

Research Article

An Integer Linear Programming Model for Partially Ordered Sets

Elsayed Badr ,¹ I.M. Selim,² Hoda Mostafa,³ and Hala Attiya⁴

¹Scientific Computing Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

²Computing Science Department, Faculty of Computers and Artificial Intelligence, University Sadat City, Sadat City, Egypt

³Mathematics and Computer Science Department, Faculty of Science Menoufia University, Shibin Al Kawm, Egypt

⁴Basic Science Department, Faculty of Technology and Education, Beni-Suef University, Beni Suef, Egypt

Correspondence should be addressed to Elsayed Badr; badrgraph@gmail.com

Received 26 April 2022; Revised 30 July 2022; Accepted 23 August 2022; Published 10 September 2022

Academic Editor: Mehar Ali Malik

Copyright © 2022 Elsayed Badr et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Linear programming is an important approach that is used to represent a large class of combinatorial optimization problems. The simplex algorithm is one of the algorithms for solving linear programming problems with exponential time complexity. Fortunately, this algorithm solves real-world problems with polynomial time complexity. In particular, a new Integer Linear Programming model (ILPM) is proposed for partially ordered sets. Robert Dilworth's Decomposition theorem is formulated by ILPM and proves its correctness using the paradigm of strong duality in linear programming. Finally, ILPM is run on fifteen benchmark partially ordered sets for finding their width. The computational experiments show the validity of the proposed model.

1. Introduction

Linear programming is a very rich problem-solving model which is used to represent a large class of optimization problems. Optimization problems are defined as problems whose solutions represent maximum values (such as maximum profit) or minimum values (such as minimum cost). There are various types of optimization problems such as constraint optimization problems and unconstrained optimization problems. On the other hand, there are many types according to the nature of the variables of the optimization problem, such as linear programming, integer linear programming, and nonlinear programming. There are various algorithms to solve linear programming problems, the most famous of which are the simplex algorithm and the interior point algorithm. The simplex algorithm is one of the ten best mathematics algorithms of the twentieth century [1]. By a common numerical example, Klee and Minty [2] proved that the simplex algorithm is not a polynomial time algorithm. Fortunately, this algorithm solves real-world problems of polynomial time complexity.

Casting optimization problems into a linear program version is an art. The tools of this art are represented in the

complete knowledge of the issue of optimization to be represented by a linear program. The second of these tools is the complete knowledge of the general form of the linear program (objective function–constraints–the right-hand side of the constraints (constants)–non-negative–etc). The third of these tools is the practice of formulating a set of optimization problems into a linear program version. For graph labeling problems, Badr et al. [3–5] introduced new integer linear programming models (ILPMs) to find the upper bounds of the radio number, a radio mean number, and radio mean square number for a given graph, respectively. The proposed ILPMs outperformed other algorithms according to the running time. For the metric dimension of a given graph, Chartrand et al [6] formulated the optimization problem of determining the metric dimension of a given graph (G). For the traveling salesman problem, there are different mathematical models [7–9] for the traveling salesman problems.

The problem of efficient tasks scheduling is ubiquitous in many areas of computer science and engineering. We are often given a collection of tasks to complete, in which some jobs can be done in parallel, and others must be done in a predefined sequence. In general, we can formulate these

problems by constructing a poset of objects (representing tasks) in which some elements are “comparable” (jobs that must be done in predefined sequence), and others are “incomparable” (jobs that can be done in parallel). If our goal is to devise an efficient task scheduling algorithm, we might be interested in finding the largest subset of tasks that can be completed in parallel (the maximum antichain problem). Or we may be interested in the related problem of distributing all the tasks between the fewest number of subgroups (e.g., for parallel processing) such that all jobs in a subgroup must be completed in a predefined sequence (the minimum chain covering problem).

For more details about the history of the decomposition of the partially ordered sets posets, the reader can refer to [10–16]. On the other hand, for more details about how to formulate a problem as a mathematical model the reader is referred to [17–19]. Finally, for more details about the well know optimal PMU placement the reader is referred to [20–22].

In this paper, a new integer linear programming model (ILPM) is proposed for posets. Robert Dilworth’s Decomposition theorem is formulated by ILPM and proves its correctness using the paradigm of strong duality in linear programming. Finally, ILPM is run on fifteen benchmark partially ordered sets for finding their width. The computational experiments show the validity of the proposed model.

The remaining of this paper is organized as follows: basic definitions and notations are introduced in Section 2. In Section 3, Dilworth theorem and its mathematical formulation are proposed. In Section 4, the experimental results analysis of the proposed ILPM on fifteen benchmark posets are provided. In Section 5, conclusions are made.

