Eccentricity, Center and Radius Computations on the Cover Graphs of Distributive Lattices with Applications to Stable Matchings

Christine T. Cheng†  Eric McDermid‡  Ichiro Suzuki§

November 24, 2014

Abstract

Birkhoff’s fundamental theorem on distributive lattices states that for every distributive lattice \( \mathcal{L} \) there is a poset \( \mathcal{P}_L \) whose lattice of down-sets is order-isomorphic to \( \mathcal{L} \). Let \( G(\mathcal{L}) \) denote the cover graph of \( \mathcal{L} \). In this paper, we consider the following problems: Suppose we are simply given \( \mathcal{P}_L \). How do we compute the eccentricity of an element of \( \mathcal{L} \) in \( G(\mathcal{L}) \)? How about a center and the radius of \( G(\mathcal{L}) \)? While eccentricity, center and radius computations have long been studied for various classes of graphs, our problems are different in that we are not given the graph explicitly; instead, we only have a structure that implicitly describes the graph. By making use of the comparability graph of \( \mathcal{P}_L \), we show that all the said problems can be solved efficiently. One of the important implications of these results is that a center stable matching, a kind of fair stable matching, can be computed in polynomial time.

1 Introduction

A finite distributive lattice \( \mathcal{L} = (L, \leq) \) is a partially ordered set (poset) where for any two elements \( x, y \in L \), (i) their meet \( x \wedge y \) or greatest lower bound exists, (ii) their join \( x \vee y \) or least upper bound exists, and (iii) the meet and join operators distribute over each other. The bottom element of \( \mathcal{L} \) is \( \wedge L = 0 \) while its top element is \( \vee L = 1 \). As an example, consider the factors of a positive integer \( z \) ordered according to the divisibility relation. For any two factors \( f_1 \) and \( f_2 \), their meet is their greatest common factor, their join is their least common multiple while the bottom and top elements of the lattice are 1 and \( z \) respectively. Many more objects form a distributive lattice including the domino tilings of a polygon, the matchings of a connected bipartite planar graph, alternating sign matrices, etc. [21, 9].

Let \( \mathcal{P} = (P, \leq) \) be a poset. A subset \( P' \) of \( P \) is a down-set or order ideal of \( \mathcal{P} \) if whenever \( p \in P' \), all the predecessors of \( p \) in \( \mathcal{P} \) are also in \( P' \). Let \( D(\mathcal{P}) \) consist of the down-sets of \( \mathcal{P} \).

---

*A preliminary version of this paper appeared in the Proceedings of ICALP, 2011.
†Supported by NSF award CCF-0830678. Department of Computer Science, University of Wisconsin–Milwaukee. Email: ccheng@uwm.edu
‡Supported in part by the UWM Research Growth Initiative and NSF award CCF-0830678. Apple, Austin TX. Email: em4617@gmail.com
§Supported by the UWM Research Growth Initiative. Department of Computer Science, University of Wisconsin–Milwaukee. Email: suzuki@uwm.edu
Figure 1: Consider the distributive lattice $\mathcal{L}$ on the left. The poset $\mathcal{P}_L$ that encodes $\mathcal{L}$ is in the middle. In Birkhoff’s proof, $\mathcal{P}_L$ is the subposet induced by the join-irreducible elements of $\mathcal{L}$ – i.e., the elements whose in-degree is 1 in the Hasse diagram of $\mathcal{L}$. The down-sets of $\mathcal{P}_L$ (labeled without the curly braces) ordered according to the subset relation are shown on the right. Clearly, $\mathcal{L}$ and $(D(\mathcal{P}_L), \subseteq)$ are isomorphic distributive lattices.

It is not difficult to see that $(D(\mathcal{P}), \subseteq)$ is a distributive lattice. An important characterization of distributive lattices states that the converse is true as well. (See Figure 1 for an example.)

**Theorem 1** (Birkhoff[3]) *For every distributive lattice $\mathcal{L}$, there is poset $\mathcal{P}_L$ such that $(D(\mathcal{P}_L), \subseteq)$ is order-isomorphic to $\mathcal{L}$.*

---

Given a graph $G = (V, E)$, let $d(v, u)$ denote the distance between $v$ and $u$ in $G$. The **eccentricity** of $v$, $\text{ecc}(v)$, is equal $\max_u d(v, u)$. The **radius** of $G$, $\text{rad}(G)$, is equal to $\min_v \text{ecc}(v)$ while the **diameter** of $G$, $\text{diam}(G)$, is equal to $\max_v \text{ecc}(v)$. A node whose eccentricity is equal to the radius of $G$ is referred to as a **center** of $G$. A node that has the smallest total (or average) distance from all other nodes of $G$ is called a **median** of $G$. To illustrate some of these concepts, consider the popular game *Six Degrees of Kevin Bacon*. A person is given the name of an actor$^1$ $a$, and the goal is to identify a sequence of at most seven actors starting with $a$ and ending with Kevin Bacon so that any two consecutive actors in the sequence have appeared in a movie together; i.e., actor $a$ can “reach” Kevin Bacon in six steps or less. Graph theorists have long known that this is just a game on the *Actor’s graph* where actors that have appeared in a movie are the vertices, and two actors are adjacent if and only if they have appeared in a movie together. An inherent assumption of the game is that the eccentricity of Kevin Bacon is at most six, and Kevin Bacon is a center of this graph.

