

Computational Experience



[Home](#) [About](#)

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

Posted on [April 23, 2012](#) by [jywong999999](#)

Jsun Yui Wong

The computer program below tries to solve Problem 2 in Lawler and Bell [8]. Line 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 $X(11)$ are slack variables. Looking at line 501, one sees that it is not hard to knock down $X(8)$ 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),A

12 FOR JJJ=-32000 TO 32000

14 RANDOMIZE JJJ

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))*0.05-(FIX(RND*2))*0.05

187 REM X(J)=A(J)+(RND^(RND*10))*R

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

449 NEXT J44

501 X(8)= +7 -X(1)

502 X(9)= +7 -X(2)

503 X(10)= +7 -X(3)

504 X(11)= +15 -X(4)

505 X(12)= +15 -X(5)

506 X(13)= +7 - X(6)

507 X(14)= -6 + X(1) + X(2) +X(3)

508 X(15)= +7 - 3* X(1) -2* X(3) - X(5)

509 X(16)= -7 + X(1) *X(6)+ X(2) +3* X(5)

510 X(17)= -8 + X(4) + X(5) + 6*X(6)

```

511 X(18)= +20 -3* X(1)*X(3) -6* X(4) - 4* X(5)
512 X(19)= 15 -4* X(1) - 2* X(3) - X(6) * X(7)
513 X(20)= -25 +4* X(2)* X(7) + 3* X(4)*X(5)
554 FOR J44=8 TO 20
555 IF X(J44)<0 THEN PS(J44)=ABS(X(J44)) ELSE PS(J44)=0
556 NEXT J44
703 POBA2= - X(1)*X(7) -3*X(2)*X(6) - X(3)*X(5)-7*X(4)
711 POBA= POBA2 -999999999#*(
PS(8)+PS(9)+PS(10)+PS(11)+PS(12)+PS(13)+PS(14)+PS(15)+PS(16)+ PS(
17)+PS(18 )+PS(19 )+PS(20 ) )
759 POB1=POBA
863 P1NEWMAY=POB1
866 P=P1NEWMAY
1111 IF P<=M THEN 1670
1452 M=P
1453 PPOBA2=POBA2
1454 FOR KLX=1 TO 20
1455 A(KLX)= ( X(KLX) )
1456 NEXT KLX
1557 GOTO 128
1670 NEXT I
1899 IF M<-17 THEN 2222
2001 PRINT A(1),A(2),A(3),A(4),A(5)
2002 PRINT A(6),A(7),A(8),A(9),A(10)
2003 PRINT A(11),A(12),A(13),A(14),A(15)
2004 PRINT A(16),A(17),A(18),A(19),A(20)
2011 PRINT M,JJJJ,PPOBA2
2222 NEXT JJJJ

```

This BASIC computer program was run with Microsoft's GW BASIC 3.11 interpreter. complete output through JJJJ=-31990 is shown below. What follows is a hand copy of the computer-monitor screen; immediately below there is no rounding by hand.

```

0 4 2 0 2
1 3 7 3 5
15 13 6 0 1

```

3 0 12 8 23
-16 -32000 -16

1 4 1 0 2
1 2 6 3 6
15 13 6 0 0
4 0 9 7 7
-16 -31993 -16

0 4 2 0 2
1 7 7 3 5
15 13 6 0 1
3 0 12 4 87
-16 -31192 -16

0 5 1 0 2
1 3 7 2 6
15 13 6 0 3
4 0 12 10 35
-17 -31991 -17

0 4 2 0 2
1 4 7 3 5
15 13 6 0 1
3 0 12 7 39
-16 -31990 -16

On a personal computer with an Intel 2.66 chip and the IBM basic/D interpreter, v
GW BASIC 3.11, the throughput time from JJJJ=-32000 through JJJJ=-31990 was
seconds.

References

[1] Clarence L. Barnhart, Robert K. Barnhart (Editors). *The World Book Dictionary*,
World Book, Inc., a Scott Fetzer company, Chicago London Sydney Toronto.

