OPTIMISING FOR MAXIMUM PROFIT FOR A PRODUCTION PROBLEM USING LINEAR PROGRAMMING

Report Submitted for Review-1

OERATIONS RESEARCH (PBL) - A1 SLOT

WINTER SEMESTER 2015-16

Faculty Name: Dr. Dega Nagaraju

Submitted by

MANORATH BAJAJ (14BCE0442)

SIMARJIT SINGH (14BPI0048)

ASHUTOSH MISRA(14BPI0066)

VIT UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

VELLORE ▪ CHENNAI

www.vit.ac.in

SCHOOL OF MECHANICAL ENGINEERING

CONTENTS

ABSTRACT

The situation that we chose to work on for our project is as follows

A manufacturer currently owns 10 machines namely Lathe Machine, Milling Machine, Cold Stamp Machine, Furnace, Dropdown Hammer, Boring Machine, CNC Machine, Turning wheel machine, Broaching Machine, Electric Saw. These machines are capable of producing 10 products which for the sake of simplicity we'll name as A,B,C,D,E,F,G,H,I,J. These products require a specific amount of time on each machine which is known to us. In order to maximize the profit, we need to determine which products should be manufactured by the manufacturer and in what amount.

This situation can be represented numerically in the form of a Linear Programming Model. Linear Programming is used worldwide for optimizing profit problem. It is a method to achieve the best outcome (Max profit at low cost) whose requirement is represented by linear relationships. There are various ways to approach this problem. We can either find a solution to this situation through manual calculation or using a Programming Language such as C,C++,Matlab etc.

We chose MATLAB over C, C++ or Python because MATLAB is a high level specifically designed to cater to the Numerical computing this project needs. MATLAB is much easier to code on allowing access to symbolic computing and graphical representation which this project needs over other High level languages. The main aim of the program will based on Maximizing of profit and we will be using the given problem to demonstrate the MATLAB code.

Rather than developing a code specific to this problem, we also chose to approach our problem in a generalized way and wrote a script which not only would find the feasible solution to the given situation but would find a feasible solution to any Linear Programming Model and thus ensuring reusability of our script.

Introduction

Our project is about solving Linear Programming problems using matlab. Our main point of concern was code reusability and the ability to solve any linear programming program using matlab. Although we have a default problem, our project was more focused on making a generalised code that saves time in the future. One script to solve all linear programming questions. The main problem we faced was to find and try to solve all the possible scenarios that can occur in the problem. We believe that we have addressed all the possible scenarios in the script.

The problem is a optimisation problem, where a manufacturer has 10 machines and has to decide what and how much to manufacture between 10 products in order to gain maximum profit.

MACHINES	Time required per unit(min) for product										
	A	B	C	D	E	F	G	H	I	J	Available Time(min)
Late Machine	7	10	4	9	0	40	69	2	5	4	3000
Milling Machine	3	40	1	1	4	0	6	8	10	8	2000
Cold stamp Machine	7	11	23	16	41	4	0	0	6	2	3200
Furnace	84	52	63	14	0	4	90	65	43	11	3060
Dropdown Hammer	6	45	7	12	0	32	0	15	0	4	1000
Boring Machine	75	64	51	32	47	0	50	0	3	15	5200
CNC	7	51	12	62	23	24	13	32	42	10	4222
Turning wheel Machine	47	52	32	12	14	41	23	35	1	100	2000
Broching Machine	45	62	15	32	47	51	22	0	0	0	3000
Electric Saw	95	43	62	15	4	77	5	2	66	5	4200
Profit Per Unit	45	100	30	50	25	70	63	42	15	160	

We use MATLAB's inbuilt function linprog() to solve the problem.

linprog

Solve linear programming problems

Linear programming solver

Finds the minimum of a problem specified by

$$\min_{x} f^T x \text{ such that } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub. \end{cases}$$

f, *x*, *b*, *beq*, *lb*, and *ub* are vectors, and *A* and *Aeq* are matrices.

The values given in the table or any general LPP problem can be converted to the following values and can be solved accordingly.

In our generalised code, we have used the If..Else loop conditions with all the possibilities and mainly utilising the linprog function.