For a distributive lattice $\mathcal{L}$, let $G(\mathcal{L})$ denote the **cover graph** of $\mathcal{L}$, the undirected Hasse diagram of $\mathcal{L}$. In this paper, we consider the following problems: Given $\mathcal{P}_L$, *is there an efficient algorithm for finding a center of $G(\mathcal{L})$, expressed as a down-set of $\mathcal{P}_L$? How about computing the radius of $G(\mathcal{L})$? Suppose we are additionally given an element of $\mathcal{L}$ expressed as a down-set of $\mathcal{P}_L$, can we compute this element’s eccentricity efficiently?*

---

$^1$Although we use the word "actor", the person can be male or female.
The problems of computing eccentricities, radii, diameters and centers of graphs have a long and rich history starting with Jordan’s theorem in 1869 [17]. The theorem states that a tree either has one center or two centers which are adjacent to each other. While we can easily compute these four parameters by running breadth-first search from each node of the graph, there has been a steady stream of research to improve upon this brute force method. Corneil, Dragan and others [7, 8], for example, have shown that Lex-BFS (a variant of BFS) can either compute the diameter of a graph exactly or within one if the graph is chordal, interval, AT-free, etc. Others like Borassi et al. [5] have developed heuristics that involve a small but related runs of BFS to determine the diameter of real-world networks. Still others like Roditty et al. [22] have presented fast approximation algorithms for diameter computations. We note though that our problems are quite different from the ones considered in these papers because the graph under consideration is not given to us explicitly. Instead, we only have an auxiliary structure that encodes the graph so we have to rely on a different set of techniques to solve the problems.

Motivation. Our interest in finding a center of \( G(\mathcal{L}) \) given \( \mathcal{P}_\mathcal{L} \) originated from our work on stable matchings. An instance \( I \) of the stable marriage problem (SM) has \( n \) men and \( n \) women each of whom has a preference list that ranks members of the opposite gender in a linear order. A matching is a set of \( n \) disjoint man-woman pairs; it is stable if there is no man-woman pair who prefer each other over their partners in the matching. The goal of the problem is to find a stable matching of \( I \) if one exists. A seminal result of Gale and Shapley in the 1960s [10] states that every SM instance has a stable matching that can be computed in \( O(n^2) \) time. Today, centralized stable matching algorithms are used to match medical residents to hospitals [23] and students to schools [1, 2].

An SM instance can have up to \( 2^{O(n)} \) stable matchings [12]. It turns out, however, that the Gale-Shapley algorithm outputs only two kinds: the man-optimal/woman-pessimal stable matching and the woman-optimal/man-pessimal stable matching. In the man-optimal stable matching, every man is matched to his best partner in all of the stable matchings while simultaneously every woman is matched to her worst partner in all of the stable matchings; the woman-optimal/man-pessimal stable matching has the opposite properties. Hence, in spite of the fact that the Gale-Shapley algorithm solves SM efficiently, we might not want to use the solution as one group is extremely happy while the other group is extremely unhappy. This motivates the problem of finding fair stable matchings.

Interestingly, stable matchings also form distributive lattices. Let \( M(I) \) denote the set of stable matchings of \( I \). For any two stable matchings \( \mu \) and \( \mu' \) in \( M(I) \), let \( \mu \preceq \mu' \) if, for each man \( m \), either \( m \) has the same partner in \( \mu \) and \( \mu' \) or \( m \) prefers his partner in \( \mu \) over his partner in \( \mu' \). Conway was the first to recognize that \( (M(I), \preceq) \) forms a distributive lattice [18]. In the SM literature, the poset that encodes \( (M(I), \preceq) \) is referred to as the rotation poset of \( I \) and can be derived directly from the preference lists of the participants. For ease of notation, we shall simply refer to it as \( \mathcal{R}(I) \) instead of \( \mathcal{P}_{(M(I), \preceq)} \). The elements of \( \mathcal{R}(I) \) are called rotations, which are the simplest moves one can make to modify one stable matching into another. Surprisingly, while \( M(I) \) can have an exponential number of elements in \( n \), \( \mathcal{R}(I) \) has \( O(n^2) \) elements and can be constructed in \( O(n^2) \) time. Moreover, given a down-set of \( \mathcal{R}(I) \), the stable matching that corresponds to the down-set can be obtained in \( O(n^2) \) time [12]. For these reasons, many computational problems on the lattice \( (M(I), \preceq) \) rely on \( \mathcal{R}(I) \). Lastly, we note that for everyly distributive lattice \( \mathcal{L} \), there is an SM instance \( I_{\mathcal{L}} \) whose lattice of stable matchings is order-isomorphic to \( \mathcal{L} \) [4, 15].

Different notions of fair stable matchings have been considered. For example, an egalitarian stable matching maximizes the total happiness of all the individuals while a minimum-regret stable
matching maximizes the happiness of the unhappiest person in the matching. Both stable matchings can be computed in polynomial time \([11, 16]\). We refer to these these types of stable matchings as *locally-fair* because they take into account the particulars of the matchings (i.e., who is matched to whom) and seek to ensure that all the participants are as happy as possible.

