Discrete Mathematics 72 (1988) 337–346
North-Holland

AN ALGORITHM FOR SOLVING THE JUMP NUMBER PROBLEM

Maciej M. SYSŁO
Institute of Computer Science, University of Wrocław, ul. Przesmyckiego 20, 51151 Wrocław, Poland

Received 15 June 1986
Revised 15 August 1987

First, Cogis and Habib (*RAIRO Inform. Théor.* 13 (1979), 3–18) solved the jump number problem for series-parallel partially ordered sets (posets) by applying the greedy algorithm and then Rival (*Proc. Amer. Math. Soc.* 89 (1983), 387–394) extended their result to N-free posets. The author (*Order* 1 (1984), 7–19) provided an interpretation of the latter result in the terms of arc diagrams of posets explaining partly tractability of this special case.

In this paper, we present an algorithm for solving the jump number problem on arbitrary posets which extends the author's approach applied to N-free posets and makes use of two new types of greedy chains in posets introduced in companion papers [8, 9]. Complexity analysis of the algorithm supports our expectation that, going from N-free to arbitrary posets, the complexity of the problem increases with the number of dummy arcs required in their arc diagrams. The algorithm works in time which is linear in the poset size but factorial in the number of dummies, therefore it is a polynomial-time algorithm for posets with bounded number of dummies in their arc diagrams.

1. Preliminaries

A partially ordered set $(P, \leqslant)$ is in this paper simply written as P and called a *poset*. We assume also that all sets are finite. A *linear extension* of P is a total ordering $L = p_1 p_2 \cdots p_n$ of P preserving the relation, that is if $p_i < p_j$ in P then $i < j$. A pair (p_i, p_{i+1}) is a *jump* of L if $p_i \not< p_{i+1}$ in P. The jumps partition L into chains C_i of P, so we can write $L = C_0 + C_1 + \cdots + C_k$. The *jump number* $s(P)$ of P is equal to minimum k taken over all linear extensions L of P. A linear extension L of P is *optimal* if it has $s(P)$ jumps. The *jump number problem* consists in evaluating $s(P)$ and constructing an optimal linear extension. Although the problem is NP-complete (see Pulleyblank [4]), there exist polynomial-time algorithms for some special classes of posets defined by forbidden substructures (see Rival [5] and also [6]) or restricted by bounding some of their parameters (see [1] and [2]). The aim of this paper is to provide yet another algorithm of the latter type which generates special linear extensions introduced in [8].

We now define some basic graph-theoretic notions since we shall make a significant use of digraph representations of posets. A *digraph* may contain loops

This paper has been presented at the First Japan Conference on Graph Theory and Applications, Hakone (Japan), June 1–5, 1986.

and multiple arcs, so it is defined as $D = (V, A, t, h)$, where V is the *vertex* set, A is the *arc* set, and t, h (t for *tail* and h for *head*) are two incidence mappings $t, h: A \to V$. An arc $a \in A$ is of the form $a = (t(a), h(a))$. A sequence of arcs $\pi = (a_1, a_2, \ldots, a_l)$ is a *path* of length l if $h(a_i) = t(a_{i+1})$ for $i = 1, 2, \ldots, l-1$. The path π *begins* with the arc a_1 and also with the vertex $t(a_1)$. Similarly, π *terminates* with a_l and with $h(a_l)$. Therefore, we can write $t(\pi) = t(a_1)$ and $h(\pi) = h(a_l)$. A digraph is *acyclic* if it contains no path of length greater than 1 and such that $h(a_l) = t(a_1)$. The *transitive closure* of D is denoted by $\mathrm{tc}\, D = (V, \mathrm{tc}\, A, t^*, h^*)$, where $(a_1, a_2, \ldots, a_l)$, $l \geq 1$, is a path in D if and only if $\mathrm{tc}\, A$ contains an arc b such that $t^*(a_1) = t^*(b)$ and $h^*(a_l) = h^*(b)$. Let us denote $A^* = \mathrm{tc}(A) \cup \{(v, v): v \in C\}$.

