IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 22, Issue 4, Ser. 1 (July. – August. 2026), PP 05-09
www.iosrjournals.org

A Digraph-Based Optimization Algorithm For Maximum -Profit Scheduling

Wei-Xiang Huang, Min-Jen Jou
(Department Of Information Technology, Ling-Tung University, Taiwan)

Abstract:
Scheduling is a fundamental problem in manufacturing systems, where effective scheduling decisions directly influence production efficiency and overall profitability. This paper proposes a digraph-based optimization algorithm for maximum-profit scheduling. An illustrative example is provided to demonstrate the effectiveness of the proposed algorithm. The computational results indicate that the proposed method successfully identifies the maximum-profit path by efficiently updating the accumulated profit and predecessor information at each node.
Key Word: *Digraph; Maximum-Profit Scheduling; Optimization Algorithm*

Date of Submission: 02-07-2026 Date of Acceptance: 12-07-2026

I. Introduction

Scheduling plays an important role in manufacturing systems because it directly influences production efficiency, resource utilization, and overall profitability. As manufacturing environments become increasingly competitive, enterprises must determine appropriate production schedules to improve operational performance while satisfying production constraints. Consequently, developing efficient scheduling methods has become an important research topic in manufacturing and operations management. In many practical manufacturing environments, scheduling decisions are concerned not only with minimizing the completion time or makespan but also with maximizing overall profit. Since different jobs contribute different levels of profit, determining an appropriate scheduling sequence to achieve the maximum economic benefit has become an important issue in scheduling research. Therefore, this study constructs a scheduling model based on directed graphs (digraphs) and employs a path-searching mechanism to identify the maximum-profit path, which serves as an important basis for scheduling decision-making.

This study adopts several scheduling approaches, including mathematical programming, graph theory, and algorithmic methods. Among these approaches, graph theory has been widely applied to scheduling problems because it effectively represents the precedence relationships among jobs and facilitates the analysis of feasible scheduling sequences. In particular, directed graphs (digraphs) provide a clear representation of the dependency relationships among tasks and enable systematic path-searching mechanisms for analyzing and solving scheduling problems. Therefore, this study employs these methods, with directed graphs serving as the primary modeling tool for scheduling.

II. Research Method

A directed graph consists of vertices and directed edges and can be used for path planning, task scheduling, and production processes. Dynamic programming is widely used in mathematics, management science, computer science, economics, and bioinformatics to solve optimization problems. For shortest path problems, dynamic programming or greedy algorithms are commonly used. We represent the set of orders as a directed graph and apply dynamic programming to determine the order acceptance sequence that yields the maximum profit. The following Algorithm 1 is applied to determine the order acceptance sequence and the maximum profit.

Algorithm 1. Digraph-Based Optimization Algorithm for Maximum Profit Scheduling
Input:
- A set of orders $V = \{v_1, v_2, \ldots, v_n\}$.
- Start date and finish date of each order.
- Profit ω of each order.

Output:
- Schedule I, Schedule II, Schedule III.

- Corresponding maximum-profits α, β, and γ

Step 1. Construct the Digraph

Construct a digraph D1=(V,E), where each vertex represents an order. A directed arc is established from order X to order Y if

- the finish date of X is earlier than the start date of Y, and
- no other feasible order can be inserted between them.

Step 2. Node Labels

For every node, assign the triplet (a,Z,ω), where

- a : maximum accumulated profit,
- Z : predecessor node producing the maximum accumulated profit,
- ω : profit of the current order.

Step 3. Compute the Maximum-Profit Path

Traverse the digraph according to the precedence relationships. For every node,

- update the accumulated profit;
- record the predecessor producing the maximum accumulated profit.

After all nodes have been processed,

- backtrack from the terminal node,
- obtain **Schedule I**,
- record the maximum accumulated profit α .

Step 4. Construct a Reduced Digraph

Eliminate all nodes in Schedule I to obtain a reduced digraph D2 composed of the remaining nodes.

Step 5. Compute the Second Maximum-Profit Path

Repeat Steps 2–4 on the reduced digraph D2. Obtain

- **Schedule II**
- maximum accumulated-profit β .

Step 6. Construct Another Reduced Digraph

By eliminating all nodes included in Schedule II from D2, a reduced digraph D3 containing the remaining nodes is constructed.

Step 7. Compute the Third Maximum-Profit Path

Repeat Steps 2–4. Obtain

- **Schedule III**
- maximum accumulated profit γ .