On the other hand, we say that stable matchings are *globally-fair* when their fairness is derived from being good representatives of \(M(I)\). Let \(G(I)\) denote the cover graph of \((M(I), \preceq)\). It turns out that in \(G(I)\) two stable matchings are adjacent if and only if they differ by a rotation. Now consider the median and center vertices of \(G(I)\), which we call the median and center stable matchings of \(I\) respectively. By definition, the former minimizes the total (or average) number of moves needed to transform it to all other stable matchings of \(I\) while the latter minimizes the maximum number of moves needed to transform it to a stable matching of \(I\). We argue that these stable matchings are good representatives of \(M(I)\) for the same reason that a median or a mean is a good representative of a set of numbers – they “summarize” the entire set. Can a median or center stable matching of \(I\) be computed efficiently? As we noted earlier, \(I\) can have an exponential number of stable matchings so \(G(I)\) cannot be constructed explicitly. Instead, we have to use \(R(I)\) as our starting point. Cheng \([6]\) showed that computing a median stable matching of \(I\) is \#P-hard. The corresponding problem for a center stable matching of \(I\) is the open problem we wish to address in this paper.

**Our Results.** Let \(L\) be a distributive lattice, and let \(P_L\) be the poset that encodes \(L\). The comparability graph of \(P_L, C(P_L)\), is the graph whose vertices are the elements of \(P_L\) and two elements are adjacent if and only if they are comparable in \(P_L\). The key insight we make in this paper is that \(C(P_L)\) is the structure that should be analyzed to compute the eccentricities of the elements of \(L\) in \(G(L)\), a center of \(G(L)\), and the radius of \(G(L)\). Our results are as follows:

First, we prove that for each element \(x\) of \(L\), the eccentricity of \(x\) in \(G(L)\) can be derived by computing the size of a maximum matching of a particular bipartite subgraph of \(C(P_L)\). Consequently, given \(P_L\), the eccentricity of \(x\) in \(G(L)\) can be obtained in \(O(f(k) + k^{2.5})\) time where \(|P_L| = k\) and \(f(k)\) is the time it takes to construct the transitive closure of \(P_L\) from the input representation of \(P_L\).

Second, we show that any maximum matching \(F\) of \(C(P_L)\) can be transformed into a center of \(G(L)\). Furthermore, the eccentricity of a center, which is also the radius of \(G(L)\), is equal to \(|P_L| - |F|\). Hence, given \(P_L\), a center and the radius of \(G(L)\) can also be computed in \(O(f(k) + k^{2.5})\) time where \(k\) and \(f(k)\) are defined in the previous paragraph.

Third, we provide a characterization of the centers of \(G(L)\) that is based on what we call the lowest maximum matchings of \(C(P_L)\). An interesting corollary of this result is that the “shape” induced by the centers of \(G(L)\) is a union of hypercubes with dimension \(|P_L| - 2|F|\), where \(F\) is a maximum matching of \(C(P_L)\).

Finally, our results imply that a center stable matching of an SM instance \(I\) can be computed in \(O(n^5)\) time where \(n\) is the number of men and women in the instance.\(^2\) This is in sharp contrast to median stable matchings which is \#P-hard to compute. Thus, center stable matchings are the first type of globally-fair stable matchings we know of that can be computed efficiently.

\(^2\)We note that \(O(n^5)\) may seem large, but \(n^5 = |I|^{2.5}\) only, where \(|I|\) is the input size, because specifying the preference lists of all the participants take \(\Theta(n^2)\) time and space.
2 Eccentricity, Center and Radius Computations

Let \( \mathcal{L} \) be a distributive lattice. Recall that \( \mathcal{P}_\mathcal{L} = (P_\mathcal{L}, \leq) \) is the poset that encodes \( \mathcal{L} \) while \( G(\mathcal{L}) \) is the cover graph of \( \mathcal{L} \). Since elements of \( \mathcal{L} \) are down-sets of \( \mathcal{P}_\mathcal{L} \), we shall refer to them using capital letters \( V, W \), etc. We shall also use \( d(V,W) \) to denote the distance between \( V \) and \( W \) in \( G(\mathcal{L}) \). Here is a straightforward way of computing \( d(V,W) \).

**Proposition 1** For any two elements \( V \) and \( W \) of \( \mathcal{L} \), \( d(V,W) = |W - V| + |V - W| \).

In \( \mathcal{L} \), the bottom element is \( \hat{0} = \emptyset \) while the top element is \( \hat{1} = P_L \). Thus, \( d(\hat{0}, \hat{1}) = |P_L| \). But for any two elements \( V \) and \( W \) of \( \mathcal{L} \), \( d(V,W) \leq |P_L| \). Hence, the diameter of \( G(\mathcal{L}) \) is \( |P_L| \), which we note is trivial to compute unlike the radius of \( G(\mathcal{L}) \).

