conditions to remain competitive and profitable in the market, it is essential to develop mechanisms that help these companies make decisions. In a factory, the set of critical factors involved in the production process is extensive. Among them, the variable demand with short delivery times, the acquisition and stock of raw materials, the stock of semi-finished and finished products, the use of machinery and available labour, and the adequate use of physical space, for storage can be mentioned. In this area, the integration of all these factors is significant for proper decision-making. Therefore, it is necessary to study and develop methodologies that address all these factors in an integrated way. Production scheduling problems are common in the day-to-day running of a factory. Faced with this scenario, finding reasonable solutions that solve these problems can become highly complex.

This paper seeks to develop a mixed-integer linear programming model for Advanced Planning and Scheduling (APS), to support decision-making in a productive system. This problem extends the Flow Shop, Job Shop, Flexible Job Shop, and Integrated Resource Selection and Operation Sequences problems [1].

According to Reference [2], the APS is a system that serves as an umbrella over the supply chain, allowing the extraction of information in real-time, with which viable programming can be developed, and also provides a fast and reliable response to customers. In this way, APS can provide reasonable product delivery dates to customers; and help reduce lead time, stock levels and costs in the supply chain.

According to Reference [3], APS tends to adopt a holistic vision to optimize operations in a plant and all activities from supplier to customer.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2022.1.1.539
ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

The APS has been studied by several researchers, resulting in various proposals to solve this problem in a multi-plant chain (MPC) production environment of a supply chain. In an MPC environment, the processes can be executed in more than one plant in coordinated management and information exchange. Each plant is equipped with its own and even flexible resources. Reference [4] proposed a heuristic method for this problem, considering the master production plan. Reference [5] studied a manufacturing and assembly process, considering capacity limitations and seeking to minimize costs in a supply chain. Reference [6] developed a heuristic method for programming an MPC, considering internal transport, transport between plants, and order lot size. Reference [7] proposed a model that integrates planning and scheduling operations to minimize the total backlog in an MPC; to solve this problem, the authors developed a genetic algorithm. Reference [8] proposed an approach to the APS problem in an MPC environment to minimize the makespan and decide the sequence of operations, the allocation of machines, considering precedence restrictions, the flexibility of the sequence of operations and availability of machines. Reference [3] formulated an APS problem as a flexible manufacturing system and presented the operation-based genetic algorithm method to minimize the makespan. Reference [9] developed a multiobjective genetic algorithm with an adaptive strategy, which calculates the probability of crossing and mutation in each generation to improve the performance of the genetic algorithm. Here, the objectives considered are to minimize the makespan and balancing of the workload in an MPC environment.

According to Reference [10], the industries are developed in global chains, including several manufacturing locations and several suppliers, including outsourcing. In this context, the manufacturing supply chain (MSC) tries to optimize an entire system. APS in an MSC environment was developed by Reference [11], who defined it as a problem that seeks to determine the sequence of operations of a set of orders and the way that they must be executed in the machines to meet the delivery deadlines; also considering outsourcing, so that one or more performance measures are optimized. Reference [10] proposed an adaptive genetic algorithm for the APS problem in an MSC environment. Reference [12] proposed a genetic algorithm with a new mutation operator, based on disturbance and local search, to minimize the makespan while avoiding generating penalties for late delivery of orders. Reference [13] solved the APS problem using a multiobjective genetic algorithm, simultaneously minimizing the makespan, the balancing of the workload of the machines and the total transport time between the machines. Reference [14] developed a multiobjective genetic algorithm with an adaptive strategy, seeking to simultaneously minimize the downtime of the machines and the penalty for delay or anticipation in the delivery of orders.