Posets can be represented by graphs and digraphs. In this paper we make a significant use of digraph representations (called arc diagrams) in which the poset elements are assigned to arcs and the relation is preserved along the paths of the digraphs. Formally, an *arc diagram* of a poset P is an acyclic digraph $D^A(P) = (V, R, t, h)$ without loops (but possibly with parallel arcs) and a mapping $\phi: P \to R$ such that for every $p, q \in P$, $p \neq q$ we have

$$p < q \text{ in } P \text{ iff } (h^*(\phi(p)), t^*(\phi(q))) \in R^*,$$

where t^*, h^* are the incidence mappings of $\mathrm{tc}\, D^A(P)$ and $R^* = \mathrm{tc}(R) \cup \{(v, v): v \in V\}$. For the sake of simplicity, we shall denote an arc diagram by a digraph $D^A(P)$ in which certain arcs (drawn in solid lines) are labelled by the poset elements in the way which preserves the relation between them along the paths in $D^A(P)$. Let us denote $S = R - \phi(P)$. An arc $a \in \phi(P)$ is a *poset arc* and otherwise a is called a *dummy arc* (and drawn in dotted lines). A path π in $D^A(P)$ is a *poset path* if it consists entirely of poset arcs. For the sake of completeness we define also the *vertex diagram* (known also as a *Hasse diagram*) of P to be a digraph $D^H(P) = (P, A)$ in which $(p, q) \in A$ if and only if q covers p in P. (Note that, since vertex diagrams contain no parallel arcs, we can drop out incidence mappings from their definition.) Figure 1 shows vertex and arc diagrams of some sample posets.

Arc diagrams of posets are best know in network analysis as event or PERT networks, where they are used mainly for time analysis of projects whose precedence relations are represented by activity-one-arc networks. In poset theory however, arc diagrams have not been widely accepted as poset representations. The main reason behind this is that, in contrast to the uniqueness of vertex diagrams, a poset may have an infinite number of arc diagrams and in fact there is no standard one. Moreover, finding an arc diagram with the minimum number of dummies (which may be blamed for non-popularity of arc diagrams) is an NP-complete problem in general.

Recently, it was demonstrated by Möhring [3] that certain hard combinatorial problems can be efficiently solved on comparability graphs by using arc diagrams of corresponding posets. For the jump number problem, arc diagrams have been