We begin by showing how the eccentricity of an arbitrary element of \( \mathcal{L} \) in \( G(\mathcal{L}) \) can be computed. Let \( C(\mathcal{P}_\mathcal{L}) \) denote the comparability graph of \( \mathcal{P}_\mathcal{L} \). For a down-set \( V \) of \( \mathcal{P}_\mathcal{L} \), let \( H[V, \overline{V}] \) denote the bipartite subgraph of \( C(\mathcal{P}_\mathcal{L}) \) induced by the partition \( [V, \overline{V}] \). That is, for \( p \in V \) and \( q \in \overline{V} \), \( pq \) is an edge in \( H[V, \overline{V}] \) if and only if \( p \) and \( q \) are comparable in \( \mathcal{P}_\mathcal{L} \). Notice though that since \( V \) is already a down-set, all the predecessors of \( p \) are in \( V \). Thus, \( p \) and \( q \) are comparable only because \( p < q \); i.e., \( q \) is a successor of \( p \) in \( \mathcal{P}_\mathcal{L} \).

Consider an arbitrary matching \( F \) of \( H[V, \overline{V}] \). We write \( F \) as \( \{(p_1, q_1), (p_2, q_2), \ldots, (p_f, q_f)\} \) where \( p_i < q_i \) for \( i = 1, \ldots, f \). Let \( B(F) = \{p_1, p_2, \ldots, p_f\} \) and \( T(F) = \{q_1, q_2, \ldots, q_f\} \). We say that \( B(F) \) contains the bottom elements of \( F \) while \( T(F) \) contains the top elements of \( F \).

**Lemma 1** Suppose \( \mathcal{P}_\mathcal{L} \) has \( k \) elements. Let \( V \) be a down-set of \( \mathcal{P}_\mathcal{L} \) and \( F \) a matching of \( H[V, \overline{V}] \). Then the eccentricity of \( V \) in \( G(\mathcal{L}) \) is at most \( k - |F| \). Consequently, when \( F_V \) is a maximum matching of \( H[V, \overline{V}] \), the eccentricity of \( V \) in \( G(\mathcal{L}) \) is at most \( k - |F_V| \).

**Proof** Let \( W \) be a down-set of \( \mathcal{P}_\mathcal{L} \). First, we note that for every edge \( (p_i, q_i) \in F \), whenever \( q_i \in W \), \( p_i \in W \) as well because \( p_i \) is a predecessor of \( q_i \). Thus, \( |W \cap T(F)| \leq |W \cap B(F)| \). Now consider the distance between \( V \) and \( W \) in \( G(\mathcal{L}) \):

\[
d(V, W) = |W - V| + |V - W| \\
= |W \cap T(F)| + |W \cap (\overline{V} - T(F))| + |V| - |W \cap B(F)| - |W \cap (V - B(F))| \\
\leq |W \cap (\overline{V} - T(F))| + |V| - |W \cap (V - B(F))| \\
\leq |\overline{V} - T(F)| + |V| - |W \cap (V - B(F))| \\
= k - |F|.
\]

Since we chose \( W \) arbitrarily, it follows that the distance of \( V \) from any element of \( \mathcal{L} \) is at most \( k - |F| \). And since \( F_V \) is another matching of \( H[V, \overline{V}] \), the result holds for \( F_V \) too. \( \square \)

**Lemma 2** Suppose \( \mathcal{P}_\mathcal{L} \) has \( k \) elements. Let \( V \) be a down-set of \( \mathcal{P}_\mathcal{L} \), and let \( F_V \) be a maximum matching of \( H[V, \overline{V}] \). Then there is a down-set \( W \) of \( \mathcal{P}_\mathcal{L} \) so that \( d(V, W) = k - |F_V| \).
Proof We obtain $W$ from a minimum vertex cover of $H[V, \overline{V}]$. Let $N_V \cup N_{\overline{V}}$, where $N_V \subseteq V$ and $N_{\overline{V}} \subseteq \overline{V}$, be a minimum vertex cover of $H[V, \overline{V}]$. By König’s theorem, the size of this vertex cover must be equal to $|F_V|$. We also note that there must be no edges between $V - N_V$ and $\overline{V} - N_{\overline{V}}$ because such edges cannot be covered by $N_V \cup N_{\overline{V}}$. Furthermore, each $p \in N_V$ must cover at least one edge whose other endpoint is in $\overline{V} - N_{\overline{V}}$; otherwise, $N_V \cup N_{\overline{V}} - \{p\}$ is still a vertex cover of $H[V, \overline{V}]$, contradicting the minimality of $N_V \cup N_{\overline{V}}$. By the same reasoning, each $q \in N_{\overline{V}}$ must cover at least one edge whose other endpoint is in $V - N_V$.

