On Identifying in Polynomial Time Violated Subtour Elimination and Precedence Forcing Constraints for the Sequential Ordering Problem

N. Ascheuer[1], L. Escudero[2],
M. Groetschel[1], and M. Stoer[1]

[1] Universität Augsburg, F.R. Germany
[2] IBM T.J. Watson Research Center, Yorktwon Heights, NY

Abstract

In this paper we consider the so-called Sequential Ordering Problem (SOP) introduced in [4] that has a broad range of applications, mainly, in production planning for manufacturing systems. Given a set of nodes, the SOP consists of finding a Hamiltonian path, such that precedence relationships among the nodes are satisfied and a given linear function is minimized. In [4] an efficient inexact algorithm for obtaining feasible solutions is described. Here, we present a strong formulation of the problem and procedures for identifying subtour elimination constraints and precedence forcing constraints that are most violated by the optimal solution of a LP relaxation of the original problem. The complexity of these separation procedures is $O(n^4)$ and $O(rn^3)$, respectively, where n denotes the number of nodes and r is the number of precedence relationships.

1. Problem Definition

Let $\mathscr{H}$ denote a Hamiltonian path through a given set of nodes, say V. Let $i \rightarrow j$ mean that node i is immediately ordered before node j in a given $\mathscr{H}$. Let the acyclic directed graph (or digraph) $P = (V,R)$, where R is the set of directed arcs, such that $(i,j) \in R$ means that node i has to be ordered (immediately or not) before node j in any feasible $\mathscr{H}$. We clearly may assume that P is transitively closed.

Let the complete digraph $D_n = (V,A_n)$ and the matrix $C = \{c_{i,j}\}$ be such that $c_{i,j}$ for $(i,j) \in A_n$ gives the cost associated with $i \rightarrow j$. Define

$$\overleftarrow{R} := \{(j,i) | (i,j) \in R\} \tag{1.1}$$

$$\overrightarrow{R} := \{(i,k) \mid \exists j \text{ s.t. } (i,j),(j,k) \in R\}$$

and let us define the digraph D=(V,A) by setting

$$A := A_n \setminus (\overrightarrow{R} \cup \overleftarrow{R}) \tag{1.2}$$

It is obvious that no feasible $\mathcal{H}$ contains an arc from $\vec{R} \bigcup \overleftarrow{R}$. The *Sequential Ordering Problem* (SOP) consists of finding a Hamiltonian path $\mathcal{H}$ with minimum weight in digraph D, such that the precedence relationships given by R are not violated.

The SOP has a broad application field. An obvious application is the ATSP with fixed city-origin and city-destination. The sequencing of the cities may require to satisfy some precedence relationships. Another typical application [6] is the sequencing of the operations' execution, mainly in scheduling manufacturing systems. See in [18] an interesting application and an inexact algorithm for the related STSP with precedence relationships.

The paper is organized as follows. Section 2 presents our favorite 0-1 model for the SOP. It also outlines the procedure for obtaining an optimal solution. It is based on an iterative tightening of the relaxations of the Subtour Elimination Constraints (SECs) and Precedence Forcing Constraints (PFCs). Sections 3 and 4 are devoted to the procedures for identifying violated SECs and PFCs, respectively. These procedures have complexity $O(n^4)$ and $O(rn^3)$, respectively, where $n \equiv |V|$ and $r \equiv |R|$. Finally, we offer some conclusions and outline future work.

2. The 0-1 Model

Let $x_{i,j}$ be a 0-1 variable such that $x_{i,j} = 1$ means that $i \rightarrow j$ (in a feasible $\mathcal{H}$) and, otherwise, it is zero. Since $x_{j,i} = 0$ must be satisfied for all $(j,i) \in R \bigcup R$ we can drop these variables. Incidentally, a key step in any practical implementation of this approach is the preprocessing procedure for reducing the cardinality of set A; i.e., arc (i,j) should be deleted from A if it has been detected that $i \not\rightarrow j$ in any feasible $\mathcal{H}$, or in any better solution than the incumbent one (if any). An efficient algorithm for the SOP preprocessing is described in [4].