2. Preliminaries

Before presenting Dilworth’s Theorem, we need to have a notion of comparability between elements in a set. To formalize this notion, we define some properties of binary relations to represent a partially ordered set (poset). We then illustrate how subsets of elements in a poset can have chain or antichain properties.

2.1. Binary Relations. Elements of a set can have relational properties such as transitivity or antisymmetry.

Definition 1. A binary relation, $<$, on a set P is transitive if and only if $(a < b \text{ and } b < c) \text{ implies } a < c$ for all $a, b, c \in P$

Definition 2. A binary relation $<$, on a set P is anti-symmetric if and only if $(a \neq b \text{ and } a < b) \text{ implies } b < a$ for all $a, b \in P$.

We use binary relations with these specific properties to represent a partial order on a set of elements, in which some pairs of elements are comparable by $a_i < a_j$. In context of a task scheduling problem, this means that job a_i must be completed before.

Definition 3. A *partially-ordered set*, or *poset*, is a set of elements P , along with a binary relation $<$, having the property that $<$ is transitive and antisymmetric for all elements in P .

Note that, two elements a_i and a_j can also be *incomparable*, that is, when we have both $a_i \not< a_j$ and $a_j \not< a_i$. In this case, tasks a_i and a_j may be completed in parallel.

We also define maximal and minimal elements in a poset:

An element a_i is maximal in a subset $S \subseteq P$ if it is not smaller than, any other element in S , i.e. $a_i \not< a_j$ for all $a_j \in S$. Similarly, an element a_i is minimal S if it is not greater than any other element i.e. $a_j \not< a_i$ for all $a_i \in S$.

Definition 4. A subset of elements $= \{a_1, \dots, a_k\} \subseteq P$ forms an *antichain* if and only if $a_i \not< a_j$ for all $i \neq j$. That is all elements in \mathcal{A} are pairwise incomparable.

Definition 5. A subset of elements $C = \{a_1, \dots, a_k\} \subseteq \mathcal{P}$ forms a *chain* if and only if they satisfy for all $a_1 < a_2 < \dots < a_k$ for some ordering of elements in C . That is all elements in C are pairwise comparable.

For example, in a task scheduling problem, if a subset of jobs forms an antichain. Then, all these jobs may be completed in parallel. If a subset of jobs forms a chain, then these jobs must be completed in the sequence determined by $a_1 < a_2 < \dots < a_k$. Furthermore, it is useful to describe a covering of a poset P by chains to be a collection of chains $\{C_1, \dots, C_k\}$ such that every element of \mathcal{P} is contained in the union of these chains, that is, $\bigcup C_i = P$. This is related to distributing all tasks on parallel processors $\{C_1, C_2, \dots\}$ so that jobs assigned to any processor C_i are to be completed in a predefined sequence.

3. Dilworth’s Theorem

With these definitions in hand, we can now formulate Dilworth’s theorem [10] on the size of maximum antichains and minimum chain coverings in a poset.

Theorem 1 (Dilworth’s Theorem). *ie size of the largest antichain in any poset P is equal to the size of the minimum coverage of P by chains.*

We can construct a proof of Dilworth’s theorem using the pigeonhole principle and Duality in linear programming. If we let:

$A = \max \{|\mathcal{A}|: \mathcal{A} \text{ is antichain of } \mathcal{P}\}$ (size of largest antichain in P), and $B = \min k: \bigcup \{C_1, \dots, C_k\} = P$ (minimum number of chains needed to cover P), then one way to prove Dilworth’s theorem is to show $A \leq B$ and $B \leq A$.

To prove $A \leq B$, we use the pigeonhole principle.

Lemma 1. *The size of any chain in a poset P is equal to at most the minimum number of chains needed to cover P , i.e. $A \leq B$.*

Proof. As above-mentioned, let B be the minimum number of chains C_1, \dots, C_B needed to cover P . Assume, for the sake of contradiction, that an antichain $\mathcal{A} = a_1, a_2, \dots, a_n$ in

\mathcal{P} has size $n > B$. Then, by the pigeonhole principle, at least two distinct elements a_i, a_j must be members of the same chain so that, by Definition 5, we have $a_i a_j$ or $a_j a_i$. However, by Definition 4, this means a_i and a_j are not in the same chain, a contradiction. Thus, no antichain in P can have size greater than B , whereby it follows the maximum size antichain in P is also no larger B . \square

3.1. Casting as a Linear Program. To prove $B \leq A$ is more difficult. We can begin by trying to cast the objective of finding the maximum size antichain of a poset as a linear program.