Set $W = N_V \cup (\overline{V} - N_{\overline{V}})$. We now argue that $W$ is a down-set of $\mathcal{P}_L$. Suppose it is not. Either some $p \in N_V$ or some $q \in (\overline{V} - N_{\overline{V}})$ is missing a predecessor in $W$. Consider the first case. If $p$ is missing a predecessor $x$, $x$ can only belong to $V - N_V$ because $V$ is already a down-set. But we know that $p$ covers an edge $qp$ such that $q \in \overline{V} - N_{\overline{V}}$; i.e., $x < p < q$. Hence, $xq$ is an edge of $H[V, \overline{V}]$. But we noted that there cannot be an edge between $V - N_V$ and $\overline{V} - N_{\overline{V}}$ so the first case cannot happen. In the second case, if $q$ is missing a predecessor $x$, $x$ either belongs to $N_{\overline{V}}$ or to $V - N_V$. If $x \in N_{\overline{V}}$, we know that $x$ covers some edge $yx$ such that $y \in V - N_V$; i.e., $y < x < q$. It follows that $yq$ is an edge of $H[V, \overline{V}]$. If $x \in V - N_V$, $xq$ is an edge of $H[V, \overline{V}]$. For both of these subcases, we conclude that there is an edge between $V - N_V$ and $\overline{V} - N_{\overline{V}}$, again a contradiction. So the second case is not possible either. Thus, $W$ is a closed subset of $\mathcal{P}_L$.

Finally, consider the distance between $V$ and $W$ in $G(L)$:

$$d(V, W) = |W - V| + |V - W| = |W \cap \overline{V} + |V| - |W \cap V| = |\overline{V} - N_{\overline{V}}| + |V| - |N_V| = |\overline{V}| - |N_{\overline{V}}| + |V| - |N_V| = k - (|N_{\overline{V}}| + |N_V|) = k - |F_V|.$$

\[\square\]

**Theorem 2** Suppose $\mathcal{P}_L$ has $k$ elements and $V$ is a down-set of $\mathcal{P}_L$. Then the eccentricity of $V$ in $G(L)$ is $k - |F_V|$ where $F_V$ is a maximum matching of $H[V, \overline{V}]$. Consequently, computing the eccentricity of $V$ in $G(L)$ can be done in $O(f(k) + k^{2.5})$ time, where $f(k)$ is the time it takes to construct the transitive closure of $\mathcal{P}_L$ from the input representation of $\mathcal{P}_L$.

**Proof** The fact that the eccentricity of $V$ is $k - |F_V|$ follows directly from Lemmas 1 and 2. Suppose $Tr(\mathcal{P}_L)$ is the transitive closure of $\mathcal{P}_L$. Then $C(\mathcal{P}_L)$ is just the undirected version of $Tr(\mathcal{P}_L)$; it has $k$ vertices and $O(k^2)$ edges. Given $V$, the bipartite subgraph $H[V, \overline{V}]$ can be obtained in $O(k^2)$ time. Finally, computing a maximum matching $F_V$ of $H[V, \overline{V}]$ can be done in $O(k^{2.5})$ using the Hopcroft-Karp algorithm [13]. Thus, computing the eccentricity of $V$ can be done in $O(f(k) + k^{2.5})$ time. \[\square\]

**Example.** Consider the poset $\mathcal{P}_L$ shown in Figure 2. When $V = \{a, b, c\}$, the bipartite graph $H[V, \overline{V}]$ has no edges so $F_V$ is an empty matching. Thus, the eccentricity of $\{a, b, c\}$ is $6 - 0 = 6$. On the other hand, when $V = \{a, b, d\}$, $F_V$ has two edges so the eccentricity of $\{a, b, d\}$ is $6 - 2 = 4$.

**Corollary 1** For each element $V$ of $\mathcal{L}$, let $F_V$ denote a maximum matching in $H[V, \overline{V}]$. The element $V^*$ is a center of $G(L)$ if and only if $|F_{V^*}| = \max_{V \in \mathcal{L}} |F_V|$. 

6
Figure 2: Suppose $P$ is missing from $B$, however, not a lowest maximum matching because $F$ is a low maximum matching and $F$ has $x$ cases to consider: $F$ is a low matching and $|F| = |F_0|$. 

**Proof** Let $F_0 = F$ be a matching of $C(P_L)$ such that $Z(F)$ is not a down set of $P_L$. To transform $F$ into a low matching, we will perform a sequence of switches. Suppose $x \in Z(F)$ has a predecessor that is missing from $Z(F)$. This predecessor must be some $q'$ such that $(p', q') \in F$. There are two cases to consider:

*Case 1:* $x \in B(F)$ so that $x = p$ and $(p, q) \in F$. Hence, $p' < q' < p < q$ in $P_L$, and $p'p, q'q$ are edges of $C(P_L)$. *Switch $(p', q')$, $(p, q)$ with $(p', p), (q', q)$*; that is, replace $F$ with $F' = F - \{(p', q'), (p, q)\} \cup \{(p', p), (q', q)\}$. Clearly, $F'$ is another matching of $C(P_L)$ with $B(F') = B(F) - \{p\} \cup \{q'\}$ and $U(F') = U(F)$. Its size is also equal to that of $F$.

*Case 2:* $x \in U(F)$. Hence, $p' < q' < x$ in $P_L$, and $p'x$ is an edge of $C(P_L)$. *Switch $(p', q')$ with $(p', x)$;* that is, replace $F$ with $F' = F - \{(p', q'), (p', x)\}$. Again, $F'$ is a matching of $C(P_L)$ with $U(F') = U(F) - \{x\} \cup \{q'\}$ and $B(F') = B(F)$. Its size is equal to that of $F$.