There are several 0-1 models for the SOP; see, e.g., [1,4]. Our favorite 0-1 model is as follows

$$\min \; c^t x \tag{2.1}$$

subject to

$$x(A) = n - 1 \tag{2.2}$$

$$x(\delta^-(v)) \leq 1 \text{ for all } v \in V \tag{2.3}$$

$$x(\delta^+(v)) \leq 1 \text{ for all } v \in V \tag{2.4}$$

It is obvious that no feasible $\mathcal{H}$ contains an arc from $\dot{R} \bigcup \dot{R}$. The *Sequential Ordering Problem* (SOP) consists of finding a Hamiltonian path $\mathcal{H}$ with minimum weight in digraph D, such that the precedence relationships given by R are not violated.

The SOP has a broad application field. An obvious application is the ATSP with fixed city-origin and city-destination. The sequencing of the cities may require to satisfy some precedence relationships. Another typical application [6] is the sequencing of the operations' execution, mainly in scheduling manufacturing systems. See in [18] an interesting application and an inexact algorithm for the related STSP with precedence relationships.

The paper is organized as follows. Section 2 presents our favorite 0-1 model for the SOP. It also outlines the procedure for obtaining an optimal solution. It is based on an iterative tightening of the relaxations of the Subtour Elimination Constraints (SECs) and Precedence Forcing Constraints (PFCs). Sections 3 and 4 are devoted to the procedures for identifying violated SECs and PFCs, respectively. These procedures have complexity $O(n^4)$ and $O(rn^3)$, respectively, where $n \equiv |V|$ and $r \equiv |R|$. Finally, we offer some conclusions and outline future work.

2. The 0-1 Model

Let $x_{i,j}$ be a 0-1 variable such that $x_{i,j} = 1$ means that $i \rightarrow j$ (in a feasible $\mathcal{H}$) and, otherwise, it is zero. Since $x_{j,i} = 0$ must be satisfied for all $(j,i) \in \vec{R} \bigcup \overleftarrow{R}$ we can drop these variables. Incidentally, a key step in any practical implementation of this approach is the preprocessing procedure for reducing the cardinality of set A; i.e., arc (i,j) should be deleted from A if it has been detected that $i \not\rightarrow j$ in any feasible $\mathcal{H}$, or in any better solution than the incumbent one (if any). An efficient algorithm for the SOP preprocessing is described in [4].

There are several 0-1 models for the SOP; see, e.g., [1,4]. Our favorite 0-1 model is as follows

$$\min\ c^t x \tag{2.1}$$

subject to

$$x(A) = n - 1 \tag{2.2}$$

$$x(\delta^-(v)) \leq 1 \text{ for all } v \in V \tag{2.3}$$

$$x(\delta^+(v)) \leq 1 \text{ for all } v \in V \tag{2.4}$$

$$0 \leq x_{i,j} \leq 1 \text{ for all } (i,j) \in A \tag{2.5}$$

$$0 \le x_{ij} \le 1 \text{ for all } (i,j) \in A \tag{2.5}$$

$$x(A(W)) \le |W| - 1 \text{ for all } W \subset V,\ 2 \le |W| \le n-1 \tag{2.6}$$

$$x_{ij} \in \{0;1\} \text{ for all } (i,j) \in A \tag{2.7}$$

$$x((j{:}W)) + x(A(W)) + x((W{:}i)) \le |W| \quad \text{for all } (j,i) \in \overleftarrow{R} \text{ and all } W \subset V \backslash \{i,j\},\ W \neq \emptyset \tag{2.8}$$

where

$$x(F) = \sum_{(i,j) \in F} x_{ij},\quad \delta^-(v) = \{(w,v) \in A\},\quad \delta^+(v) = \{(v,w) \in A\},$$

$$A(W) = \{(i,j) \in A \mid i,j \in W\},\quad (j{:}W) = \{(j,w) \in A \mid w \in W\} \text{ and } (W{:}i) = \{(w,i) \in A \mid w \in W.$$

We restrict c in (2.1) to the coordinates $A \subseteq A_n$. Constraint (2.2) defines the number of arcs in any $\mathscr{H}$. Constraints (2.3) (resp., (2.4)) prevent that more than one node is sequenced immediately before (resp., after) any other node. Constraints (2.6) are the Subtour Elimination Constraints SECs (see e.g. [14]). Constraints (2.8) are the Precedence Forcing Constraints PFCs.