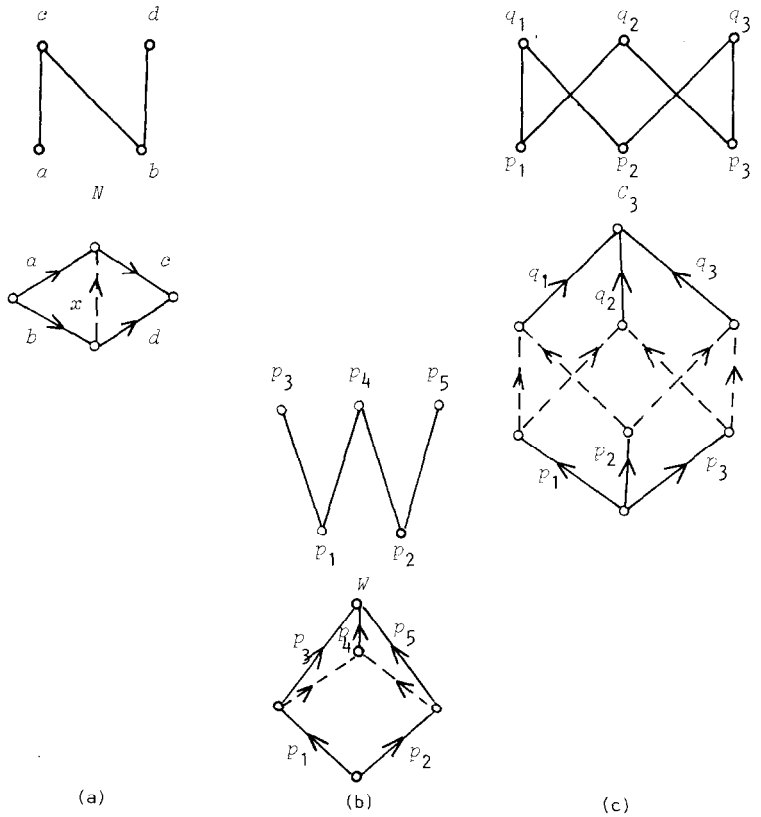

Fig. 1. Vertex and arc diagrams of some posets.

used in [6] to simplify the solution and derive the number of jumps in N-free posets. (N-free posets have arc diagrams with no dummy arcs, see Section 4 in [7].) Then, this approach has been also used in [7] to attack the problem for arbitrary posets. Finally in [8], a new proper subfamily of linear extensions has been defined on an arc diagram of a poset that contains an optimal solution and significantly reduces the search space. The result of [8] have been reformulated and improved in [9], where considerations are carried on directly on posets.

The purpose of this paper is to present an implementation of the results of [8] and [9], and discuss its complexity.

2. Greedy paths and linear extensions

We assume in the sequel that a poset P is represented by an arc diagram $D^A(P)$ which:

(i) has exactly one source and exactly one sink, and

(ii) for every dummy arc a, $\text{indeg}(h(a)) > 1$ and $\text{outdeg}(t(a)) > 1$.

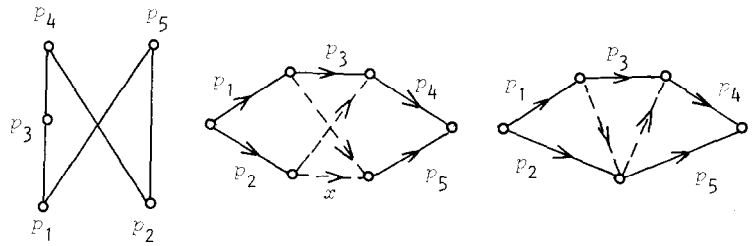

Fig. 2. Poset and its two compact arc representations.

An arc diagram $D^A(P)$ which satisfies conditions (i) and (ii) is called *compact*. Note that we weakened the conditions imposed on arc diagrams to be compact formulated in [8]. For instance, both arc diagrams shown in Fig. 2 are compact, whereas the first one is not compact in the sense defined in [8] since dummy arc x can be contracted.

An arbitrary arc diagram of a poset can be easily transformed to its compact form. Moreover, an arc diagram which has minimum number of vertices is compact and can be produced for every poset by a polynomial-time algorithm, we refer the reader to Section 4 of [7] for details.

A chain C of a poset P is *greedy* if there are no elements $p \in P - C$ and $q \in C$ such that $p < q$ and moreover for no r which covers $\sup C$ in P, the chain $C \cup \{r\}$ has this property. A linear extension $L = C_0 + C_1 + \cdots + C_s$ of P is *greedy* if C_i is a greedy chain in $P - \bigcup_{j<i} C_j$. It is easy to see that every poset has an optimal linear extension which is greedy. It was proved in [8] that for every poset the class of greedy linear extensions can be further restricted to the class of *semi-strongly greedy* ones which contains an optimal one. The purpose of this paper is to provide an efficient algorithm for generating an optimal semi-strongly greedy linear extension of a poset.

A semi-strongly greedy linear extension consists of two types of greedy chains which are naturally defined in the terms of arc diagrams. Let $D^A(P) = (V, R, t, h)$ be a compact arc diagram of a poset P. A path $\pi = (x_1, x_2, \ldots, x_k)$, $k \geq 1$ of $D^A(P)$ is called a *greedy path* if it has the following properties:

1. no vertex of π except $h(x_k)$ is a head of any arc $y \in R$, $y \neq x_j$ for every j, $0 \leq j \leq k$;
2. x_k is a poset arc; and
3. π cannot be extended to a path which satisfies 1 and 2, and has more poset arcs than π has.