3.1.1. Primal Form. Assign non-negative weights $\vec{x} = (x_1, \dots, x_n)$ respectively to each element $a_1, \dots, a_n \in P$. Each chain $C_k \subset P$ will have an associated constrain inequality over the elements $a_i \in C_k$ that comprise it. Consider the following linear program in canonical form:

$$\begin{aligned} \max \quad & z = \sum_{i=1}^n x_i, \\ \text{s.t} \quad & \sum_{i: a_i \in C_k} x_i \leq 1 \quad \text{for all chains } C_k \subseteq P \cdot \\ & x_i \geq 0 \quad \text{for } i = 1, \dots, n. \end{aligned} \tag{1}$$

3.1.2. Dual Form. In addition, we can convert (1) to its dual form, where variables represent chains on P , and each element $a_i \in P$ has an associated constrain inequality over the collection of chains that contain it (note the symmetry). Assign non-negative weights.

$\vec{y} = (y_1, \dots, y_k)$, respectively, to each chains $C_1, \dots, C_k \subseteq P$. The dual of (1) (henceforth denoted (2)) is

$$\begin{aligned} \text{Minimize } w = & \sum_{j=1}^k y_j, \\ \text{Subject to : } & \sum_{j: a_i \in C_j} y_j \geq 1 \quad \text{for all elements } a_i \in P \\ & y_j \geq 0 \quad \text{for } j = 1, \dots, k. \end{aligned} \tag{2}$$

We begin by showing that if the primal (1) and (2) have integer solutions, then the objective z in (1) gives the size of the maximum antichain in P , and the objective w in (2) gives the size of the minimum coverage of P by chains.

Proposition 1. *If $x_i \in \mathbb{Z}$ for all weights in \vec{x} , then solving primal (1) gives the maximum antichain in P .*

Proof. In the case that all x_i are non-negative integers, we wish to show that the objective of (1), $z = \sum_{i=1}^n x_i$, is equal to $|\mathcal{A}_{\max}|$, where \mathcal{A}_{\max} is any maximum antichain in P .

Observe that constrains in (1), $\sum_{i: a_i \in C_k} x_i \leq 1$ for all chains $C_k \subseteq P$ impose the following requirement on the variable x_i if we assume they are all non-negative integers: \square

Claim 1. For all chains $C_k \subseteq P$, at most one of the weights associated with elements in C_k is equal to 1. All other weights must be 0.

Proof of Claim 1. For the sake of contradiction, assume that more than one element in chain C has a positive weight. We would then have $\sum_{i: a_i \in C} x_i > 1$ since all x_i are non-negative integers, thereby violating constrains of the linear program.

It follows that $x_i \in \{0, 1\}$ for all i . Now consider each x_i as an indicator variable for whether element a_i appears in an antichain \mathcal{A}_{\max} , i.e. let $x_i = 1$ if $a_i \in \mathcal{A}_{\max}$ and $x_i = 0$ otherwise.

Since at most one indicator variable in any chain is positive, the solution to (1) satisfies the requirement that no two elements chosen for the maximum antichain come from the same chain.

Furthermore, under the indicator variable interpretation, the objective of maximizing $z = \sum x_i$ corresponds to choosing the maximum number of elements to include in an antichain of P , so we have $Maxz = |\mathcal{A}_{\max}|$, as desired. This proves Proposition 1 that an integer solution to (1) gives the size of the maximum antichain in P .

Now, we prove the analogous proposition for the dual form. \square

Proposition 2. *If $y_j \in \mathbb{Z}$ for all weights in \vec{y} then solving dual (2) gives the size of the minimum covering of P by chains.*

Proof. The reasoning is analogous to that of Proposition 1.

The constrains, in (2), $\sum_{j: a_i \in C_j} y_j \geq 1$ for all elements $a_i \in P$, impose the following requirement on variables y_j if we assume they are all non-negative integers: \square

Claim 2. For all $a_i \in P$, at least one of weights associated with the set of chains containing a_i is positive.

Proof of Claim 2. It follows immediately from the constrains. If, for some element a_i , none of the weights associated with chains that contain a_i are positive, then $\sum_{j: a_i \in C_j} y_j = 0 \not\geq 1$, thereby violating the constrain of the linear program.

Furthermore, since the linear program seeks to minimize $\sum y_j$ with all y_j as integers, we must have that $y_j = 1$ whenever y_j is chosen positive. Thus, with $y_j \in \{0, 1\}$ for all j , we can again consider each y_j as an indicator variable for whether chain C_j in a minimum covering of P .

