TWO COMPUTATIONALLY DIFFICULT SET COVERING PROBLEMS THAT ARISE IN COMPUTING THE 1-WIDTH OF INCIDENCE MATRICES OF STEINER TRIPLE SYSTEMS

D. R. Fulkerson, et al

Wisconsin University

Prepared for:

Army Research Office-Durham
Office of Naval Research
National Science Foundation

February 1974
Two minimum cardinality set covering problems of similar structure are presented as difficult test problems for evaluating the computational efficiency of integer programming and set covering algorithms. The smaller problem has 117 constraints and 27 variables and the larger one, constructed by H. J. Ryser, has 330 constraints and 45 variables. The constraint matrices of the two set covering problems are incidence matrices of Steiner triple systems. An optimal solution to the problem that we were able to solve (the smaller one) gives some new information on the 1-widths of members of this class of (0,1)-matrices.
TWO COMPUTATIONALLY DIFFICULT SET COVERING PROBLEMS THAT ARISE IN COMPUTING THE 1-WIDTH OF INCIDENCE MATRICES OF STEINER TRIPLE SYSTEMS


This document has been approved for public release and sale; its distribution is unlimited.

MRC Technical Summary Report #1411
February 1974

Received November 15, 1973
Two minimum cardinality set covering problems of similar structure are presented as difficult test problems for evaluating the computational efficiency of integer programming and set covering algorithms. The smaller problem has 117 constraints and 27 variables and the larger one, constructed by H. J. Ryser, has 330 constraints and 45 variables. The constraint matrices of the two set covering problems are incidence matrices of Steiner triple systems. An optimal solution to the problem that we were able to solve (the smaller one) gives some new information on the 1-widths of members of this class of (0,1)-matrices.

Two Computationally Difficult Set-Covering Problems That Arise in Computing the 1-Width of Incidence Matrices of Steiner Triple Systems


1. Introduction. The purpose of this note is two-fold. First, we supply data for two integer programming (set covering) problems which we believe are computationally hard. Our experience indicates that optimal solutions to these problems are very tedious to compute and verify, even though they have far fewer variables than numerous solved problems in the literature. Thus these problems seem to be appropriate difficult test problems for evaluating the computational efficiency of integer programming and set covering algorithms.

The constraint matrices of the two set covering problems are incidence matrices of Steiner triple systems. An optimal solution to the problem that we were able to solve (the smaller one) gives some new information on the 1-widths of members of this particular class of \((0,1)\)-matrices. This new information is the second purpose of this note.

2. Origin of the Problems. The \(\alpha\)-width of a \((0,1)\)-matrix \(A\) is the minimum number of columns that can be selected from \(A\) so that all row sums of the resulting submatrix of \(A\) are at least \(\alpha\). Here \(\alpha\) is an integer parameter ranging from zero to the smallest row sum of the matrix \(A\). This notion was introduced and studied by Fulkerson and Ryser in a series of papers [3, 4, 5]. Henceforth we restrict attention to the case \(\alpha = 1\) and denote the 1-width of an \(m \times n\) \((0,1)\)-matrix \(A\) having no zero rows by \(w(A)\). The integer \(w(A)\) can thus be determined from an optimal solution to the set covering problem

\[
(2.1) \quad w(A) = \min_{\mathbf{x}} \frac{1}{n} \mathbf{x}, \quad \mathbf{Ax} \geq \frac{1}{m}, \quad \mathbf{x} \geq 0 \quad \text{and integral.}
\]
Here $1_k$ is the $k$-vector all of whose components are 1, and we are viewing $A$ as the incidence matrix of $m$ elements (rows) vs. $n$ subsets (columns) of the $m$-set. (Alternatively, and this point of view is perhaps more appropriate for the discussion to follow, we can view (2.1) as the problem of determining the least number of elements of an $n$-set required to represent all members of a family of $m$ subsets of the $n$-set.)