Let C_π denote the sequence of poset arcs of the path π. It is easy to see that a greedy path π in a compact arc diagram $D^A(P)$ of P contains no dummy arcs. Moreover, the corresponding C_π is a greedy chain in P. On the other hand, every greedy chain C of P generates a greedy path in a compact arc diagram $D^A(P)$ – let $\pi(C)$ denote the corresponding path. Thus, we have a one-to-one correspondence between greedy chains of a poset and greedy paths in its compact

arc diagram. Therefore, we can use greedy paths and greedy chains interchangably.

A poset is *N-free* if its vertex diagram contains no induced subgraph isomorphic to the vertex digram N – see Fig. 1(a). N-free posets are precisely those which admit arc diagrams with no dummy arcs (see [6]). In what follows we assume that an N-free poset is represented by such a diagram.

If a poset P is N-free then every greedy linear extensions L of P is optimal (Rival [5]). In the other words, an optimal linear extension of an N-free poset P may begin with an arbitrary greedy path of $D^A(P)$. We now identify some greedy chains in arbitrary posets which share this property.

A greedy path π in a compact arc diagram $D^A(P)$ of a poset P is called *strongly greedy* if, additionally to conditions 1–3, π satisfies:

4. either (a) $h(\pi)$ is the sink of $D^A(P)$,
   or (b) $h(\pi)$ is the head of a poset arc b $(b \neq a_k)$ such that every path terminating with b has no vertex which is incident with a dummy arc.

In particular, π is strongly greedy if π is greedy and $h(\pi)$ terminates a greedy path π' $(\pi' \neq \pi)$ whose no vertex is a tail of a dummy arc. The diagram in Fig. 1(c) shows that not every poset has a strongly greedy path. The main property of strongly greedy paths is described in the following theorem.

Theorem 1 [8]. *If $D^A(P)$ admits a strongly greedy path π then there exists an optimal linear extension of P which begins with C_π.*

We proved in fact that for a strongly greedy path π in $D^A(P)$, every greedy linear extension L of P can be transformed to another linear extension L^* of P which begins with C_π and L^* has no more jumps than L. The examples in [8] show that such a transformation may result in decreasing the number of jumps in linear extensions. Since every N-free poset P admits a strongly greedy path π and a subposet $P - C_\pi$ of P is also N-free, Theorem 1 provides another proof of the formula for the jump number of N-free posets, see [6] and [8].

If $D^A(P)$ contains no strongly greedy paths then $D^A(P)$ contains greedy paths which in a certain sense are better than the other. Note that if π is a greedy path and π contains a vertex v which is a tail of at least one dummy arc but not a head of any dummy arc then taking C_π to a linear extension of P results in removing from $D^A(P)$ (together with π) all dummy arcs incident with v. More precisely, a greedy path π is called *semi-strongly greedy* if:

4′. π contains a vertex which is a tail of a dummy arc but not a head of any dummy arc.

For instance the path (b, d) in Fig. 1(a) and the paths (p_1), (p_2) and (p_3) in Fig. 1(c) are semi-strongly greedy. Note that all strongly greedy paths in diagrams of Figs. 1(a) and (b) are also semi-strongly greedy. An arc diagram may not contain a semi-strongly greedy path, e.g. when it has no dummy arcs. It is

however easy to show that every arc diagram which has no strongly greedy paths contains a semi-strongly greedy path. A formal proof of this fact given in [8] is based on elementary properties of acyclic digraphs formed by dummy arcs in compact arc diagrams.