II. PROBLEM FORMULATION

Given a set of K orders of batch size q_k $(k = 1, \dots, K)$, where each order k is composed of a series of J_k operations that have a precedence relation. In addition, there is a set of D plants, where each plant d $(d = 1, \dots, D)$ has M_d machines. Each operation o_{ki} $(k = 1, \dots, K, i = 1, \dots, J_k)$ must be processed by some machine M_l $(l = 1, \dots, L)$, within the set of machines M_{ki} capable of processing the operation o_{ki}. Consider that the batch size q_k is divided into equal sub-batches, called the unit load when it is processed on the machines. To process customer orders, setup times between operations, transport times between machines, among other operational conditions, must be included in the programming. If transport between plants is necessary, the transfer lot is the lot size. We want to find the sequence of order operations and the allocation of operations in the machines so the precedence relationship between operations and the availability of the machines are satisfied and is optimal concerning to the minimization of the makespan [3].

The assumptions considered in the APS problem are:

- All orders arrive simultaneously.
- All machines are available from the start.
- The machines cannot stop until they finish processing the batch size.
- For the same order, the processing in one operation begins after the load unit of the predecessor operation has been completed and transported, as long as these operations are processed on different machines.
- For the same order to be processed in different plants, an operation begins after the lot size of the predecessor operation has been completed and transported.
- The transport time between different machines is considered.
- The setup time between operations is considered.

The notation used for the indices, parameters, and decision variables to formulate the mathematical model of the APS problem is detailed below:

Sets

k: Order index, $k = 1, 2, \dots, K$
i, j: Operation index, $i, j = 1, 2, \dots, J$
m, n: Machine index, $m, n = 1, 2, \dots, N$
d: Plant index, $d = 1, 2, \dots, D$

Parameters

K: Number of orders
J: Number of operations
N: Number of machines
D: Number of plants
r_{ij}: Precedence relationship between order operations

$$r_{ij} = \begin{cases} 1, \text{ if operation } i \text{ precedes operation } j \\ 0, \text{ opposite case} \end{cases}$$

A_{im}: Operation-machine correspondence matrix

$$A_{im} = \begin{cases} 1, \text{ if operation } i \text{ can be processed by machine } m \\ 0, \text{ opposite case} \end{cases}$$

k_i: Returns the order index of the operation i
q_i: Returns the batch size of the operation order i
p_{im}: Operation processing time i in machine m
L_m: Machine capacity m
u_{ij}: Order loading operation i for operation j
t^S_{mn}: Transport time between the machine m and machine n (includes transport time of machines that are in different plants)
t^U_{ij}: Setup time between operation i and operation j
b_m: Returns the plant index where the machine belongs m

Variables

$$y_{ij} = \begin{cases} 1, \text{ if operation } i \text{ is performed immediately before operation } j \\ 0, \text{ opposite case} \end{cases}$$

$$x_{im} = \begin{cases} 1, \text{ if machine } m \text{ is selected for operation } i \\ 0, \text{ opposite case} \end{cases}$$

s_i: Operation start time i
c_i: Operation completion time i
c_M: Makespan

The mathematical model of the APS problem is formulated as:

$$\min c_M \tag{1}$$

subject to:

$$\sum_{m=1}^{N} x_{im} = 1 \quad \forall i \tag{2}$$

$$s_i + q_i \sum_{m=1}^{N} p_{im} x_{im} = c_i \quad \forall i \tag{3}$$

$$y_{ij} = 1 \quad \forall i, j \mid r_{ij} = 1 \tag{4}$$

$$y_{ij} + y_{ji} = 1 \quad \forall i, j \tag{5}$$

$$s_j \geq c_i + t^U_{ij} - M\left(3 - x_{im} - x_{jm} - y_{ij}\right) \quad \forall i, j, m \mid A_{im} = 1, A_{jm} = 1 \tag{6}$$

$$s_j \geq s_i + u_{ij} p_{im} + t^S_{mn} - M\left(3 - x_{im} - x_{jn} - y_{ij}\right) \quad \forall i, j, m, n \mid k_i = k_j, m \neq n, b_m = b_n \tag{7}$$

$$c_j \geq c_i + t^S_{mn} + u_{ij}p_{jn} - M\left(3 - x_{im} - x_{jn} - y_{ij}\right)$$
$$\forall i,j,m,n|\ k_i = k_j, m \neq n, b_m = b_n \quad (8)$$