A Steiner triple system on $n$ elements is a pair $(S,T)$ where $S = \{1,2,\ldots,n\}$ and $T = \{S_1,\ldots,S_m\}$ is a family of triples (subsets of $S$ of cardinality three) such that every pair of elements of $S$ (every subset of $S$ of cardinality two) is contained in precisely one of the triples. It is well known that Steiner triple systems exist if and only if $n \geq 3$ and $n \equiv 1, 3 \pmod{6}$, in which case $m = \frac{n(n-1)}{6}$, and each element of $S$ appears in precisely $\frac{n-1}{2}$ of the triples. (Steiner triple systems are particular cases of combinatorial configurations known as balanced incomplete block designs, which are used in the statistical design of experiments. For example, if $n$ drugs are to be tested on $m$ patients, and each patient is to be given three drugs, a Steiner triple system provides a design in which each pair of drugs is tested on one patient.)

The incidence matrix $A = (a_{ij})$ of a Steiner triple system is a $(0,1)$-matrix whose rows correspond to the triples and whose columns correspond to the elements of $S$. Thus $a_{ij} = 1$ if and only if $j \in S_i$. Corresponding to each of the parameters $n = 3, 7, 9$ there is a unique Steiner triple system; for $n = 13$, there are two distinct systems, and for $n = 15$, there are eighty distinct systems [14]. The unique system for $n = 9$ is given by the 12 by 9 incidence matrix

\[
A_9 = \begin{bmatrix}
Z & I & 0 \\
0 & Z & I \\
I & 0 & Z \\
I & I & I
\end{bmatrix}
\]
where 0 is the zero matrix of order 3, I is the identity matrix of order 3, and

\[
Z = \begin{bmatrix}
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0
\end{bmatrix}.
\]

For the system (2.2), we have \( w(A_g) = 5 \), since the rows of \( A_g \) are covered by its first five columns, but by no smaller set of columns. Both systems corresponding to \( n = 13 \) have 1-width 7; for \( n = 15 \), the 1-width varies from 7 to 9 [5]. Indeed, in their study of 1-widths of Steiner triple systems, Fulkerson and Ryser [5] derived the lower bound

\[
w(A) \geq \frac{n - 1}{2}
\]

and determined conditions under which equality holds in (2.3). They remarked, however, that good upper bounds seem difficult to obtain. Since all triple systems on 15 or fewer elements have 1-widths at most \( \frac{2}{3} n - 1 \), they at one time speculated that this might be an upper bound. Ryser subsequently constructed a 45 element system as a potential counter-example, and conjectured that it had 1-width 30. Ryser's construction of the 45-element system starts with a particular 15-element system that has 1-width 9. This 15-element system has the 35 by 15 incidence matrix

\[
A_{15} = \begin{bmatrix}
Z & E & 0 \\
0 & Z & E \\
E & 0 & Z \\
I & I & I
\end{bmatrix}
\]
where

\[
Z = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
E = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

and \( I \) is the 5 by 5 identity. He then constructs a 45-element system having a 330 by 45 incidence matrix of the following form:

\[
A_{45} = \begin{bmatrix}
A_{15} & 0 & 0 \\
0 & A_{15} & 0 \\
0 & 0 & A_{15} \\
C^1 & I & p^1 \\
\ldots & \ldots & \ldots \\
C^{15} & I & p^{15}
\end{bmatrix}
\]

In (2.5) \( I \) is the 15 by 15 identity matrix; for \( k = 1, \ldots, 15 \) \( C^k \) is a 15 by 15 matrix whose \( k^{th} \) column is all ones and whose other columns are all zeros, and \( p^k \) is a 15 by 15 permutation matrix with

\[
\sum_{k=1}^{15} p^k = J,
\]

where \( J \) is the matrix of all ones.
We attempted to determine $w(A_{45})$ by solving the set covering problem (2.1), but did not succeed. We then used the matrix $A_9$ from (2.2) to construct a 27-element system structurally similar to $A_{45}$. The resulting 117 by 27 incidence matrix is

\[
A_{27} = \begin{bmatrix}
A_9 & 0 & 0 \\
0 & A_9 & 0 \\
0 & 0 & A_9 \\
C^1 & I & \beta^1 \\
\vdots & \vdots & \vdots \\
C^9 & I & \beta^9
\end{bmatrix}
\]