Literature Review

Löfberg, J. on 4 Sept. 2004 concluded his thesis as follows. The MATLAB toolbox YALMIP is introduced. It is described how YALMIP can be used to model and solve optimization problems typically occurring in systems and control theory. Thus, The MATLAB toolbox YALMIP is introduced which is one of the major toolbox of MATLAB.

Yin Zhang on 8 Aug 1997 concluded his thesis as follows. Implementation of a primal-dual infeasible-interior-point algorithm for large-scale linear programming under the MATLAB environment is described. The resulting software is called LIPSOL — Linear-programming Interior-Point SOLvers. LIPSOL is designed to take the advantages of MATLAB's sparse-matrix functions and external interface facilities, and of existing Fortran sparse Cholesky codes. Under the MATLAB environment, LIPSOL inherits a high degree of simplicity and versatility in comparison to its counterparts in Fortran or C language.

Kenneth Holmström on 1999 concluded his thesis as follows. TOMLAB is a general purpose, open and integrated Matlab development environment for research and teaching in optimization on Unix and PC systems.

Venkataraman, P on October 22, 1999 concluded his thesis as follows.The paper presents a compact Matlab implementation of a topology optimization code for compliance minimization of statically loaded structures. The total number of Matlab input lines is 99 including optimizer and Finite Element subroutine. The paper presents a compact Matlab implementation of a topology optimization code for compliance minimization of statically loaded structures.

Huang, Y. and McColl, W.F. on 1996, October concluded their thesis as follows. In this paper, an improved method based on Nelder and Mead's simplex method (1965) is described for unconstrained function minimization. The information of the function values evaluated in the previous steps is combined into the direction determination of the succeeding simplex. In this way, the new method reflects a more reasonable descendant search nature of the simplex method. The information of the function values evaluated in the previous steps is combined into the direction determination of the succeeding simplex.

Lei, Z in 2004 concluded his thesis as follows. PID parameter tuning and optimizing is an important issue in the field of automatic control. Based on the open design method, the discrete controller is designed. In order to optimize the design of discrete controller, Visual C++ language is adopted to edit the control program, and simplex optimization methed is utilized to tune the PID parameters, making it possess the general meaning. Simulation results show that the controller designed with this method has good validity and generality. PID parameter tuning and optimizing is an important issue in the field of automatic control.

MATHEMATICAL MODEL DEVELOPMENT

3.1 Assumptions

We assume that the only materials required by the product are the ones mentioned. We also assume that the time taken by the machines to produce is the time given in the table. Everything happens under ideal condition. No extra costs other than the given are assumed.

As the linprog() function gives the answer in decimal, which certainly cannot be true as quantities are integer values, we assume the that the optimal quantities of products are integer and hence the floor value of the decimal value is taken and multiplied with the objective function(negative of the objective function in this case to get the correct values)

3.2 Notations and Model Development

The Notations that we have used in the code are as follows.

Let x1 be the quantity of product A to be produced
Let x2 be the quantity of product B to be produced
Let x3 be the quantity of product C to be produced
Let x4 be the quantity of product D to be produced
Let x5 be the quantity of product E to be produced
Let x6 be the quantity of product F to be produced
Let x7 be the quantity of product G to be produced
Let x8 be the quantity of product H to be produced
Let x9 be the quantity of product I to be produced
Let x10 be the quantity of product j to be produced

Xi>=0
Where I ranges from 1 to 10.

Flags = flag, aflag, bflag, cflag y
Flags are used to take in user inputs and to direct the program in a particular direction depending on the user, currently all flags are displayed in the main program which wont be the case in the finished program. It is currently displayed to check for bugs.

f= The function vector that has to be maximised or minimised. The general code multiplies -1 to this function if the function has to be maximised.
The function for the default matrix is

```
f=[-45 -100 -30 -50 -25 -70 -63 -42 -15 -160]
```

A =is a matrix of all the LHS values to which the function f is subjected too(inequalities)

The A Matrix for the default question is

```
A=[ 7 10 4 9 0 40 69 2 5 4
  3 40 1 1 4 0 6 8 10 8
  7 11 23 16 41 4 0 0 6 2
  84 52 63 14 0 4 90 65 43 11
  6 45 7 12 0 32 0 15 0 4
  75 64 51 32 47 0 50 0 3 15
  7 51 12 62 23 24 13 32 42 10
  47 52 32 12 14 41 23 35 1 100
  45 62 15 32 47 51 22 0 0 0
  95 43 62 15 4 77 5 2 66 5]
```

b= The vector of the RHS value of the equations to which the function f is subjected too

the b Value for the default question is.

```
b=[3000 2000 3200 3060 1000 5200 4222 2000 3000 4200]
```

lb= the lower bound which is zero in all LPP practical problems for quantity optimisation.