$$s_j \geq c_i + t^S_{mn} - M\left(3 - x_{im} - x_{jn} - y_{ij}\right)$$
$$\forall i,j,m|k_i = k_j, m \neq n, b_m \neq b_n \quad (9)$$

$$c_M \geq c_i \ \ \forall i \quad (10)$$

$$\sum_{i=1}^{J} q_i p_{im} x_{im} \leq L_m \ \ \forall m \quad (11)$$

$$x_{im} \in \{0,1\} \ \ \forall i,m|A_{im} = 1 \quad (12)$$

$$y_{ij} \in \{0,1\} \ \ \forall i,j|i \neq j \quad (13)$$

$$s_i, c_i \geq 0 \ \ \forall i \quad (14)$$

$$c_M \geq 0 \quad (15)$$

This mathematical model considers the minimization of the makespan. Constraint (2) ensures that each operation selects only one machine for processing. Constraint (3) determines the completion time of each operation. Constraints (4) and (5) guarantee the viability of the sequence between operations. Constraint (6) avoids interference between operations assigned to the same machine; that is, an operation cannot be processed until the predecessor operation is completed. In addition, the setup time is added. Constraints (7) and (8) ensure the transport of an order between two different machines in the same plant. These restrictions must be satisfied simultaneously to guarantee that operations are executed on one machine without interruption. Here $u_{ij}p_{im}$ and $u_{ij}p_{jn}$ represent the order loading processing times. Constraint (9) allows transport between different plants. An operation cannot be carried out at another plant until the lot size of the predecessor operation is completed. Constraint (10) determines the makespan as the maximum completion time of all operations. Constraint (11) limits the available production time of each machine. Constraints (12) - (15) define the domain of the decision variables.

III. COMPUTATIONAL EXPERIMENTS

There is a group of instances available in the literature and found in Reference [8]. In the resolution of all the instances, there is the possibility of working in two plants; each has three machines. The mathematical model was implemented in the AMPL software using the CPLEX 12.9 solver and run on a computer equipped with an AMD Ryzen 5 3500U processor with 2.1 GHz and 8 GB of RAM. For each instance, the assignment of the operations in the machines of each plant and the start and end time of each operation is presented:

TABLE I.
INSTANCE N1 - 1 ORDER × 4 OPERATIONS

Plants	Machines	Operation (start time/end time)	
Plant 1	Machine 1		
	Machine 2		
	Machine 3		
Plant 2	Machine 4	1(0/200)	2(217/457)
	Machine 5	3(55/375)	
	Machine 6	4(313/513)	

TABLE II.
INSTANCE N2 - 2 ORDERS × 7 OPERATIONS

Plants	Machines	Operation (start time/end time)	
Plant 1	Machine 1	6(0/210)	7(232/792)
	Machine 2	5(35/665)	
	Machine 3	3(592/792)	
Plant 2	Machine 4	1(0/200)	2(217/457)
	Machine 5		
	Machine 6	4(313/513)	

TABLE III.
INSTANCE N3 - 3 ORDERS × 12 OPERATIONS

Plants	Machines	Operation (start time/end time)		
Plant 1	Machine 1	7(155/715)	4(810/1050)	
	Machine 2	5(0/630)	6(700/1050)	
	Machine 3	8(0/300)	3(507/707)	
Plant 2	Machine 4	1(0/200)	2(217/457)	11(543/903)
	Machine 5	10(350/710)	12(750/1050)	
	Machine 6	9(417/837)		

TABLE IV.
INSTANCE N4 - 4 ORDERS × 17 OPERATIONS

Plants	Machines	Operation (start time/end time)		
Plant 1	Machine 1	1(0/280)	7(291/851)	17(939/1089)
	Machine 2	5(0/630)	6(679/1029)	
	Machine 3	8(0/300)	3(620/820)	15(849/1029)
Plant 2	Machine 4	13(0/180)	2(330/570)	11(600/960)
	Machine 5	14(65/335)	10(381/741)	12(789/1089)
	Machine 6	16(242/392)	9(448/868)	4(889/1089)