Final Output

Return:

- Schedule I, Schedule II, and Schedule III.
- The corresponding maximum-profits α, β, and γ.

The three schedules are assigned according to their profit ranking

III. Simulation Process

We implement the proposed algorithm to obtain the optimal scheduling results. Table 1 lists the order dates.

Table 1. The Dates of Orders

Job ID	Start Date	Finish Date	Duration (Days)	Job ID	Start Date	Finish Date	Duration (Days)
A	Jan. 2	Jan. 5	4	N	Feb. 3	Feb. 10	8
B	Jan. 1	Jan. 6	6	O	Jan. 31	Feb. 4	5
C	Jan. 5	Jan. 7	3	P	Feb. 6	Feb. 14	9
D	Jan. 5	Jan. 10	6	Q	Feb. 15	Feb. 21	7
E	Jan. 13	Jan. 18	6	R	Feb. 8	Feb. 12	5
F	Jan. 9	Jan. 13	5	S	Feb. 10	Feb. 16	7
G	Jan. 10	Jan. 13	4	T	Feb. 11	Feb. 18	8
I	Jan. 10	Jan. 12	3	U	Feb. 5	Feb. 8	4
J	Jan. 19	Jan. 21	3	V	Feb. 14	Feb. 17	4
H	Jan. 20	Jan. 24	5	W	Feb. 17	Feb. 20	4
K	Jan. 27	Feb. 1	6	X	Feb. 18	Feb. 25	8
L	Jan. 27	Feb. 3	8	Y	Feb. 6	Feb. 13	8
M	Feb. 1	Feb. 7	7	Z	Feb. 21	Feb. 28	8

Note: Duration (Days) = Number of days from Start Date to Finish Date (inclusive).

Step 1: Construct a digraph D_1=(V,E), where each vertex represents an order and each directed edge represents a feasible precedence relationship between two orders.

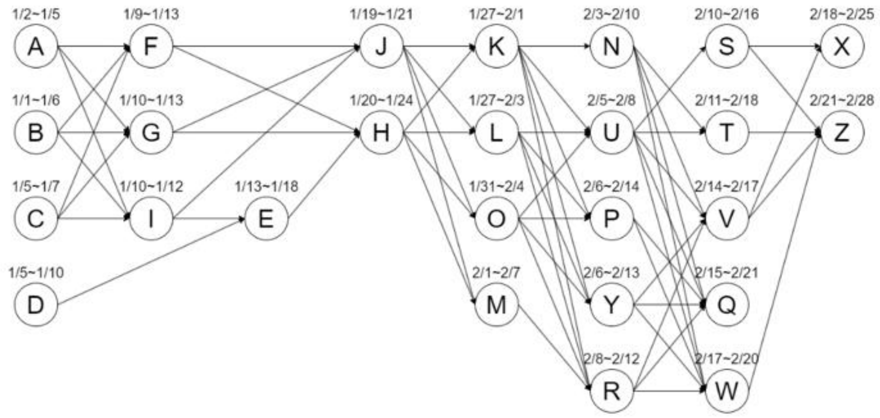

Figure 1. The digraph D1

Step 2: Calculate values (a,Z,ω) for each node in D1 ,where a denotes the maximum accumulated profit, Z denotes the predecessor node, and ω denotes the profit of the current order.

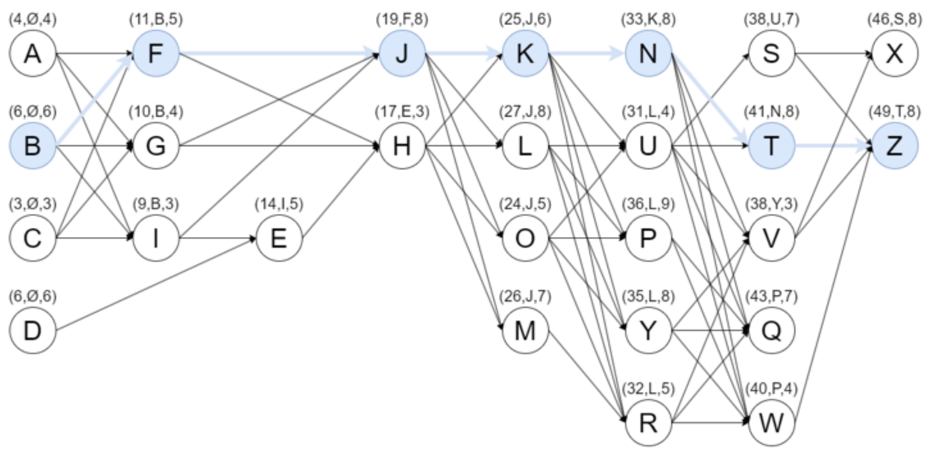

Figure 2. Calculate (a,Z,ω)

Step 3: From Step 2, obtain the optimal schedule P_1 and maximum-profit α in D1.