Aeq = matrix of the LHS values of the equations the function is subjected to(equality eqeuations)

The Default question does not have equality equations.

beq= the Vector of RHS value of equality equations the function is subjected too

The Default question does not have any equality equations.

X is a matrix containing the values of x1,x2,x3…x10.
fval is has the multiplication of objective function to X matrix.

Lambda,exitflag and output are used for error handling.

Results and Discussions

As the values of X matrix were in decimal, using the floor function gives us the following results

The products A, B,C, E,F, H, I will not be produced for maximum optimisation.

The Quantity of product D is 62.
The Quantity of product G is 23.
The Quantity of Product J is 7.

The Total Profit (Maximum value of Objective function) is ₹ 5669.

The Optimisation of the given problem was successful and The Factory Owner is advised to Produce Only Product D, G and J in the given quantity for maximum profit. The Total Maximum profit value was also found. The objective function can be and has been maximised as the Error handling Flags show no Error

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
  ---------------------------------------------------
  Iter    0:  8.24e+004 6.29e+002 2.10e+005 6.27e+004
  Iter    1:  3.12e+004 2.78e+001 3.17e+004 2.99e+000
  Iter    2:  2.98e+004 2.50e+001 3.17e+004 2.85e+000
  Iter    3:  2.98e+004 2.50e+001 3.16e+004 2.85e+000
  Iter    4:  2.96e+004 2.46e+001 2.94e+004 2.84e+000
  Iter    5:  2.44e+004 2.25e+001 2.80e+004 2.34e+000
  Iter    6:  2.23e+004 2.07e+001 2.59e+004 2.14e+000
  Iter    7:  1.08e+004 6.96e+000 1.31e+004 1.04e+000
  Iter    8:  7.39e-011 1.80e+000 4.05e+003 2.78e-001
  Iter    9:  1.21e-010 2.79e-001 1.03e+003 1.00e-001
  Iter   10:  7.28e-011 5.79e-002 2.21e+002 2.25e-002
  Iter   11:  6.72e-012 5.12e-014 1.30e+001 2.27e-003
  Iter   12:  8.52e-012 4.49e-014 9.70e-003 1.69e-006
  Iter   13:  1.07e-011 4.63e-014 9.70e-008 1.70e-011
Optimization terminated.
X=
     0
     0
     0
    62
     0
     0
    23
     0
     0
     7

Total Profit=        5669

>>
```

Conclusion

As it has been shown, the given problem has been successfully optimised using Linear Programming using Matlab. Matlab is a very powerful mathematical tool and can be used to solve Linear programming problems with any number of variables unless sufficient equations are not given. Optimising using the Built-in linprog() makes High variable Problems Easy and quick which might take a lot of time if done in Simplex or Graphical Method.

The Generalised Code for any linear programming question has also been scripted alongside the main script. It is an easier way to enter any linear programming problem rather than making a new script for every problem every time.

We think we were able to optimise for Maximum profit for the given production problem and were able to answer it successfully.

APPENDIX