Note that the maximum completion time in operations is the makespan. All these instances have been resolved satisfactorily. Besides, it should be noted that the solutions have been obtained in a reduced computational time. Instance N4 has been addressed by References [8], [10] and [3]. These authors propose different ways of adapting the metaheuristics of the genetic algorithm to solve this instance. Below is a comparison of results for Instance N4:

TABLE V.
COMPARISON OF COMPUTATIONAL RESULTS OF INSTANCE N4

Approach Moon et al. (2004)	Approach Moon et al. (2006)	Approach Zhang et al. (2006)	Proposed mathematical model
1207	1102	1102	1089

In this table, it can be seen that the minimum value of the makespan is 1089. This value is the optimum since it is based on an exact method, opposite the approximate methods, such as the metaheuristics of the genetic algorithm, which obtain values close to the optimum.

IV. CONCLUSIONS

In this paper, a mixed-integer linear programming model is proposed to solve the APS problem in an MPC environment. The computational experiments demonstrated that the proposed mathematical model successfully solved the instances of the literature and in a reduced computational time. In this way, the proposed mathematical model can be used by various companies that have difficulties scheduling orders and fulfilling them.

As future work, it is intended to address a multi-objective problem that includes the minimization of the makespan, the balancing of the workload of the machines, the time of delay in the delivery of the orders, the anticipation time in the delivery of orders and transport time between machines.

REFERENCES

[1] M. Gen, M. L. Lin, H. Zhang, “Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey,” *Computers & Industrial Engineering*, vol. 56, pp. 779-808, 2009.

[2] M. Eck, *Advanced Planning and Scheduling: Is logistics everything? A research on the use (fullness) of advanced planning and scheduling systems*, Technique report, Vrije University, Amsterdam, Holland, 2003.

[3] H. Zhang, M. Gen, “Effective Genetic Approach for Optimizing Advanced and Scheduling in Flexible Manufacturing System,” *Proc. of GECCO*, pp. 1841-1848, 2006.

[4] C. Vercellis, “Multi-plant production planning in capacitated self-configuring two-stage serial systems,” *European Journal of Operational Research*, vol. 119, pp. 451-460, 1999.

[5] R. Kolisch, “Integration of assembly and fabrication for make-to-order production,” *International Journal of Production Economics*, vol. 68, pp. 287-306, 2000.

[6] K. A. Thoney, T. J. Hodgson, R. E. King, M. R. Taner, “Satisfying due-dates in large multi-factory supply chains,” *IIE Transactions*, vol. 34, pp. 803-811, 2002.

[7] C. Moon, J. Kim, S. Hur, “Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain,” *Computers & Industrial Engineering*, vol. 43, pp. 331-349, 2002.

[8] C. Moon, J. Kim, M. Gen, “Advanced planning and scheduling based on precedence and resource constraint for e-plant chains,” *International Journal of Production Research*, vol. 42, no. 15, pp. 2941-2955, 2004.

[9] C. Moon, Y. Seo, “Evolutionary algorithm for advanced process planning and scheduling in a multi-plant,” *Computers & Industrial Engineering*, vol. 48, pp. 311-325, 2005.

[10] Moon, Y. Seo, Y. Yun, M. Gen, “Adaptive genetic algorithm for advanced planning in manufacturing supply chain,” *Journal of Intelligent Manufacturing*, vol. 17, pp. 509-522, 2006.

[11] Y. H. Lee, C. S. Yeong, C. Moon, “Advanced planning and scheduling with outsourcing in manufacturing supply chain,” *Computers & Industrial engineering*, vol. 43, pp. 351-374, 2002.

[12] P. Yan, D. Liu, D. Yuan, J. Yu, “Genetic Algorithm with Local Search for Advanced Planning and Scheduling,” *Third International Conference on Natural Computation*, vol. 3, pp. 781-785, 2007.

[13] D. Liu, P. Yan, J. Yu, “Development of a multiobjective GA for advanced planning and scheduling problem,” *The International Journal of Advanced Manufacturing Technology*, vol. 42, pp. 974-992, 2009.