Note again that for each element $a_i \in P$, at least one indicator variable is associated with a chain containing a_i equal 1, so the solution to (2) satisfies the requirement that all elements $a \in P$ appear in the minimum chain covering.

Furthermore, under the indicator variable interpretation, the objective of minimizing $w = \sum y_j$ corresponds to choosing the minimum number of chains to include in a covering of P , so we have $Minw = \min\{c: C_1, \dots, C_c \text{ covers } P\}$, as desired. This proves Proposition 2 that an integer solution to (2) gives the minimum covering of P by chains.

Now, we need to show that both the primal and dual linear programs indeed have integer solutions. \square

Lemma 2. For any distribution of chains in a poset, (1) has an optimal solution where all variables take on integer values.

Proof. Proof is by the Well-ordering Principle and contradiction of an optimal solution to (1) with the fewest number of noninteger variables (x_i weights). Let N be the set of elements in P corresponding to these noninteger weights. Some of these elements will be maximal in N , and others will be minimal (recall definitions in Section 2). We first need to verify that. \square

Claim 3. No elements in N can be both minimal and maximal.

Proof of Claim 3. Suppose, for the sake of contradiction, that an element $a_m \in N$ is both minimal and maximal. Then, a_m is incomparable to all other elements in N (recall definitions in Section 2). It follows that is associated weight x_m is subject only to a single constrain that $x_m \leq 1$ in (1), whereby x_m would be assigned a value of 1 to maximize the objective. This contradicts the assumption that $x_m \in N$ is a not an integer, thus proving the claim.

Since no elements in N are both minimal and maximal, it follows that each chain $C \in N$ has exactly one maximal element, and a different minimal element.

Suppose we add a small value $\epsilon > 0$ to each maximal element in N , and subtract ϵ from each minimal in N . Since all elements in N are nonintegers we ensure that the resulting assignment of variables is feasible by choosing $\epsilon > 0$ small enough so that $0 \leq x_i \leq 1$ is still satisfied by all x_i , as required by the constraints of (1).

If we keep repeating this procedure, then at least one variable corresponding to a maximal/minimal element in N will reach a boundary value of 0 or 1. This contradicts the assumption that N has the fewest number of noninteger variables in a feasible solution. Thus, all variable assignments in an optimal solution to (1) must be integers, so Lemma 2 holds for primal (1). Next, we need to show that dual (2) also has integer solutions. \square

Lemma 3. For any distribution of chains in a poset, (2) has an optimal solution where all variables take on integer values.

Proof. First, we apply the result of strong duality in linear programming (see [19] for details). Which says that the primal and dual forms of a linear program have the same optimal value. We prove using induction that if $k \in \mathbb{Z}$ is the optimal value to LP2 on a poset P , then P has a covering by k chains where all variables \vec{y} in (2) take on integer values.

Inductive hypothesis: for any poset P having an integer optimal (2) objective value of k , the optimal solutions to the LPs associated with strict subsets of P are integers.

Base case. Consider a post P_1 in which all elements form a single, with optimal objective value $\text{Min } w = k = 1$. From the constraints of (2), this optimal solution will have a single variable y_1 (corresponding to the chain containing all elements in P_1) take on the value of 1, and the other y 's take on

the value of 0. It follows that optimal solutions to each LP associated with subsets of P_1 will also have integer weights. Thus, the inductive hypothesis holds for $k = 1$.

Now, suppose that an optimal solution for (2) on poset P has a variable assignment $y_i = a$ that is not an integer (note that the optimal value for this LP must be an integer $\text{Min } w = k \in \mathbb{Z}$). We can remove from P all elements in the chain C_i associated with variable y_i to obtain a new poset $P/C = P' \subset P$.

Let w' be the objective function value of LP2' for this a new poset P' . We observe that $\text{Min } w'$ is at most $k - a$ because all elements comprising the chain with weight $y_i = a$ were removed so that $w' \leq w - a = k - a$. Since $P' \subset P$ is a strict subset of P , $k - a$ is an integer by the inductive hypothesis, so $\text{Min } w' = k - 1$. Furthermore, the inductive hypothesis tells us that P' has a covering by $k - 1$ chains where all weights assignments to the chains are integers. It follows that we get a covering of $P = P' \cup C_i$ by k chains if we simply assign weight of $y_i = 1$ to C_i , achieving the desired objective the desired value of $\text{Min } w = \text{Min } w' + y_i = (k - 1) + 1 = k$. This completes the inductive argument and proof of Lemma 3.