Note that (2.1)-(2.7) can be easily converted into the classical Asymmetric Traveling Salesman Problem (ATSP). Let the AP-like LP problem (2.1)-(2.5) be named LPAP.

The basic methodology of the exact algorithm that we are using for solving (2.1)-(2.8) has the following main steps: (1) Obtaining an initial feasible solution. The inexact algorithm described in [4,5] can be used. (2) Optimizing a relaxation of the original problem. We start with LPAP. (3) Reduced cost based variable fixing coupled with an implication fixing analysis. We use the preprocessing procedure described in [4]. (4) Cutting plane generation by identification of a most violated SEC (Section 3) and PFC (Section 4). We also consider (see [1,10,12]) some types of lifted SEC's and PFC's. (5) The new constraints are added, and non-active constraints previously appended are deleted from the current LP relaxation. (6) The dual-based optimization of the new problem is performed. (7) If there is not any violated constraint except (2.7), a branch-and-cut phase is executed. We should emphasize the synergetic effect of combining the procedures for obtaining initial feasible solutions, performing reduced cost fixing and implications, and generating violated constraints. The overall framework of this methodology has been previously described and its computational results have been extensively analyzed in [3,6,11,13,14,16,17] among others. Elsewhere [1] we report our provisional computational experience on some real-life problems. In [8] a different approach for getting strong lower bounds for the ATSP imbedded in the SOP is described. See related work in [2,7].

3. Identifying Violated Subtour Elimination Constraints

The problem of determining a violated subtour elimination constraint (2.6) is reduced (in the obvious way) to the same problem for the symmetric case. We outline here the procedure for completeness.

Let us assume that $\bar{x}$ is the optimal solution of problem (2.1)-(2.5) where some constraints (2.6) and (2.8) may have been appended. We now consider the values $\bar{x}_{i,j}$ as capacities of the arcs $(i,j) \in A$. The goal consists of identifying a node set $W \subset V$ such that the related constraint (2.6) is a most violated SEC (if any). Then, the set W will be

$$W = arg.max\{\bar{x}(A(W')) - |W'| + 1 > 0\} \text{ for all } W' \subset V, 2 \leq |W'| \leq n-1. \qquad (3.1)$$

For this purpose we will construct a symmetric directed graph, say $\bar{D}$, for which a certain mincut problem is to be solved. But, first, let us introduce additional notation. Let

$$\bar{y}_v = \bar{x}(\delta^-(v)) + \bar{x}(\delta^+(v)) \text{ for all } v \in V. \qquad (3.2)$$

Next, we define an (auxiliary) digraph, say $D_0 = (V_0, A_0)$, as follows:

$$V_0 = V \cup \{0\}, \text{ where 0 is a new node }, \qquad (3.3)$$

$$A_0 = A \cup \{ (0,v) \mid v \in V\} \cup \{(j,i) \mid (i,j) \in A \text{ and } (j,i) \notin A\}.$$

(That is, we add to V a node 0 (the "source"), and we add to A arcs $(0,v)$ for all $v \in V$ and all reverse arcs but do not create parallel arcs). Let us define the following capacities for D_0

$$c^0_{0,v} := 1 - \frac{1}{2}\bar{y}_v \text{ for all } v \in V \qquad (3.4)$$

(Note that $c^0_{0,v} \geq 0$ for all $v \in V$, where $c^0_{0,v} = 0$ whenever the related constraints (2.3) and (2.4) are satisfied as equalities). Further, we set

$$c^0_{i,j} := c^0_{j,i} := \frac{1}{2}(\bar{x}_{i,j} + \bar{x}_{j,i}) \text{ for all } (i,j) \in A_0 \backslash \{(0,v) | v \in V\} \qquad (3.5)$$

(If $(i,j) \notin A$ (resp., $(j,i) \notin A$) we set $\bar{x}_{i,j} := 0$ (resp., $\bar{x}_{j,i} := 0$).)