We solved (2.1) for $A_{27}$ and found that $w(A_{27}) = 18$. Thus $\frac{2}{3}n - 1$ is not an upper bound on 1-widths of Steiner triple systems on $n$ elements. The system (2.5) constructed by Ryser has $w(A_{45}) \leq 30$, and we feel reasonably sure that Ryser's speculation that $w(A_{45}) = 30$ is correct.

More detailed descriptions of $A_{27}$ and $A_{45}$ are given in the Appendix.

3. Computational Experience. An optimal linear programming solution in both problems is, of course, to set all variables equal to $\frac{1}{3}$, giving values of 9 and 15, respectively, for the sum of variables in the two problems. Optimal linear programming bases are not unique for $A_{27}$ and $A_{45}$; the computer produced optimal bases with determinants of magnitude $(27)^3$ and $(45)^3$, respectively. An optimal solution to (2.1) with $A = A_{27}$ is given by columns 1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15, 19, 20, 21, 25, 26, 27.
Both problems were attempted by an implicit enumeration algorithm [17] and a cutting plane algorithm\textsuperscript{1} based on Gomory's method of integer forms [7]. All computing was done on a Univac 1108. We also sent $A_{45}$ to several people who have obtained computational experience with integer programming codes different from those available to us. Of particular interest would be results obtained from codes based on the group-theoretic approach to integer programming [2, 8, 9, 10]\textsuperscript{2} and codes based on specialized algorithms for set covering problems [1, 11, 13, 16]. Nobody has informed us that they have solved (2.1) for $A_{45}$. There were a few negative responses. These generally did not report unequivocal failure, but difficulty in treating a problem of this size with an experimental code, etc.

Using the cutting plane code, we solved (2.1) for $A = A_9$ in about 20 seconds after adding 44 cuts. The cutting plane code was unsuccessful on the larger triple systems $A_{15}$, $A_{27}$, and $A_{45}$. Typically, the first few iterations yielded cuts that caused an increase in the objective value, but this was followed by long sequences of cuts that failed to produce a significant change in the value of the objective function. For example, in treating $A_{15}$, the initial sequence of 19 cuts raised the objective value from 5.0 to 5.425, after which 32 cuts left the value unaltered. (Recall that $w(A_{15}) = 9$.) This particular run took 82 seconds. In contrast, this problem was solved in 5 seconds using implicit enumeration.

This enumeration algorithm is very similar to one developed and tested by Geoffrion [6]. It uses simple tests for infeasibilities, and uses linear programming relaxations to obtain lower bounds and thus a necessary condition

\textsuperscript{1}The particular cutting plane code used was DULPDX, which is available on the MACC Program Library at the University of Wisconsin, Madison.

\textsuperscript{2}Shapiro [15] attempted to solve (2.1) with $A_{27}$ using IPA [9]. The enumeration was abandoned after about 1 minute of computer time on the IBM 360/67. Optimal solutions found for the group problems were highly infeasible and little progress was being made towards achieving feasibility in the enumeration phase.
on whether a given partial solution to (2.1) has a feasible completion of smaller value than the best available solution. The linear programming solutions also generate surrogate constraints. A dual algorithm is used to solve the linear programs. When fathoming occurs, the next partial solution considered is determined by backtracking. Branching is done by fixing a free variable at zero or one.

In our successful run with \( A_{27} \), we began with the partial solution corresponding to the optimal solution, which was found by inspection. About 6000 partial solutions were considered before optimality was verified. The run consumed about 16 minutes of computer time. Most of the fathoming came from the L.P. bounds, but this bounding test was quite ineffective for partial solutions having a small number of fixed variables. Several attempts at solving the problem for \( A_{45} \) failed, although we began with what is probably an optimal solution.