P1: B,F,J,K,N,T,Z.
$\alpha = 49$

Step 4: Remove the vertices of P_1 from D1 to generate a new directed graph D2 = (V2, E2).

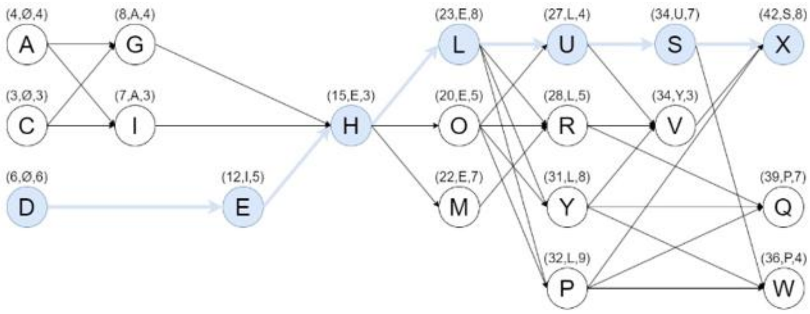

Figure 3. The digraph D2

Step 5: Repeat Steps 2 and 3 on D2 to obtain the second maximum-profit schedule P_2 and maximum-profit β in D2.

P2:D,E,H,L,U,S,X.
$\beta = 42$

Step 6: Remove P_2 from D2 to form D3 = (V3, E3).

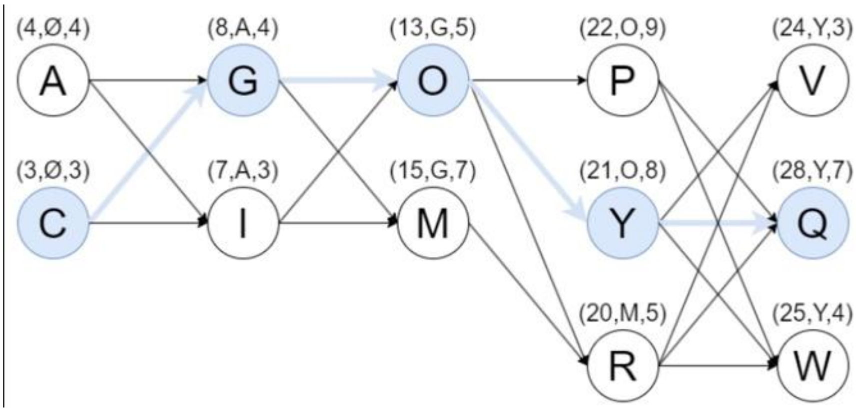

Figure 4. The digraph D3

Step 7: Repeat Steps 2 and 3 again on D_2 to obtain the third-largest maximum-profit schedule P_3 and maximum-profit γ in D3.

P3:C,G,O,Y,Q.
$\gamma = 28$

IV. Conclusion

This paper proposed a digraph-based optimization algorithm for maximum-profit scheduling. By representing scheduling relationships as a directed graph and applying dynamic programming, the proposed method efficiently determines the schedule that yields the maximum accumulated profit. The algorithm records the accumulated profit and predecessor information at each node, enabling the optimal scheduling sequence to be obtained through a backtracking process. To identify multiple feasible schedules, reduced digraphs are constructed by iteratively removing the nodes included in the previously obtained schedule. This procedure enables the algorithm to generate the first, second, and third maximum-profit schedules while maintaining computational efficiency. The simulation results demonstrate that the proposed method can effectively identify high-profit scheduling sequences and provide corresponding profit values for decision-making. The proposed algorithm is simple to implement and can be applied to various scheduling problems involving precedence constraints, such as manufacturing systems, production planning, and task scheduling. Future research may extend the proposed method by considering additional practical constraints, including limited resources, multiple machines, processing times, and uncertainty, to further improve its applicability to real-world scheduling environments.

References

[1]. Baker, K. R., & Trietsch, D. Principles Of Sequencing And Scheduling. John Wiley & Sons, 2009.
[2]. Pinedo, M. L. Scheduling: Theory, Algorithms, And Systems, 6th Ed. Springer, 2022.
[3]. Bondy, J. A., & Murty, U. S. R. Graph Theory. Springer, 2008.