Observe that in both cases $Z(F')$ is obtained from $Z(F)$ by replacing the element with a missing predecessor by its missing predecessor. Hence, if we process the elements of $P_L$ in a top-down manner, once an element is removed from $Z(F)$, it is never added into the set again. That is, do the following:

*Step 1:* Topologically sort the elements of $P_L$, and let the result be $x_1, x_2, \ldots, x_k$; i.e., for each $x_i$, all its predecessors occur before it in the ordering.

*Step 2:* For $i = k$ to 1, if $x_i \in Z(F)$, check that all the predecessors of $x_i$ also belong to $Z(F)$. If not, perform a switch, and update both $F$ and $Z(F)$. At the end of the for loop, return $F$.

Let us call the output $F_1$. Since switches preserve matchings and size, $F_1$ is another matching of $C(P_L)$ whose size is equal to that of $F_0$. But suppose it is not a low matching of $C(P_L)$. This must mean that for some $x_i \in Z(F_1)$, a predecessor $x_h, h < i$, was part of $Z(F)$ when $x_i$ was processed in Step 2 but was later removed. For the latter to happen, however, $x_h$ must have a
missing predecessor and was replaced by this predecessor. That is, when \( x \) was processed, it also had a missing predecessor – a contradiction. Hence, the output \( F_1 \) must be a low matching of \( C(P_L) \). \( \square \)

**Theorem 3** Suppose \( P_L \) has \( k \) elements. Let \( F^* \) be a low maximum matching of \( C(P_L) \). Then \( Z(F^*) \) is a center of \( L \) and its eccentricity is \( k - |F^*| \). Furthermore, given \( P_L \), such a center can be computed in \( O(f(k) + k^{2.5}) \) time, where \( f(k) \) is the time it takes to construct the transitive closure of \( P_L \) from the input representation of \( P_L \).

**Proof** Since \( F^* \) is a low maximum matching of \( C(P_L) \), \( Z(F^*) \) is a down-set of \( P_L \) and therefore an element of \( L \). Let \( V^* = Z(F^*) \) and consider \( H[V^*, V^*] \). By the definition of \( Z(F^*) \), \( V^* = T(F^*) \). Thus, \( F^* \) is a matching in \( H[V^*, V^*] \). And since \( F^* \) is a maximum matching of \( C(P_L) \) and \( H[V^*, V^*] \) is a subgraph of \( C(P_L) \), it follows that \( F^* \) is also a maximum matching of \( H[V^*, V^*] \). Additionally, for any element \( V \) of \( L \), the subgraph \( H[V, V] \) cannot have a maximum matching larger than \( F^* \). Hence, according to Corollary 1, \( V^* \) has to be a center of \( L \). According to Theorem 2, its eccentricity is \( k - |F^*| \).

Suppose we are given \( Tr(P_L) \), the transitive closure of \( P_L \). Then \( C(P_L) \) is just its undirected version with \( k \) vertices and \( O(k^2) \) edges. Using the algorithm of Micali and Vazirani [19], a maximum matching \( F \) of \( C(P_L) \) can be computed in \( O(k^{2.5}) \) time. Checking if \( F \) is a low maximum matching can be done in \( O(k^2) \) time. If it is not, the proof of Lemma 3 describes how it can be transformed into one. Step 1 takes \( O(k^2) \) time. In step 2, each element’s predecessor is examined at most once; switching takes \( O(1) \) time. Hence, step 2 also takes \( O(k^2) \) time. Thus, finding a low maximum matching of \( C(P_L) \) given \( Tr(P_L) \) takes \( O(k^{2.5}) \). Finally, extracting \( V^* \) from the low maximum matching takes \( O(k) \) time so the overall running time for computing \( V^* \) is \( O(f(k) + k^{2.5}) \). \( \square \)

### 3 Describing all the centers of \( G(L) \)

Applying the techniques used in the previous section, we shall now provide another characterization of the centers of \( G(L) \) that will give us some insight into the “shape” formed by the these elements in \( G(L) \). Let us say that a matching \( F \) of \( C(P_L) \) is a **lowest matching** if \( Z(F) \) and \( B(F) \) are down-sets of \( P_L \) (see Figure 2).

**Lemma 4** Every matching \( F_0 \) in \( C(P_L) \) can be transformed into another matching \( F_2 \) so that \( F_2 \) is a lowest matching and \( |F_2| = |F_0| \).

**Proof** Lemma 3 showed that \( F_0 \) can be transformed into another matching \( F_1 \) so that \( |F_1| = |F_0| \) and \( F_1 \) is a low matching of \( C(P_L) \). But suppose \( F_1 \) is not a lowest matching. This means that some \( p \in B(F_1) \) with \( (p, q) \in F_1 \) has a predecessor \( x \) that is missing from \( B(F_1) \). But since \( Z(F_1) \) is already a down-set of \( P_L \), \( x \) must lie in \( U(F_1) \). Thus, \( x < p < q \) and \( xq \) is an edge of \( C(P_L) \). Switch \( (p, q) \) with \( (x, q) \); i.e., replace \( F_1 \) with \( F'_1 = F_1 - \{p, q\} \cup \{x, q\} \) so that \( B(F'_1) = B(F_1) - \{p\} \cup \{x\} \) and \( Z(F'_1) = Z(F_1) \). Again, \(|F'_1| = |F_1| \). Following the two-step algorithm in the proof of Lemma 3, where \( Z(F) \) is replaced by \( B(F_1) \) in the algorithm and the switch is as defined above we can now transform \( F_1 \) into a lowest matching \( F_2 \) such that \( |F_2| = |F_1| \). \( \square \)