Let $\delta_0(W)$ denote the minimum capacity cut in $D_0 = (V_0, A_0)$ with respect to the capacities (3.4) and (3.5), such that $0 \notin W$, and let $c(\delta_0(W))$ denote the associated capacity. It follows from our construction that if

$$c(\delta_0^-(W)) \geq 1 \tag{3.6}$$

is satisfied then there is no SEC violated by the current LP solution $\bar{x}$. Otherwise, the SEC induced by W is a most violated constraint (2.6).

We now construct a symmetric directed graph, say $\bar{D} := (V_0, \bar{A})$, related to digraph $D_0 = (V_0, A_0)$, by setting

$$\bar{A} := A_0 \cup \{(v,0) | v \in V\} \tag{3.7}$$

with capacities

$$\bar{c}_{i,j} := \bar{c}_{j,i} := c^0_{i,j} = c^0_{j,i} \text{ for all } (i,j) \in A_0 \tag{3.8}$$

$$\bar{c}_{v,0} := c^0_{0,v} \text{ for all } v \in V \tag{3.9}$$

($\bar{D}$ is a symmetric digraph since if any two nodes, say u and v are linked by an arc then both arcs (u,v) and (v,u) are in the digraph). The underlying undirected graph $G = (V_0, E)$ of $\bar{D} = (V_0, \bar{A})$ is defined by

$$E = \{ ij \,|\, (i,j) \in \bar{A} \text{ and } (j,i) \in \bar{A} \} \tag{3.10}$$

with capacities

$$\hat{c}_{ij} := \bar{c}_{i,j} = \bar{c}_{j,i} \text{ for all } ij \in E \tag{3.11}$$

It has the property that for each $W \subset V$,

$$\hat{c}(\delta(W)) = c(\delta_0^-(W)) \tag{3.12}$$

where $\delta(W) = \{ ij \in E \,|\, i \in V \setminus W, j \in W \}$.

Thus, a minimum capacity cut $\delta(W)$ in G corresponds to a minimum capacity cut $\delta_0^-(W)$ and vice versa. Such a cut $\delta(W)$ can be obtained with the Gomory-Hu procedure. Padberg and Rinaldi [15] describe a practically efficient version of this procedure.

The main loop of the algorithm requires at most n major iterations. At any major iteration two nodes at least are "contracted" or "shrunk" into a a single node. Two main steps are executed at each major iteration: first, a series of tests are designed to shrink as many nodes as possible; and, second, a max-flow algorithm is executed by selecting any two nodes of the shrunk graph as nodes "source" and "sink". The algorithm described by Goldberg and Tarjan [9] is used in the second step; its complexity is $p(n) = O(mn \log(n^2/m))$, where $m = |E|$. The tests that are performed in the first step have a complexity not worse than $p(n)$. Then, the overall complexity is $O(n^4)$. Although this is not better than the complexity of other published algorithms, the "shrinking" mechanism suggests a better efficiency in practice; this is confirmed by extensive computational experience reported in [16,17]. Since the algorithm obtains successively a series of better cuts till an optimal one is reached, we may generate the related induced SECs if they are violated, instead of using only a 'best' one.

4. Identifying Violated Precedence Forcing Constraints

We now describe how one can decide whether a given point satisfies all precedence forcing constraints, and if it does not, how one can find a (most) violated PFC (2.8). Let us assume that a vector $\bar{x}$, for practical purposes the current LP-solution, is given. For every $(j,i) \in R$ we do the following. Construct a new digraph $D_{j,i} = (V_{j,i}, A_{j,i})$ from D=(V,A) by deleting some arcs (those which never appear in (2.8)) and by shrinking the nodes i and j. Formally, set

$$V_{j,i} = V \setminus \{i,j\} \cup \{v_{j,i}\} \tag{4.1}$$

where $v_{j,i}$ is a new 'special' node representing j and i, and

$$\begin{aligned} A_{j,i} = \{(k,l) \mid (k,l) \in A,\ k \notin \{i,j\},\ l \notin \{i,j\}\} \\ \cup \{(v_{j,i}, l) \mid (j,l) \in A,\ l \notin \{i,j\}\} \\ \cup \{(k, v_{j,i}) \mid (k,i) \in A,\ k \notin \{i,j\}\} \end{aligned} \tag{4.2}$$

