The Domino Method of General Integer Nonlinear Programming Applied to Problem 2 of Lawler and Bell

Posted on April 23, 2012 by jywong99999

Jsun Yui Wong

The computer program below tries to solve Problem 2 in Lawler and Bell [8]. Line 181 through line 1111 of the following computer program partly describe the problem.

The following computer program was originally modeled after the nuclear-chain-reaction picture on page 336 of the World Book Dictionary [1] and after the domino method of solving nonlinear systems of equations [20]. Line 501 through line 711 are illustrative. The intent behind line 501 through line 513 is to produce domino effect; X(8) through X(20) are slack variables. Looking at line 501, one sees that it is not hard to knock down X(8), domino one.

```
0 REM DEFDBL A-Z
2 DEFINT I,J,K,X
3 DIM B(519),N(519),A(2002),H(519),L(519),U(519),X(2002),D(511),P(511),PS(33),AA(1111)
12 FOR JJJJ=-32000 TO 32000
14 RANDOMIZE JJJJ
16 M=-1D+37
20 FOR J44=1 TO 7
21 A(J44)=FIX(RND*8)
22 NEXT J44
128 FOR I=1 TO 100
129 FOR KKQQ=1 TO 7
130 X(KKQQ)=A(KKQQ)
131 NEXT KKQQ
133 FOR IPP=1 TO FIX(RND*7)
181 J=1+FIX(RND*7)
182 REM GOTO 190
183 REM R=(1-RND*2)^A(J)
184 REM X(J)=A(J)+(FIX(RND*2))*R-(FIX(RND*2))*.05
185 REM X(J)=A(J)+(RND ^ (RND*10))
186 REM GOTO 222
188 REM GOTO 222
189 REM X(J)=A(J)+PA
190 X(J)=A(J)+(FIX(RND*2)-FIX(RND*2))
191 REM X(J)=A(J)+(FIX(RND*3)-FIX(RND*3)
396 NEXT IPP
443 FOR J44=1 TO 7
446 IF X(J44)<0 THEN 1670
```

Search
Recent Posts
- The Domino Method Solving a Nonlinear System of 41 Simultaneous Equations and 34 General Integer Variables, Second Edition
- Testing the Domino Method with a System of Four nonlinear Equations from the Literature
- The Domino Method Applied to Solving a Nonlinear System of Equations, Fourth Edition

Recent Comments
- Mr WordPress on Hello world!

Archives
- January 2014
- March 2013
- February 2013
- January 2013
- December 2012
- November 2012
- October 2012
- September 2012
- August 2012
- July 2012
- June 2012
- May 2012
- April 2012
- March 2012
- February 2012
- January 2012
- December 2011
- November 2011
- October 2011
- September 2011
- August 2011
- July 2011
- June 2011
- May 2011
- April 2011
This BASIC computer program was run with Microsoft's GW BASIC 3.11 interpreter. The complete output through JJJJ=-31990 is shown below. What follows is a hand copy from the computer-monitor screen; immediately below there is no rounding by hand.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>12</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>-16</td>
<td>-32000</td>
<td>-16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>9</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>-16</td>
<td>-31993</td>
<td>-16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
On a personal computer with an Intel 2.66 chip and the IBM basica/D interpreter, version GW BASIC 3.11, the throughput time from JJJJ=-32000 through JJJJ=-31990 was two seconds.

References


2. General terminology for integer programming. The most general problem called the mixed integer programming problem can be specified as: \( \min x_0 = c^T x \) subject to \( A x = b, \quad x_j \geq 0 \).

Integer programming problems generally take much longer to solve than the corresponding linear program obtained by ignoring integrality. It is wise therefore to consider the possibility of solving as a straightforward LP and then rounding, e.g., in the trim loss problem. The idea is quite general and has been applied to many other discrete optimisation problems (e.g., travelling salesman, job shop scheduling).

Let us assume we are trying to solve the mixed integer problem. Let us call this problem \( P_0 \). Integer-programming models arise in practically every area of application of mathematical programming. To develop a preliminary appreciation for the importance of these models, we introduce, in this section, three areas where integer programming has played an important role in supporting managerial decisions. We do not provide the most intricate available formulations in each case, but rather give basic models and suggest possible extensions.

9.1 Some Integer-Programming Models