A counterpart of Theorem 1 for semi-strongly greedy paths has a weaker form and says that in the absence of strongly greedy paths in $D^A(P)$, a good candidate to begin an optimal linear extension is one of the semi-strongly greedy paths in $D^A(P)$. Formally we have

Theorem 2 [8]. *If an arc diagram $D^A(P)$ of P contains no strongly greedy paths then P has an optimal linear extension which begins with a semi-strongly greedy path.*

Theorems 1 and 2 guarantee that every poset has an optimal linear extension $L = C_0 + C_1 + \cdots + C_s$, herafter called *semi-strongly greedy* such that each chain C_i is strongly greedy in $P_i = P - \bigcup_{j<i} C_i$ or semi-strongly greedy in P_i if P_i has no strongly greedy chains. In the next section we present an algorithm which finds an optimal linear extension of a poset among its semi-strongly greedy ones.

We conclude this section with a lower bound to the jump number which is also suggested by arc diagrams.

Observation 1. *If $D^A(P) = (V, R, t, h)$ is an arc diagram of a poset P then*

$$\sum_{v \in V} \max\{0, \operatorname{index}_P(v) - 1\} \leq s(P), \tag{1}$$

where $\operatorname{indeg}_P(v)$ *is the number of poset arcs coming into* v.

Proof. Note first that if $\operatorname{indeg}_P(v) > 1$ $(v \in V)$ then $\operatorname{indeg}_P(v)$ cannot be decreased by any greedy path which does not terminate at v. On the other hand, the removal of a greedy path from $D^A(P)$ that does not contain v results in a compact arc diagram in which $\operatorname{indeg}_P(v)$ is unchanged. Hence, only a greedy path terminating at v may reduce $\operatorname{indeg}_P(v)$ and each such a path contributes one to the jump number of P. This proves the bound. □

The bound (1) can be further improved by adding 1 to the left hand side if, after removing strongly greedy paths from $D^A(P)$, the reduced diagram admits only semi-strongly greedy paths π whose terminal vertices $h(\pi)$ are of total indegree 1 (in this case, the only other arcs incident with $h(\pi)$ are dummies which go out of $h(\pi)$).

3. An algorithm

A search method for finding for a poset a semi-strongly greedy linear extension with the minimum number of jumps is implemented in procedure OPTLINEXT.

A poset P is given by its compact arc diagram $D = (V, R, t, h)$. Such a diagram can be produced by a polynomial-time algorithm which constructs arc diagrams with minimum number of vertices, see Section 4 of [7] for a survey of construction methods of arc diagrams. An arbitrary arc diagram can be easily transformed to a compact form by contracting dummy arc a for which $\text{indeg}(h(a)) = 1$. Such a transformation can be applied in the algorithm whenever a current diagram (which is compact) is reduced by the removal of a (strongly or semi-strongly) greedy path. The diagram is represented by adjacency lists of arcs coming to and going out of each vertex. It is convenient to partition each list into two sublists of poset and dummy arcs, respectively.

Greedy paths (strongly and semi-strongly) in D can be easily identified by inspecting the vertices of D in a topological order, that is, when visiting vertex v we can be sure that all its predecessors have been already visited. To keep track of greedy paths, we define two queues S and W which store strongly and semi-stongly greedy paths in the current diagram. The paths in S and W are uniquely represented by their end-arcs. Statement 1 in the procedure is assumed to find all these quantities in the given arc diagram and then procedure REMOVE takes care of updating S and W. The number of jumps in a linear extension L of P is denoted by $s(L, P)$.

procedure OPTLINEXT(arc diagram: D, linear extension: **var** L^*);
  {The procedure finds an optimal semi-strongly greedy linear extension L^* of a poset p represented by its arc diagram D.}
procedure REMOVE(α, **var** D, S, W);
  {This procedure removes greedy path α from arc diagram D, transforms $D - \alpha$ into a compact form and updates queues S and W of strongly and semi-strongly greedy paths in the current arc diagram. Implementation details of this procedure are left to the reader.}
procedure SUBLINEXT(π, L, D, S, W);
  {The procedure extends L with a semi-strongly greedy path π removes π from D, extends L with strongly greedy paths which may result and then attempts to repeat this process.}
begin
  augment L with π;
  REMOVE(π, D, S, W);
  while $S \neq \emptyset$ **do begin**
    $\delta \leftarrow S$; augment L with δ; REMOVE(δ, D, S, W) **end**;
  if $D \neq \emptyset$ **then**
    for every ρ in W **do**
      SUBLINEXT(ρ, L, D, S, W)
  else if $s(L, P) < r$ **then begin** $r \leftarrow s(L, P)$; $L^* \leftarrow L$ **end**
end {SUBLINEXT};
begin
  $L \leftarrow S \leftarrow W \leftarrow \emptyset$; {$L$, S, W are handled as queues} $r \leftarrow \infty$;