Taking Lemma 2 and 3 in conjunction with Proposition 1 and 2 gives the desired result that, for all partially-ordered sets P :

Solving (1) gives the size of the largest antichain in P

Solving (2) gives the size of the smallest covering of P by chains

Again, by applying the result of strong duality [19] in linear programming, we conclude that primal (1) and dual (2) have the same optimal value. Therefore, (1) s solution for the size of the largest antichain in a poset and (2) solution for the size of the minimum chain covering of a poset are equal, proving Theorem 1 (Dilworth's Theorem). \square

3.2. A Polynomial-Size Linear Program. In this section, we introduce Proposition 3 on which the proposed integer linear programming model is based.

Proposition 3. An assignment of indicator values $\{0, 1\}$ to $\vec{x} = (x_1, \dots, x_n)$ produces an antichain if and only if $x_i + x_j \leq 1$ for all edges $(a_i, a_j) \in E(G)$.

Proof. First, we show that any antichain in P satisfies the constraints. An antichain cannot contain any two elements that are comparable (Definition 4). Note that, any two comparable elements in P are connected by an edge in G . Thus, we may choose at most one vertex from each edge $(a_i, a_j) \in E(G)$, meaning that at most one of x_i, x_j equals 1, from which it follows that $x_i + x_j \leq 1$ for all edges $(a_i, a_j) \in E(G)$.

Conversely, we show that any assignment of indicator variables satisfying the constraints produces a valid antichain.

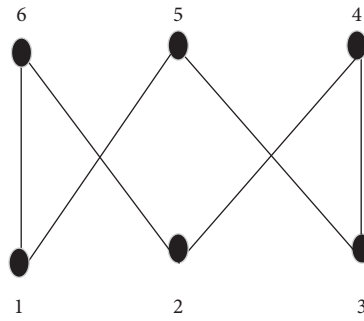


FIGURE 1: A poset with 6 elements and its width equal to three.

<p>$(P_1; W(P_1) = 2; S(P_1) = 1)$</p>	<p>$(P_2; W(P_2) = 2; S(P_2) = 3)$</p>	<p>$(P_3; W(P_3) = 2; S(P_3) = 2)$</p>
<p>$(P_4; W(P_4) = 3; S(P_4) = 3)$</p>	<p>$(P_5; W(P_5) = 3; S(P_5) = 3)$</p>	<p>$(P_6; W(P_6) = 3; S(P_6) = 3)$</p>
<p>$(P_7; W(P_7) = 3; S(P_7) = 3)$</p>	<p>$(P_8; W(P_8) = 3; S(P_8) = 4)$</p>	<p>$(P_9; W(P_9) = 3; S(P_9) = 3)$</p>
<p>$(P_{10}; W(P_{10}) = 4; S(P_{10}) = 4)$</p>	<p>$(P_{11}; W(P_{11}) = 4; S(P_{11}) = 4)$</p>	<p>$(P_{12}; W(P_{12}) = 4; S(P_{12}) = 4)$</p>
<p>$(P_{13}; W(P_{13}) = 4; S(P_{13}) = 4)$</p>	<p>$(P_{14}; W(P_{14}) = 4; S(P_{14}) = 4)$</p>	<p>$(P_{15}; W(P_{15}) = 4; S(P_{15}) = 4)$</p>

FIGURE 2: 15-Benchmark partially ordered sets with their width and jump number.

TABLE 1: A comparison between the standard width and directed acyclic graph (DAG).

Poset	n	Standard $w(P)$	Proposed model MATLAB	
			$w(P)$	CPU time
P1	4	2	2	0.073318
P2	6	2	2	0.089859
P3	6	2	2	0.101679
P4	6	3	3	0.102334
P5	6	3	3	0.104611
P6	5	3	3	0.108036
P7	7	3	3	0.123854
P8	8	3	3	0.146553
P9	5	3	3	0.166726
P10	8	4	4	0.207229
P11	8	4	4	0.328825
P12	7	4	4	0.471679
P13	7	4	4	0.47268
P14	7	4	4	0.546337
P15	7	4	4	0.649021

If $x_i + x_j \leq 1$ for all edges $(a_i, a_j) \in E(G)$, then most one indicator variable x_i, x_j is positive because $x_i \in \{0, 1\}$. Hence, at most one vertex (element in P) on each edge is chosen to be in the antichain. Since all pairwise comparable elements in P are connected by edges in our construction of $E(G)$, it follows that no two elements chosen in the antichain will be comparable, satisfying Definition 4. This proves Proposition 3.6. \square

3.2.1. Formulation of the Problem as a Mathematical Model. For any poset P with element set $\{a_1, \dots, a_n\}$, we can construct an $n \times n$ comparable incidence matrix A , having entry $A(i, j) = 1$ if elements $a_i, a_j \in P$ are comparable and $A(i, j) = 0$ otherwise. Now, associated with each element $a_i \in P$ an indicator variable $x_i \in \{0, 1\}$ that corresponds to using or not using that element in an antichain of P .