[14] J. Yang, W. Tang, “Preference-based Adaptive Genetic Algorithm for Multiobjective Advanced Planning and Scheduling Problem,” *IEEE International Conference*, pp. 1935-1940, 2009.

APPENDIX: EXPERIMENTAL DATA

Precedence relationship between order operations (r_{ij})

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-	1	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-
2	0	-	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-
3	0	0	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-
4	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	0	1	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	0	-	0	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	0	0	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	0	1	0	0	-	-	-	-	-
9	-	-	-	-	-	-	-	0	-	0	1	0	-	-	-	-	-
10	-	-	-	-	-	-	-	0	0	-	0	1	-	-	-	-	-
11	-	-	-	-	-	-	-	0	0	0	-	1	-	-	-	-	-
12	-	-	-	-	-	-	-	0	0	0	0	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1	0	0
14	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	1	0
15	-	-	-	-	-	-	-	-	-	-	-	-	0	0	-	0	1
16	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	-	1
17	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	-

Operation processing time i in machine m (p_{im})

	1	2	3	4	5	6
1	7	-	-	5	-	-
2	7	-	-	6	-	-
3	-	6	5	-	8	-
4	6	-	-	-	-	5
5	-	9	-	8	-	-
6	3	5	-	-	6	-
7	8	-	12	9	-	8
8	-	-	5	-	8	-
9	10	-	-	10	-	7
10	6	5	-	-	6	-
11	15	-	-	6	-	5
12	-	6	-	-	5	-
13	-	-	6	6	-	8
14	-	5	-	-	9	-
15	-	-	6	4	-	-
16	-	5	-	3	-	5
17	5	-	-	-	4	-

Order index of the operation i (k_i)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1	1	2	2	2	3	3	3	3	3	4	4	4	4	4

Batch size of the operation order i (q_i)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
40	40	40	40	70	70	70	60	60	60	60	60	30	30	30	30	30

Setup time between operation i and operation j (t_{ij}^{U})

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-	17	36	6	37	20	11	30	5	32	30	36	23	21	1	28	20
2	42	-	32	3	2	15	15	22	44	39	30	37	47	12	5	38	31
3	6	6	-	37	26	0	23	29	12	5	13	3	13	24	29	26	8
4	12	3	40	-	19	46	31	30	31	49	49	27	39	45	9	0	3
5	2	48	43	25	-	49	10	11	4	8	17	39	34	31	11	0	24
6	24	26	43	31	49	-	22	31	21	43	31	10	30	23	2	34	38
7	20	45	28	43	22	16	-	39	46	25	43	34	9	22	38	12	7
8	15	41	44	35	14	10	30	-	2	14	7	8	22	3	18	45	18
9	25	47	22	21	47	39	26	0	-	22	33	7	37	20	25	20	7
10	3	46	9	10	35	18	5	21	24	-	33	40	22	23	41	37	31
11	1	17	31	3	30	15	23	21	37	3	-	15	23	32	3	46	6
12	4	18	41	37	26	39	43	46	44	28	13	-	45	47	7	32	2
13	18	45	24	27	47	21	8	21	35	38	26	39	-	21	2	12	33
14	48	37	46	44	25	24	1	8	38	46	48	37	6	-	6	41	10
15	43	3	39	3	44	17	46	24	46	33	9	16	15	4	-	4	12
16	9	44	40	21	16	12	36	37	44	16	41	31	7	3	8	-	44
17	14	21	5	29	44	48	8	31	10	44	1	7	18	2	11	22	-

Transport time between the machine m and machine n (t_{mn}^{S})

	1	2	3	4	5	6
1	-	5	6	50	50	50
2	5	-	7	50	50	50
3	6	7	-	50	50	50
4	50	50	50	-	5	6
5	50	50	50	5	-	7
6	50	50	50	6	7	-

Machine capacity m (L_m)

1	2	3	4	5	6
1000	1000	2000	2000	2000	2000

Order loading operation i for operation j (u_{ij}) = 10

Plant index where the machine belongs m (b_m)

1	2	3	4	5	6
1	1	1	2	2	2