**Lemma 5** Let \( F \) be a lowest maximum matching of \( C(P_L) \). Then every set of the form \( V = B(F) \cup U' \) where \( U' \subseteq U(F) \) is a center of \( G(L) \).
Theorem 3, the eccentricity of $V$ the necessity direction. Suppose $V$

Proof

By Theorem 2, the bipartite graph $H$ of $F$

Theorem 4

Of $V$

Hence,$V$ show that for $C$

is a lowest maximum matching of $C$ of $L$

be an element of $L$. Using the same property mentioned in the proof of Lemma 1 (i.e., for each $q \in T(F)$, there is a distinct $p \in B(F)$) and the fact that $W$ is a closed subset of $P_L$, it must be the case that $|W \cap T(F)| \leq |W \cap B(F)|$. So consider the distance between $W$ and $V$ in $G(L)$:

$$d(W, V) = |W - V| + |V - W|$$

$$= |W \cap T(F)| + |W \cap (U(F) - U')| + |V| - |W \cap B(F)| - |W \cap U'|$$

$$\leq |V| + |W \cap (U(F) - U')| - |W \cap U'|$$

$$= |B(F)| + |U'| + |W \cap (U(F) - U')| - |W \cap U'|$$

$$\leq |B(F)| + |U'| + |U(F) - U'|$$

$$= |B(F)| + |U(F)|$$

$$= |F| + (k - 2|F|)$$

$$= k - |F|.$$  

Thus, the eccentricity of $V$ is at most $k - |F|$ so $V$ must be a center of $G(L)$. □

Theorem 4 Let $V$ be an element of $L$. Then $V$ is a center of $G(L)$ if and only if there is a lowest maximum matching $F$ of $C(P_L)$ such that $V = B(F) \cup U'$ with $U' \subseteq U(F)$.

Proof The necessity direction of the theorem follows immediately from Lemma 5. So consider the necessity direction. Suppose $V$ is a center of $G(L)$ and $P_L$ has $k$ elements. According to Theorem 3, the eccentricity of $V$ is $k - f$ where $f$ is the size of a maximum matching of $C(P_L)$. By Theorem 2, the bipartite graph $H[V, \overline{V}]$ has a maximum matching $F$ whose size is precisely $f$. Thus, $F$ is a maximum matching of $C(P_L)$ too. Clearly, $V = B(F) \cup U'$ for some $U' \subseteq U(F)$. If $F$ is a lowest maximum matching of $C(P_L)$, we are done. But suppose it is not. Let us assume the worst case – that $F$ is not even a low maximum matching of $C(P_L)$. Using the algorithm outlined in the proof of Lemma 3, transform $F$ into a low maximum matching $F'$. If $F'$ is not a lowest maximum matching, transform it into one using a similar algorithm as described in Lemma 4 and call it $F''$. Let us now show that, like $F$, both $F'$ and $F''$ are maximum matchings of $H[V, \overline{V}]$.

If $F$ is not a lowest maximum matching, $Z(F)$ is not a closed subset of $C(P_L)$. Since $V$ is a closed subset of $C(P_L)$, the only way this can happen is for some $x \in U(F) \cap \overline{V}$ to have a predecessor $q$ with $(p, q) \in F$. To correct this, the algorithm will switch $(p, q)$ with $(p, x)$ in $F$. Possibly many more switches like this will have to be performed until $Z(F)$ is a closed subset of $C(P_L)$. What is true though is that every such switch preserves the property that $F$ is a maximum matching of $H[V, \overline{V}]$. Hence, at the end of the algorithm, the resulting low maximum matching $F'$ is still a matching of $H[V, \overline{V}]$.

Now, if $F'$ is not a lowest maximum matching, it is because $B(F')$ is not a closed subset of $C(P_L)$. Again, since $V$ is a closed subset of $C(P_L)$, this must mean that some $p \in B(F')$ with $(p, q) \in F'$ is missing a predecessor $x \in U(F') \cap V$. To correct this, the algorithm will switch $(p, q)$
with \((x, q)\) in \(F'\); that is, \(F'\) remains a maximum matching of \(H[V, \overline{V}]\). Applying the same reasoning as the previous paragraph, it must be the case that the resulting lowest maximum matching \(F''\) at the end of the algorithm is still a maximum matching of \(H[V, \overline{V}]\). As a result, \(V = B(F'') \cup U'\) for some \(U' \subseteq U(F'')\). \(\square\)

Let us now describe the “shape” formed by the centers of \(G(\mathcal{L})\) using the above characterization. The corollary below is quite interesting in light of the fact that the medians of \(G(\mathcal{L})\) induce a hypercube [20, 14]. That is, either (i) \(G(\mathcal{L})\) has one median only or (ii) there are two elements \(W_1\) and \(W_2\) of \(\mathcal{L}\) with \(W_1 \subseteq W \subseteq W_2\) such that every \(W\) where \(W_1 \subseteq W \subseteq W_2\) is a median of \(G(\mathcal{L})\).