Figure 1 shows the digraph $D_{j,i}$ induced by shrinking the nodes i and j in digraph D. Note that we throw out every arc directed into j and every arc directed from i; by definition, $(j,i) \notin A$. Every arc $a \in A_{j,i}$ gets a capacity $\hat{x}_a$ that is nothing but the value $\bar{x}_a$ of the corresponding arc a in $D = (V,A)$. Note that a PFC (2.8) for the old digraph D can be written as

$$x(A_{j,i}(\hat{W})) \leq |\hat{W}| - 1 \tag{4.3}$$

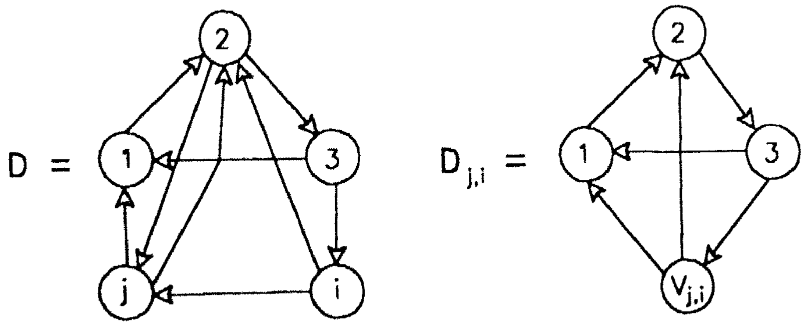

Figure 1. Original and shrinking graphs of precedence relationships

with respect to the new digraph $D_{j,i}$, where $\hat{W} = W \bigcup \{v_{j,i}\}$.

Our goal is to find a set $\hat{W}$ in $D_{j,i}$ with $v_{j,i} \in \hat{W}$ and $\hat{W} > 2$ such that the related constraint (4.3) forcing the precedence relation (i,j) is a most violated one or to show that no violated constraint of that type exists. To do this, we create an auxiliary digraph D_0 from $D_{j,i}$ (exactly in the same way as D_0 was constructed from D in Section 3) and determine a minimum capacity cut in D_0 that separates 0 from $v_{j,i}$. This can be done by applying any max-flow algorithm. Suppose $\delta_0(\hat{W})$ is such a cut with $0 \notin \hat{W}$ and $v_{j,i} \in \hat{W}$, and let $\gamma = c^{v_{j,i}}(\delta_0(\hat{W}))$ be its capacity. One can show that the following holds.

If $\gamma \geq 1$ then no PFC related to (i,j) is violated by $\bar{x}$. If $\gamma < 1$ then $\hat{W} \geq 2$ and letting $W^* := \hat{W} \setminus \{v_{j,i}\}$ the PFC

$$x((j:W^*)) + x(A(W^*)) + x((W^*:i)) \leq |W^*| \tag{4.4}$$

is violated by $\bar{x}$, in fact, this is a most violated constraint forcing the precedence relation (i,j).

For obtaining a minimum capacity cut $\delta_0(\hat{W})$ separating 0 from $v_{j,i}$ we use the algorithm described in [9]; its complexity is $O(n^3)$.

For illustrative purposes let us consider the complete digraph $D_n = (V, A_n)$ for $n = 7$ and the matrix C shown in Table 1. The set R will be $R = \{(1,j)\} \bigcup \{(i,5)\}$ for $j = 4,5,6,7$ and $i = 1,4,6,7$. Our inexact algorithm [4] gives the feasible solution $1 \rightarrow 4 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 3$. The value of the objective function is $\bar{z} = 21.25$. The LPAP optimal value is $\underline{z} = 18.00$ and, then, there is a 14.11% gap. The current implementation of our (exact) algorithm adds 8 cuts. The strongest LP lower bound is $\underline{z} = 20.75$ (then the gap is 2.29%). A branch-and-cut phase requires two nodes to prove the optimality of the initial solution. We currently have a PC and a mainframe implementation of the algorithm described above. The PC-version solves problems up to $n = 100$ nodes and $r = 280$ precedence relationships in less than two and a half CPU hours. The mainframe version does this in a few seconds.