```
clc
clear all
%Default values
y=input('Press 1 to use deafult values, press 2 to solve your own problem')
if y==1
    f=[-45 -100 -30 -50 -25 -70 -63 -42 -15 -160];
    A=[ 7 10 4 9 0 40 69 2 5 4
        3 40 1 1 4 0 6 8 10 8
        7 11 23 16 41 4 0 0 6 2
        84 52 63 14 0 4 90 65 43 11
        6 45 7 12 0 32 0 15 0 4
        75 64 51 32 47 0 50 0 3 15
        7 51 12 62 23 24 13 32 42 10
        47 52 32 12 14 41 23 35 1 100
        45 62 15 32 47 51 22 0 0 0
        95 43 62 15 4 77 5 2 66 5];
    b=[3000 2000 3200 3060 1000 5200 4222 2000 3000 4200];
    lb=zeros(10,1);
    options=optimset('display','iter');
    [x,fval,exitflag,output,lambda]=linprog(f,A,b,[],[],lb,[],[],options);
    fprintf('X=');
    disp(floor(x));
    zval=0;
    for i=1:length(f)
        zval=zval+(-1*((f(i)*floor(x(i)))));
    end
    fprintf('Total Profit=');
    disp(zval)


else
    %user values
    z=input('Enter the number of variables in the equation')
    f=input('enter the objective function as a vector')
    flag=input('Press 1 for maximisation anything else for minimisation')
    if flag==1
        f=f*(-1);
    end
    A=input('Enter the LHS of inequality equations as a matrix')
    b=input('Enter the RHS of inequality equations as a vector')
    lb=zeros(z,1);
    cflag=input('Press 1 if there are equality equations');
    if cflag==1
       Aeq=input('Enter the LHS of equality equations as a matrix')
       beq=input('Enter the RHS of equality equations as a vector')
       aflag=input('press 1 if you have the initial solution')
       if aflag==1
           x0=input('Enter the initial solution as a vector')
           options=optimset('display','iter');

[x,fval,exitflag,output,lambda]=linprog(f,A,b,Aeq,beq,lb,[],x0,options)
       else
            options=optimset('display','iter');

[x,fval,exitflag,output,lambda]=linprog(f,A,b,Aeq,beq,lb,[],[],options)
       end
```

```
    else
        bflag=input('press 1 if you have the initial solution')
        if bflag==1
            x0=input('Enter the initial solution as a vector')
            options=optimset('display','iter');

[x,fval,exitflag,output,lambda]=linprog(f,A,b,[],[],lb,[],x0,options)
        else
            options=optimset('display','iter');

[x,fval,exitflag,output,lambda]=linprog(f,A,b,[],[],lb,[],[],options)
        end
    end
end
```

References

- Vanderbei, R.J., 2014. *Linear programming*. Springer.
- Holmström, K., 1999. The TOMLAB optimization environment in Matlab.
- Zhang, Y., 1998. Solving large-scale linear programs by interior-point methods under the Matlab∗
- Environment†. *Optimization Methods and Software*, *10*(1), pp.1-31.
- Venkataraman, P., 2009. *Applied optimization with MATLAB programming*. John Wiley & Sons.
- Löfberg, J., 2004, September. YALMIP: A toolbox for modeling and optimization in MATLAB.
- In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on* (pp. 284-289). IEEE.
- Rao, S.S. and Rao, S.S., 2009. *Engineering optimization: theory and practice*. John Wiley & Sons.
- Sigmund, O., 2001. A 99 line topology optimization code written in Matlab.*Structural and multidisciplinary optimization*, *21*(2), pp.120-127.
- Venkataraman, P., 2009. *Applied optimization with MATLAB programming*. John Wiley & Sons.
- Houck, Christopher R., Jeff Joines, and Michael G. Kay. "A genetic algorithm for function optimization: a Matlab implementation." *NCSU-IE TR*95.09 (1995).
- Luenberger, D.G., 1973. *Introduction to linear and nonlinear programming*(Vol. 28). Reading, MA: Addison-Wesley.
- Huang, Y. and McColl, W.F., 1996, October. An improved simplex method for function minimization.
- In *Systems, Man, and Cybernetics, 1996., IEEE International Conference on* (Vol. 3, pp. 1702-1705). IEEE.
- Butt, R., 2011. *Applied linear algebra and optimization using Matlab*. Mercury Learning and Information.
- Lei, Z., 2004. Simplex method based optimal design of PID controller.*Information and Control*, *33*(3), pp.376-379.
- Simon, H.A. and Newell, A., 1958. Heuristic problem solving: The next advance in operations research. *Operations research*, *6*(1), pp.1-10.
- Vidale, M.L. and Wolfe, H.B., 1957. An operations-research study of sales response to advertising. *Operations research*, *5*(3), pp.370-381.