Now, we can formulate the problem of the decomposed poset P as follows:

$$\max z = \sum_{i=1}^n x_i, \tag{3}$$

Subject to: $A^* x \leq b \quad : x_1, x_2, \dots, x_n \in \{0, 1\}$,

where $A^* \in \mathfrak{R}^{m \times n}$ is the $m \times n$ matrix which is generated from the matrix A (based on Proposition 3) such that m is the number of ones in the matrix A , $x \in \mathfrak{R}^n$ and $b \in \mathfrak{R}^m$ (all elements in vector b equal to 1).

Example 1. We show how to formulate the problem of decomposed partially ordered set (as shown in Figure 1) as an integer linear programming model.

The $n \times n$ comparable incidence matrix A that is corresponding to the post P .

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

The $m \times n$ matrix $A^* \in \mathfrak{R}^{m \times n}$ is generated from matrix A :

$$A^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{5}$$

Each row in the matrix $A^* \in \mathfrak{R}^{m \times n}$ represents an inequality based on Proposition 3.7.

Finally, the complete integer linear programming model is as the following:

$$\max z = \sum_{i=1}^6 x_i,$$

$$\begin{aligned} x_1 & 0 & 0 & 0 & x_5 & 0 \leq 1 \\ x_1 & 0 & 0 & 0 & 0 & x_6 \leq 1 \\ 0 & x_2 & 0 & x_4 & 0 & 0 \leq 1 \\ \text{Subject to: } & 0 & x_2 & 0 & 0 & x_5 & 0 \leq 1 \\ & 0 & 0 & x_3 & x_4 & 0 & 0 \leq 1 \\ & 0 & 0 & x_3 & 0 & 0 & 0 \leq 1 \\ & x_1, x_2, x_3, x_4, x_5, x_6 & \in & \{0, 1\}. \end{aligned} \tag{6}$$

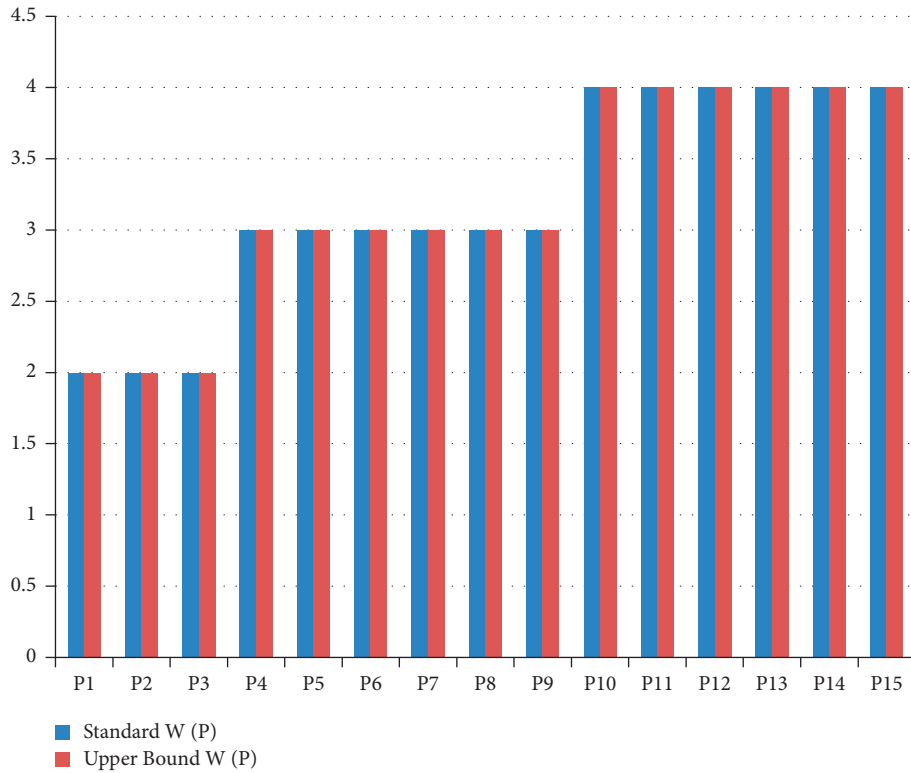


FIGURE 3: Shows that the relation between posets and standard width, (DAG).