\n2 \ n1\	1	2	3	4	5	6	7
1	–	1.00	2.00	0.75	0.00	3.00	1.00
2	4.00	–	5.00	3.25	4.00	6.00	0.00
3	7.00	8.00	–	5.50	7.00	9.00	8.00
4	2.75	2.50	2.25	–	2.75	5.25	2.50
5	0.00	1.00	2.00	0.75	–	3.00	1.00
6	10.00	11.00	12.00	10.75	10.00	–	11.00
7	4.00	0.00	5.00	3.25	4.00	6.00	–

Table 1. Matrix C

Conclusions

In this work we have presented a 0-1 model for the Sequential Ordering Problem. It is stronger than the model introduced in [4,5]. We have also outlined an (exact) LP-based algorithm. An extensive computational study is in progress. More theoretical work is required mainly for identifying in reasonable polynomial time violated lifted SECs and PFCs. In any case, we believe that the LP-based approach is a most promising way to get strong lower bounds on optimal solutions for this type of problems.

Acknowledgements

We appreciate Manfred Padberg's and Giovanni Rinaldi's comments.

References

[1] N. Ascheuer, L.F. Escudero, M. Groetschel, and M. Stoer, On LP bounds for the sequential ordering problem (in preparation).

[2] E. Balas and M. Fischetti, A lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets, MSRR-556, GSIA, Carnegie Mellon University, Pittsburg, PA.

[3] H. Crowder and M. Padberg, Solving large-scale symmetric traveling salesman problems to optimality, **Management Sciences** 26 (1980) 495-509.

[4] L.F. Escudero, An inexact algorithm for the sequential ordering problem, **European J. of Operational Research** 37 (1988) 236-253.

[5] L.F. Escudero, On the implementation of an algorithm for improving a solution to the sequential ordering problem, **Trabajos de Investigacion-Operativa** 3 (1988) 117-140.

[6] L.F. Escudero, A production planning problem in FMS, **Annals of Operations Research** 17 (1989) 69-104.

[7] M. Fischetti, Facets of the asymmetric traveling salesman polytope, **Mathematics of Operations Research** (1989, to appear).

[8] M. Fischetti and P. Toth, An additive bounding procedure for combinatorial optimization problems, **Operations Research** 37 (1989) 319-328.

[9] A.V. Goldberg and R.E. Tarjan, A new approach to the maximum flow problem, **Journal of ACM** 35 (1988) 921-940.

[10] M. Groetschel, Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme (Hain, Meisenheim am Glan, 1977).

[11] M. Groetschel, M. Juenger and G. Reinelt, A cutting plane algorithm for the linear ordering problem, **Operations Research** 34 (1984) 1195-1220.

[12] M. Groetschel and M. Padberg, Lineare Charakterisierungen von Travelling Salesman Problemen, **Z. Operations Research** 21 (1977) 33-64.

[13] K. Hoffman and M. Padberg, LP-based combinatorial problem solving, **Annals of Operations Research** 5 (1986) 145-194.

[14] M. Padberg and M. Groetschel, Polyhedral computations, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy-Kan and D.B. Shmoys (eds.), **The Traveling Salesman Problem, A guided tour of combinatorial optimization** (Wiley, NY, 1985) 251-360.

[15] M. Padberg and G. Rinaldi, On an efficient algorithm for the minimum capacity cut problem, Preprint, New York University, NY, 1987.

[16] M. Padberg and G. Rinaldi, Optimization of a 532-city symmetric traveling salesman problem, **Operations Research Letters** 6 (1987) 1-7.

[17] M. Padberg and G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, num. 319, Laboratoire d'Econometrie de l'Ecole Polytechnique, Paris, 1989.

[18] W. Pulleyblank and M. Timlin, Precedence constrained routing, ORSA/TIMS Joint National Meeting, New York, 1989.