In contrast with our experience on these problems, others have reported considerable success in solving set covering problems with a variety of algorithms [1, 6, 9, 10, 11, 13, 16]. For example, Geoffrion [6] has reported solving, with an algorithm essentially identical to our implicit enumeration algorithm, several set covering problems with \( m = 30 \) and \( 30 \leq n \leq 90 \) in times varying from one to twelve seconds on the IBM 7044, a much slower computer. Lemke et al. [13] have solved considerably larger problems in a reasonable amount of time, e.g. a problem with \( m = 50, n = 450 \) was solved in about 2 minutes on an IBM 360/50. Gorry et al. [9] have solved some airline crew scheduling problems; one with \( m = 313 \) and \( n = 482 \) was solved in about 3 minutes on the IBM 360/85.

Compared to most of the covering problems considered in the literature, the problems \( A_{27} \) and \( A_{45} \) have a relatively small number of variables.
but a large number of constraints. Supposedly, however, of the two parameters, the number of variables is the more significant in solving a covering problem by implicit enumeration. Furthermore, the densities (number of ones/mn) of $A_{27}$ and $A_{45}$ are close to the densities in the problems considered in [6] and [13] mentioned above.

Why, then, are these two problems $A_{27}$ and $A_{45}$ difficult? We don't really know, but some plausible explanations might be:

1. The symmetries in the problems no doubt tend to increase the amount of enumeration required.

2. The rather large determinants of optimal L.P. bases contribute to the unattractiveness of cutting plane methods and indicate that group-theoretic methods may encounter difficulties.

3. The optimal value in the integer problem is large compared to the optimal value in the real (or rational) problem, thus emasculating the power of linear programming.

Recently Jeroslow [12] has constructed a simple family of $n$ variable, (0-1)-integer programs that cannot be solved by implicit enumeration, even using linear programming for fathoming, without enumerating at least $2^{n/2}$ possibilities. Although $A_{27}$ and $A_{45}$ are not in this family, they show that problems that arise naturally can be nasty. We hope that a reasonable set of such hard problems can be accumulated for the purpose of evaluating the efficiency of proposed integer programming algorithms.
Table 1: The triples of $A_{27}$. 

<table>
<thead>
<tr>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
<td>4 5 6</td>
<td>7 8 9</td>
<td>10 11</td>
<td>12 13 14</td>
<td>15 16 17</td>
<td>18 19 20</td>
<td>21 22 23</td>
<td>24 25 26</td>
<td>27 28 29</td>
</tr>
<tr>
<td>1 2 3</td>
<td>4 5 6</td>
<td>7 8 9</td>
<td>10 11</td>
<td>12 13 14</td>
<td>15 16 17</td>
<td>18 19 20</td>
<td>21 22 23</td>
<td>24 25 26</td>
<td>27 28 29</td>
</tr>
<tr>
<td>1 2 3</td>
<td>4 5 6</td>
<td>7 8 9</td>
<td>10 11</td>
<td>12 13 14</td>
<td>15 16 17</td>
<td>18 19 20</td>
<td>21 22 23</td>
<td>24 25 26</td>
<td>27 28 29</td>
</tr>
<tr>
<td>1 2 3</td>
<td>4 5 6</td>
<td>7 8 9</td>
<td>10 11</td>
<td>12 13 14</td>
<td>15 16 17</td>
<td>18 19 20</td>
<td>21 22 23</td>
<td>24 25 26</td>
<td>27 28 29</td>
</tr>
</tbody>
</table>

#1411
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 2: The triples of $A_{45}$. 

1. The table contains triples of numbers.
2. The triples are listed in a grid format.
3. Each cell contains a pair of numbers, separated by a comma.
4. The numbers range from 1 to 45.
5. The table is used to represent the triples of $A_{45}$. 

#1411
REFERENCES