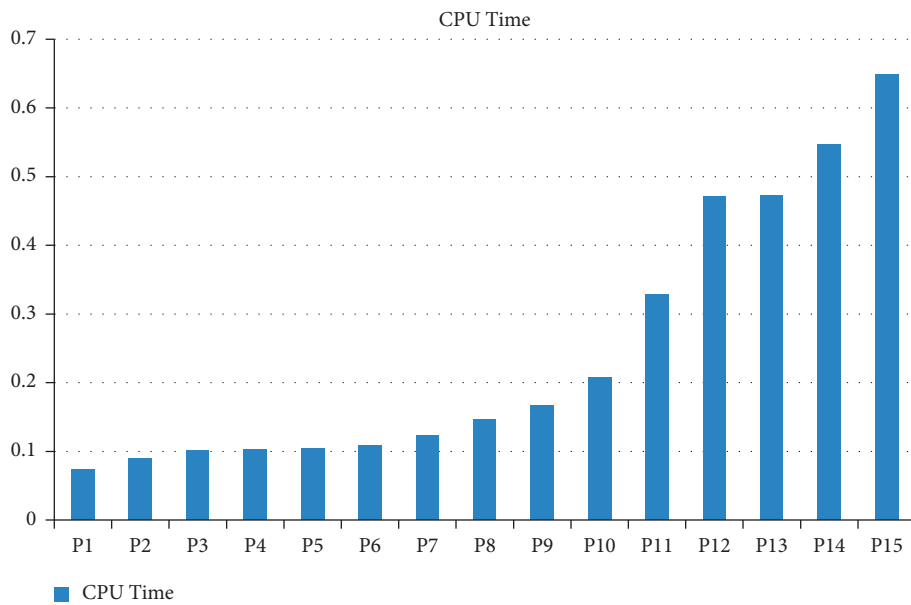


FIGURE 4: Shows that the relation between posets and CPU Time.

We can use an available software package such as (MATLAB, Python, and LINGO) for solving the above mathematical model which returns the solution is equal to three.

4. Computational Experiments

In this section, we evaluate the proposed ILPM by fifteen benchmark posets. These posets are characterized by some characteristics such as (various widths $w(P)$, various sizes

(n) , and various heights $h(P)$) Figure 2 contains fifteen benchmark posets with their width and jump number.

The used environment is a 64-bit Windows 8.1 Operating System with Core (TM) i5 CPU M 460 @2.53 GHz, 4.00 GB of memory, and MATLAB v7.01 Software Package.

From Table 1 and Figure 3, it is obvious that the results of the proposed mathematical model (ILPM) coincide with the standard results for all fifteen benchmark posets.

It is also evident from Table 1 and Figure 4 that the execution time of ILPM increases as the width of the poset increases. That is, the running time of ILPM is directly proportional to the width of the post. Fortunately, the behavior of the simplex algorithm used to solve ILPM is a polynomial time to solve real-world applications, regardless of being exponential time as proven in [2]. For this reason, we have done our best to create a poset in the form of the examples provided by Klee and Minty in 1972 in which they proved that the simplex algorithm is exponential time, but we have failed to do.

Because the examples presented by Klee and Minty depend on (right-hand side constants, the coefficients of the constraint matrix, and the coefficients of the objective function), rarely, do these problems exist as real-world problems. We conclude from this that the presented model will be implemented in a polynomial time. It is well-known that the complexity of the simplex algorithm is $O(n^2)$ for real-world applications in the average case where n is the number of elements of a poset.

5. Conclusion

A new integer linear programming model (ILPM) has been proposed for posets. Robert Dilworth's Decomposition theorem is formulated by ILPM and proved its correctness using the paradigm of strong duality in linear programming. Finally, ILPM is run on fifteen benchmark partially ordered sets for finding their width. The computational experiments showed the validity of the proposed model. On the other hand, CPU Time has been calculated for fifteen benchmark partially ordered sets. From computational experiments, it was concluded that the CPU time increases as the width increases. In future work, the proposed ILPM will be evaluated by using special posets such as interval posets, lattice, crown posets, and k -tower poset. ILPM will be applied on the jump-critical ordered sets [23].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are thankful to Runpeng Liu for his suggestions that make this manuscript better.