1. **for** $v \in V(D)$ **do** {vertices in $V(D)$ are in topological ordering}
   update S and W;
2. **while** $S \neq \emptyset$ **do begin**
   $\delta \leftarrow S$; augment L with δ; REMOVE(δ, D, S, W) **end**;
3. **if** D contains no arc **then begin** $r \leftarrow s(L, P)$; $L^* \leftarrow L$ **end**
   else for every π in W **do**
   SUBLINEXT(π, L, D, S, w)

end.

Theorem 1 guarantees that every strongly greedy path of D can be removed from D and added to L as a part of an optimal linear extension. Statement 2 of OPTLINEXT takes care of this step. The removal of a greedy path α from an arc diagram D is done by procedure REMOVE which, if necessary, transforms also $D - \alpha$ into a compact form and updates queues S and W. We leave implementation details of the procedure REMOVE to the reader.

The removal of strongly greedy paths from D may lead to exhausting all poset elements (arcs) of the diagram and this can happen not only for N-free posets. Otherwise, procedure OPTLINEXT attempts to build an optimal linear extension starting with an arbitrary semi-strongly greedy path π of D. This is done by the call of procedure SUBLINEXT(π, L, D, S, W) which: augments partial extension L with π, removes π from D and repeats recursively this process. To this end, procedure REMOVE is called, L is extended with strongly greedy paths of the reduced diagram and finally, if the current diagram D is non-empty, procedure SUBLINEXT is called again.

Regarding the time complexity of the algorithm note that each sequence of recursive calls of SUBLINEXT is successful in the sense that a semi-strongly greedy linear extension of the poset is obtained. Hence, it is easy to see that the time spent by procedure OPTLINEXT on producing one complete semi-strongly greedy linear extension of the given poset is bounded by $0(n + k)$, since one has only to reduce the diagram to the empty set by removing greedy paths. A poset with k dummy arcs can have at most k semi-strongly greedy paths, and the removal of a semi-strongly greedy path reduces the number of dummy arcs by at least one. Hence, there are at most $k!$ semi-strongly greedy linear extensions produced by the procedure and thus, total time complexity of the algorithm is bounded by $0(k!\,(n + k))$. This estimation is very rough since it assumes that no dummy arc is reduced from the diagram by strongly greedy paths and that exactly one dummy arc is involved with a semi-strongly greedy path.

The performance of the algorithm implemented in OPTLINEXT can be improved at least in two directions. First, it is not necessary to go forward if a partial linear extension L has already the same number of chains as L^*, the best one found so far. Secondly, the algorithm may be terminated when a linear extension is found which has the number of jumps equal to the lower bound (1).

As an illustration, let us apply the algorithm to the arc diagram D of the

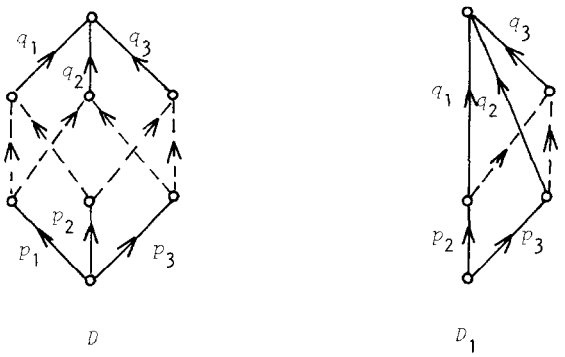

Fig. 3. Illustration of the algorithm.