**Corollary 2** Let \(\mathcal{L}\) be a distributive lattice. Assume \(\mathcal{P}_\mathcal{L}\) has \(k\) elements, and a maximum matching of \(C(\mathcal{P}_\mathcal{L})\) has size \(f\). The set of centers of \(G(\mathcal{L})\) is a union of hypercubes with dimension \(k - 2f\).

**Proof** Let \(F\) be a lowest maximum matching of \(C(\mathcal{P}_\mathcal{L})\). Since the elements of the set \(\{B(F) \cup U' : U' \subseteq U(F)\}\) are in one-to-one correspondence with the subsets of \(U(F)\) and \(|U(F)| = k - 2|F|\), the set must induce a hypercube of dimension \(k - 2|F|\) in \(G(\mathcal{L})\). According to Theorem 4, every center of \(G(\mathcal{L})\) must come from such a set. It follows that the centers of \(G(\mathcal{L})\) is composed of hypercubes with dimension \(k - 2|F|\). \(\square\)

When we first considered the notion of a center stable matching, our initial guess was that all elements \(V\) with \(|V| = |\mathcal{P}_\mathcal{L}|/2\) or \(|\mathcal{P}_\mathcal{L}|/2\) are centers of \(G(\mathcal{L})\) because they lie at the middle of paths connecting the bottom and top elements of \(\mathcal{L}\). It is not difficult to show that we guessed wrong. For instance, consider \(\mathcal{P}_\mathcal{L}\) in Figure 2. From Theorem 3, we know that the eccentricity of a center of \(G(\mathcal{L})\) is \(6 - 2 = 4\). However, the set \(\{a, b, c\}\), which contains half of the elements of \(\mathcal{P}_\mathcal{L}\), has eccentricity 6 as we noted earlier. It turns out though that our intuition was partly correct.

**Corollary 3** For every distributive lattice \(\mathcal{L}\), there is a maximum-length chain from \(\hat{0}\) to \(\hat{1}\) whose middle elements are centers of \(G(\mathcal{L})\). Equivalently, in \(G(\mathcal{L})\), there is a path whose length is the diameter of \(G(\mathcal{L})\) and whose middle nodes are centers of \(G(\mathcal{L})\).

**Proof** Let \(F\) be a lowest maximum matching of \(C(\mathcal{P}_\mathcal{L})\) with \(V_1 = B(F)\) and \(V_2 = B(F) \cup U(F)\). Let \(R_1\) be a contiguous chain that connects \(\hat{0}\) to \(V_1\) and \(R_2\) be a contiguous chain that connects \(V_2\) to \(\hat{1}\). According to Theorem 4, any contiguous chain \(R\) that connects \(V_1\) to \(V_2\) in \(\mathcal{L}\) is made up of centers of \(G(\mathcal{L})\). Let \(|\mathcal{P}_\mathcal{L}| = k\). Since \(d(\hat{0}, \hat{1}) = k\), \(d(\hat{0}, V_1) = |F|\), \(d(V_2, \hat{1}) = |F|\), and \(|F| \leq k/2\), the concatenation of \(R_1\), \(R\), \(R_2\) is a maximum-length chain from \(\hat{0}\) to \(\hat{1}\) whose middle nodes are centers of \(G(\mathcal{L})\). \(\square\)

4 Final remarks

Given the poset \(\mathcal{P}_\mathcal{L}\) that encodes a distributive lattice \(\mathcal{L}\) we have shown that the eccentricity of an element of \(\mathcal{L}\) in \(G(\mathcal{L})\), a center and the radius of \(G(\mathcal{L})\) can all be computed efficiently. These results are surprising in that we never construct \(G(\mathcal{L})\) explicitly; instead our analysis and computation is based solely on the auxiliary structure \(\mathcal{P}_\mathcal{L}\). In contrast, finding a median of \(G(\mathcal{L})\) given \(\mathcal{P}_\mathcal{L}\) is \#P-hard as shown in [6]. We now describe the implications of our result for finding a center stable matching.
Corollary 4  Let $I$ be an SM instance with $n$ men and $n$ women. Finding a center stable matching of $I$ takes $O(n^5)$ time.

**Proof**  As noted in the introduction, $\mathcal{R}(I)$, the poset that encodes the distributive lattice of stable matchings of $I$, has $O(n^2)$ elements; i.e., $k$ in Theorem 3 is $O(n^2)$. Furthermore, a directed acyclic graph representation of $\mathcal{R}(I)$ has $O(n^2)$ vertices and edges and can be constructed from $I$ in $O(n^2)$ time [12]. Applying the depth-first search algorithm from each node of this directed graph will enable us to construct the transitive closure of $\mathcal{R}(I)$ in $O(n^4)$ time. Hence, $f(k)$ in Theorem 3 is $O(n^4)$, and computing a center stable matching of $I$ expressed as a closed subset of $\mathcal{R}(I)$ takes $O(n^5)$ time. Finally, converting this closed subset into a stable matching of $I$ takes $O(n^2)$ time [12]. The corollary follows.  

**References**