References

- [1] A. Barry, "Cipra "the best of the 20th century: editors name top 10 algorithms" from," *SIAM News*, vol. 33, 2000.
- [2] V. Klee and G. J. Minty, "How good is the simplex algorithm?" *Inequalities III (Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, Calif., September 1-9, 1969, Dedicated to the Memory of Theodore S. Motzkin)*, pp. 159-175, Academic Press, Cambridge, MA, USA, 1972.
- [3] E. Badr, S. Almotairi, A. Eirokh, A. Abdel-Hay, and B. Almutairi, "An integer linear programming model for solving radio mean labeling problem," *IEEE Access*, vol. 8, pp. 162343-162349, 2020.
- [4] E. M. Badr and M. I. Moussa, "An upper bound of radio k -coloring problem and its integer linear programming model," *Wireless Networks*, vol. 26, no. 7, pp. 4955-4964, 2020.
- [5] E. Badr, S. Nada, M. M. Ali Al-Shamiri, A. Abdel-Hay, and A. ELrokh, "A novel mathematical model for radio mean square labeling problem," *Journal of Mathematics*, vol. 2022, Article ID 3303433, 9 pages, 2022.
- [6] G. Chartrand, L. Eroh, M. A. Johnson, and O. R. Oellermann, "Resolvability in graphs and the metric dimension of a graph," *Discrete Applied Mathematics*, vol. 105, pp. 99-113, 2000.
- [7] Ö. Salehi, A. Glos, and J. A. Miszczak, "Unconstrained binary models of the travelling salesman problem variants for quantum optimization," *Quantum Information Processing*, vol. 21, no. 2, p. 67, 2022.
- [8] G. Gutin and A. Punnen, *The Traveling Salesman Problem and its Variations. Combinatorial Optimization*, Kluwer Academic Press, Dordrecht, Netherlands, 2002.
- [9] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti, "A QUBO model for the traveling salesman problem with time windows," *Algorithms*, vol. 12, no. 11, p. 224, 2019.
- [10] R. P. A. Dilworth, "A decomposition theorem for partially ordered sets," *Annals of Mathematics*, vol. 51, no. 1, pp. 161-166, 1950.
- [11] G. B. Dantzig and D. R. Fulkerson, "Minimizing the number of tankers to meet a fixed schedule," *Naval Research Logistics Quarterly*, vol. 1, no. 3, pp. 217-222, 1954.
- [12] G. B. Dantzig and A. J. Hoffman, "Dilworth's theorem on partially ordered sets," in *Linear Inequalities and Related Systems*, pp. 207-214, Princeton University Press, Princeton, NJ, USA, 1956.
- [13] R. P. Dilworth, "Some combinatorial problems on partially ordered sets Combinatorial Analysis," in *Book: The Dilworth Theorems: Selected Papers of Robert P. Dil Worth*, Springer Science + Business Media, Berlin, Germany, 1990.
- [14] D. R. Fulkerson, "Note on Dilworth's decomposition theorem for partially ordered sets," *Proceedings of the American Mathematical Society*, vol. 7, no. 4, pp. 701-702, 1956.
- [15] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 1962.
- [16] P. C. Fishburn, *Interval Graphs and Interval Orders*, Wiley, Hoboken, NY, USA, 1985.
- [17] E. M. Badr and S. Almotairi, "On a dual direct cosine simplex type Algorithm and its computational behavior," *Mathematical Problems in Engineering*, Article ID 7361092, 8 pages, 2020.
- [18] E. M. Badr and H. elgendy, "A Hybrid water cycle—particle swarm optimization for solving the fuzzy underground water confined steady flow," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, p. 492, 2020.
- [19] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, John Wiley, Hoboken, NY, USA, 2010.
- [20] N. P. Theodorakatos, "Optimal phasor measurement unit placement for numerical observability using branch-and-bound and a binary-coded genetic algorithm," *Electric Power Components and Systems*, vol. 47, pp. 357-371, 2019.

- [21] N. P. Theodorakatos, "Fault location observability using phasor measurement units in a power network through deterministic and stochastic algorithms," *Electric Power Components and Systems*, vol. 47, no. 3, pp. 212–229, 2019.
- [22] N. P. Theodorakatos, N. M. Manousakis, and G. N. Korres, "Optimal placement of phasor measurement units with linear and non-linear models," *Electric Power Components and Systems*, vol. 43, no. 4, pp. 357–373, 2015.
- [23] E. Badr and M. Moussa, "On jump-critical ordered sets with jump number four," *Journal of Advances in Applied & Computational Mathematics*, vol. 1, pp. 8–13, 2014.