3-crown which is hown in Fig. 1(c). D has no strongly greedy path and contains three semi-strongly greedy paths (p_1), (p_2), and (p_3). The diagram is symmetric with respect to each of these paths, what is not however recognized by the algorithm. We first choose the path (p_1). Fig. 3(b) shows the compact arc diagram D_1 of the poset reduced by removing (p_1). Note that this reduction results in removing four dummy arcs from the diagram. D_1 contains two strongly greedy paths (p_2, q_1) and (p_3, q_2) since q_1 and q_2 are maximal. Removing any of them we obtain a diagram which contains no dummy arcs. Therefore, the algorithm will produce only 3 complete linear extensions (instead of 6!), all with the same number of jumps. Note that in this case left-hand side of (1) can be increased by 1 since all semi-strongly greedy paths of the diagram terminate at vertices of indegree 1. Therefore, the algorithm may terminate after producing the first linear extension.

4. Conclusions

We have presented the algorithm for solving the jump number problem which is linear in the poset size and factorial in k, the number of dummy arcs in arc representations of posets. Therefore it is a linear-time algorithm in the class of posets with bounded k, which may be considered as a degree of non N-freeness of posets. The algorithm works directly on an arc diagram of a poset and searches only a portion of the solution space which consists of semi-strongly greedy linear extensions.

There exist in the literature two other algorithms for the jump number problem which are polynomial in certain restricted classes of posets. Colbourn and Pulleyblank [1] presented a dynamic programming algorithm which works on an arbitrary chain partition of a poset and finds the jump number in $0(n^m m^2)$ time, where n is the poset size and m is the number of chains in a partition. In

particular, one may apply this algorithm to a partition with the minimum number of chains which can be found efficiently by a bipartite matching algorithm.

Another algorithm, presented by Habib and Möhring [2], makes use of the substitution decomposition of posets into prime (i.e. indecomposable) posets and solves the problem in time which depends polynomially on the poset size but is a highly non-polynomial function in l, the maximum size of a prime poset. Unfortunately, the approach of Habib and Möhring requires the knowledge of the jump number of all prime posets of size at most l.

It is difficult to compare our algorithm with the two just mentioned above since there are no evident relations between the poset parameters which occur in the time complexity formulae of these algorithms. (We note only that a poset with bounded m can have an arbitrary k and conversally.) As an advantage of our approach over the two others we can list: very little preprocessing of the data, the existence of deterministic steps which depend on augmenting linear extensions with strongly greedy chains and, at last but not at least, its linear complexity in the poset size.

References

[1] C.J. Colbourn and W.R. Pulleyblank, Minimizing setups in ordered sets of fixed width, Order 1 (1985) 225–229.

[2] M. Habib and R.H. Möhring, On some complexity properties of N-free posets and posets with bounded decomposition diameter, Discrete Math., 63 (1987) 157–182.

[3] R.H. Möhring, Algorithmic aspects of comparability graphs and interval graphs, in: I. Rival (ed.) Graphs and Order (D. Reidel, Dordrecht, 1985) 41–101.

[4] W.R. Pulleyblank, On minimizing setups in precedence-constrained scheduling, Discrete Appl. Math., to appear.

[5] I. Rival, Optimal linear extensions by interchanging chains, Proc. Amer. Math. Soc. 89 (1983) 387–394.

[6] M.M. Sysło, Minimizing the jump number for partially-ordered sets: a graph-theoretic approach, Order 1 (1984) 7–19.

[7] M.M. Sysło, A graph–theoretic approach to the jump-number problem, in: I. Rival (ed.), Graphs and Order (D. Reidel, Dordrecht, 1985) 185–215.

[8] M.M. Sysło, Minimizing the jump number for partially-ordered sets: a graph-theoretic approach, II, Discrete Math. 63 (1987) 279–295.

[9] M.M. Sysło, On some new types of greedy chains and greedy linear extensions of partially ordered sets, submitted to Order.