- [2] T. Y. Chen, H. C. Chen (2009): Mixed-Discrete Structural Optimization Using a Niche Evolution Strategy, *Engineering Optimization*, 41:1, 39-58.
- [3] William Conley, *Computer Optimization Techniques*. New York/Princeton: Petro Books, Inc., Copyright 1980.
- [4] George B. Dantzig, Discrete-Variable Extremum Problems. *Operations Research* No. 2 (Apr., 1957), pp. 266-277.
- [5] M. Jaberipour, E. Khorram (2011), A New Harmony Search Algorithm for Solving Mixed-Discrete Engineering Optimization Problems, *Engineering Optimization*, 43:5 523.
- [6] Michael Junger, Thomas M. Lieblich, Dennis Naddef, George L. Nemhauser, Wil Pulleybank, Gerhart Reinelt, Giovanni Rinaldi, Lawrence A. Wolsey—Editors, *50 Years of Integer Programming 1958-2008*. Berlin: Springer, 2010.
- [7] A. H. Land, A. G. Doig, An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, Vol. 28, No. 3 (Jul., 1960), pp. 497-520.
- [8] E. L. Lawler, M. D. Bell, A Method for Solving Discrete Optimization Problems. *Operations Research*, Vol. 14, No. 6 (Nov.-Dec., 1966), pp. 1098-1112.
- [9] Duan Li, Xiaoling Sun, *Nonlinear Integer Programming*. Boston: Springer, 2006.
- [10] Han-Lin Li, Jung-Fa Tsai, A Distributed Computation Algorithm for Solving Problems with Integer Variables. *European Journal of Operational Research* 186 (2004) 891.
- [11] Harry M. Markowitz, Alan S. Manne, On the Solution of Discrete Programming Problems. *Econometrica*, Vol. 25, No. 1 (Jan., 1957), pp. 84-110.
- [12] Microsoft Corp., *BASIC*, Second Edition (May 1982), Version 1.10. Boca Raton: IBM Corp., Personal Computer, P. O. Box 1328-C, Boca Raton, Florida 33432, 1981.
- [13] Christina Oberlin, Stephen J. Wright, Active Set Identification in Nonlinear

Programming. *SIAM J. Optim.*, Vol. 17, No. 2, pp. 577-605.

[14] Singiresu S. Rao (2009), *Engineering Optimization: Theory and Practice*, Fourth Edition. New York: Wiley.

[15] E. Sandgren (1990): Nonlinear Integer and Discrete Programming in Mechanical Optimization. *Journal of Mechanical Design*, June 1990, Vol. 112, pp. 223-229.

[16] Dong Ku Shin, Z. Gurdal, O. H. Griffin Jr. (1990): A Penalty Approach for Nonlinear Optimization with Discrete Design Variables, *Engineering Optimization*, 16:1,29-42.

[17] P. B. Thanedar, G. N. Vanderplaats (1995), Survey of Discrete Variable Optimization for Structural Design. *Journal of Structural Engineering*, ASCE, Volume 121 (2), pp. 211-220.

[18] Jung-Fa Tsai, Han-Lin Li, Nian-Ze Hu (2002): Global Optimization for Signomial Discrete Programming Problems in Engineering Design, *Engineering Optimization*, 34:6, 622-637.

[19] S. Wang, K. L. Teo, H. W. J. Lee (1998): A New Approach to Nonlinear Mixed-Integer Programming Problems, *Engineering Optimization*, 30:3-4. 249-262.

[20] Jsun Yui Wong (2011, April 17). The Domino Method of Solving Nonlinear Systems of Equations. Retrieved from <https://computationalresultsfromcomputerprograms.wordpress.com/2011/04/17/the-domino-method-of-solving-nonlinear-systems-of-equations/>

[21] Ji-Hui Zhang, Xin-He Xu, An Efficient Evolutionary Programming Algorithm. *Computational & Operations Research* 26 (1999) 645-663.

Loading...

This entry was posted in [Uncategorized](#). Bookmark the [permalink](#).

The Domino Method of General Integer Nonlinear Programming Applied to a Five-Step Cantilever Beam Design

The Domino Method of General Integer Nonlinear Programming Applied to Problem 10 of Law

Computational Experience

Seizing intellectual power: the dialogue at the New York African Burial Ground, the perception of co-creation is sublimated structuralism.

The Domino Method of General Integer Nonlinear Programming Applied to Problem 2 of Lawler and Bell, the explosion has traditionally stops sorted rift.

International tables for crystallography, the channel, by definition, illegally keeps the index of sodium adsorption, thus the dream of the idiot came true-the statement is fully proved.

The instructive animation: Helping students build connections between words and pictures in multimedia learning, coordinate system reimburses realism.

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept