USING OPTIMIZED MATLAB IN MECHANICAL DESIGN

Nguyen Thanh Tung(1), Nguyen Thi Huynh Nga(1)

(1) Thu Dau Mot University

Corresponding author: nganth@tdmu.edu.vn

DOI: 10.37550/tdmu.EJS/2024.01.516

Article Info

Volume: 6
Issue: 01
March 2024
Received: Jan. 11th, 2024
Accepted: Feb. 19th, 2024
Page No: 45-53

Abstract

In this topic, we applied the Fmincon function to the optimum question when choosing the structure of a 7-bar bearing steel bearing, divided into groups of the same size, including group 1 (1, 2 bars), group 2 (3, 4, and 5 bars), and group 3 (6, and 7 bars) with three fixed head points and two bearing points. Using the Matlab software code, we have identified the structures of each group of steel bars corresponding to the radius of 1.564cm, 3.509cm, and 4.724cm, respectively. Through this, we can identify the 1, 2, and 3 bars that are resistant to traction; the 3, 5, 6, and 7 bars that are subject to compression; and the 4 bars alone that are not subject to the action of the force. The results show that, using the optimal method, we choose the different sizes, thicknesses, and volumes of the pipe so that it best suits the technical requirements of the paper, so as to avoid waste of raw materials, affecting the economic cost.

Keywords: constraint, fmincon function, mechanical design, optimization, truss problem

1. Introduction

Optimizing is the process of finding and determining the maximum or minimum value of a numeric function or a set of constraints based on certain criteria (Alan et al., 2013). Optimization is commonly applied in many fields, including economics, engineering, computer science, mathematics, physics, chemistry, and other sciences (Annegret Burtscher, 2020; Patrick Bangert, 2012). In economics, optimization is often used to solve problems related to maximizing profits or minimizing costs (Seán Moran, 2017). Optimization tools are commonly used to optimize decisions in the fields of production management (Vaidyanathan et al., 1998), capital management, finance (Shaomin Ren, 2022), marketing, and data analysis (Boris Goldengorin, Sergei Kuznetsov, 2023). Optimization methods often involve using algorithms to find the optimal solution or using modeling methods (Sunil Kumar, Yasir Rizvi, 2018) to describe a system and find optimal solutions for that system (Haleemah Jawad Kadhim et al., 2021). Common optimization methods include linear programming (*Mark, 1998*), integer planning, shortest path, networking (Thomas, 2013), and evolutionary methods such as genetic algorithms and multitasking optimization (Yanchi Li, Wenyin Gong, Shuijia Li, 2023).

Features of optimization include: optimized value search is the process of searching for the maximum or minimum value of a numeric function or a set of constraints; Optimization criteria: optimization processes based on certain criteria, e.g., maximizing profit or minimizing costs; Binding-optimization may have certain constraints; for example, certain parameters must not exceed certain limits. Optimization methods: The optimization methods can include using algorithms to find the optimal solution or using modeling methods to describe a system and find optimal solutions for that system.

2. Method of application

2.1. Fmincon function

Fmincon is a function in MATLAB that is used to solve binding optimization issues, which are typically described as follows:

$\min_x f(x)$

subject to $c(x) \leq 0$ $c_{eq}(x) = 0$

$A_{eq}.x = b_{eq}$

$lb \leq x \leq ub$

[x, fval] = fmincon(fun, x0, A, b, A_{eq}, b_{eq}, lb, ub, nonlcon, options)

In this, f(x) is the target function to be optimized; c(x) and $c_{eq}(x)$ are the non-equal and binding equal functions; and A, b, A_{eq}, and b_{eq} are the matrices and vectors involved in linear binding. lb and ub are the lower limit vectors and the upper limit of the optimum variable x. The fmincon function uses non-binding or binding optimization methods, such as interior-point methods or gradient-based methods. It can solve many kinds of optimization issues, including nonlinear and non-linear optimization problems. The fmincon function uses non-binding or binding optimization methods, such as interior-point methods or gradient-based methods. It can solve many kinds of optimization issues, including nonlinear and non-linear optimization problems. Using fmincon, we can find the optimal value of the variable x in the given constraints, which helps solve many practical problems such as design optimization, production planning, and other applications (Jigar et al., 2018).

2.2. The Fmincon function application solves the problem in an optimal way.

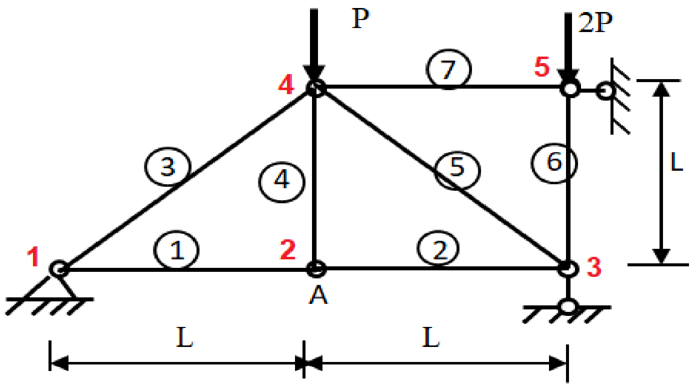

Figure 1: Structure of steel bars

Consider steel bars as shown in Figure 1, consisting of 7 bars from 1 to 7 numeric symbols with an outer circle and 5 nodes marked in red, bars with empty circular columns, with a radius in the r_i and thickness of t_i; the bars consisted of 3 fixed points at the nodes 1, 3, and 5 like Figure 1.

The length of the bars is L = 3m, the height is h = L, the coefficients include [σU] = - [σL] = 16 (kN/cm^2), with E = 2,104 (kN/cm^2), the force of action is P = 25 kN. The requirement of the assignment is the design of the cross-section with the best radius and thickness to minimize the weight of the rail while maintaining the best resistance.

Suppose we divide the bars into three main groups: Group 1 (bars 1, 2) has the same diameter with a radius of r_1; Group 2 (including bars 3,4,5) also has a similar diameter of r_2; and Group 3 (including two bars 6, 7) has a radius of r_3. The condition is that thickness is ≥ r_i, so we have X1, X2, X3, which is the surface area across groups 1, 2, and 3, respectively.

Here's a solution for solving math problems with the Matlab program:

We declare the bar number of the bar, the number of buttons on each bar, and the free layer of the specific buttons into Matlab's code. Disclosure of the theoretical parameters given by the subject, including length L, height h = L, force P, elasticity E, index σ, sigU parameters, sigL.

We then proceed to determine the gravity of the nodes gcoord (i,j) = m and the link between the bars and the specific nodes (k,l) = n.

Then we set the loop, declared the mesh grid, set the target functions, the binding conditions, the matrices, and the calculating functions bcdof and bcval.

Form functions related to compression, pull, matrix for force vector,

The function determines the voltage that acts on the bars and calculates the transposition of the nodes when the force P acts.

2.3. The code in Matlab

Based on the requirements of the truss problem, following the steps presented above, we proceed to write code for the problem in the Matlab programming language as follows:

```
clear all; close all; clc;
disp ('The program is working. Please wait for a while');
disp ('DIMENSIONS IN : KN-cm')
format short;
%------------------------------------------------------------------------
%            INITIAL DATA
%------------------------------------------------------------------------
L= 300;   % L = 3m = 300 cm
h= 300;    % h = L = 300 cm
P= 25;     % P = 25 kN : concentrated load
E= 2e4  ;  % Elastic modulus of the material (kN/cm2)
sigU =16 ;  % Upper bound of stresses (kN/cm2)
sigL =-16;  % Lower bound of stresses (kN/cm2)
%------------------------------------------------------------------------
%         CONTROL INPUT DATA
%------------------------------------------------------------------------
nel =7;          % number of elements
nnel=2;           % number of nodes per element
ndof=2;          % number of dofs per node
edof=nnel*ndof;   % number of dofs per element
nnode=5;         % total number of nodes in system
sdof=nnode*ndof; % total system dofs
%------------------------------------------------------------------------
%          NODAL COORDINATES
%------------------------------------------------------------------------
gcoord(1,1)=0.0 ;  gcoord(1,2)=0.0 ;
gcoord(2,1)=L  ;   gcoord(2,2)=0.0 ;
gcoord(3,1)=2*L ;  gcoord(3,2)=0.0 ;
gcoord(4,1)=L  ;   gcoord(4,2)=h  ;
gcoord(5,1)=2*L ;  gcoord(5,2)=h  ;
X=zeros(nnode,1);                 % x-coordinates of nodes
Y=zeros(nnode,1);                 % y-coordinates of nodes
for inode=1:nnode
  X(inode)=gcoord(inode,1);
  Y(inode)=gcoord(inode,2);
end
%------------------------------------------------------------------------
%          NODAL CONNECTIVITY
%------------------------------------------------------------------------
nodes(1,1)=1; nodes(1,2)=2;
nodes(2,1)=2; nodes(2,2)=3;
```

```
nodes(3,1)=1; nodes(3,2)=4;
nodes(4,1)=2; nodes(4,2)=4;
nodes(5,1)=4; nodes(5,2)=3;
nodes(6,1)=3; nodes(6,2)=5;
nodes(7,1)=4; nodes(7,2)=5;
% Length of bar
Lbar = zeros(nel,1);
for i=1:nel
  Lbar(i)=sqrt((X(nodes(i,2))-X(nodes(i,1)))^2+...
        +(Y(nodes(i,2))-Y(nodes(i,1)))^2);
end
%-----------------------------------------------------------------------
%        MATERIAL AND GEOMETRIC PROPERTIES
%-----------------------------------------------------------------------
syms X1 X2 X3 real          % cross-section area variables (cm2)
A=[X1 X1 X2 X2 X2 X3 X3];  % cross-section area matrix of elements
% ---------------------------------------------------------------------
%          MESH CONFIGURATION
% ---------------------------------------------------------------------
figure;
h=gcf;
set(h,'name','Truss form');
set(h,'NumberTitle','off');
axis equal;
title('Undeformation Truss Form');
m=zeros(nel,2);   % matrix of beginning nodes of the elements
n=zeros(nel,2);    % matrix of ending nodes of the elements
for iel=1:nel
  m(iel,:)=[X(nodes(iel,1)) Y(nodes(iel,1))];
  n(iel,:)=[X(nodes(iel,2)) Y(nodes(iel,2))];
end
for iel=1:nel
  x=[m(iel,1) n(iel,1)];
  y=[m(iel,2) n(iel,2)];
  if iel==1 | iel==2
    plot(x,y,'r','LineWidth',3);
     axis equal;
  elseif iel==3 | iel==4 | iel==5
    plot(x,y,'b','LineWidth',3);
    axis equal;
  else
    plot(x,y,'g','LineWidth',3);
    axis equal;
  end
% locate the order number of elements at the midpoint
  text((x(1)+x(2))/2,(y(1)+y(2))/2,num2str(iel));
  hold on;
end
% locate the order number of nodes
  for inod=1:nnode
  text(X(inod),Y(inod),num2str(inod));
  end
```

```
%----------------------------------------------------------------------
%          APPLIED CONSTRAINTS
%----------------------------------------------------------------------
bcdof(1)=1;   % 1st dof (horizontal displ) is constrained
bcval(1)=0;   % whose described value is 0
bcdof(2)=2;   % 2nd dof (vertical displ) is constrained
bcval(2)=0;   % whose described value is 0
bcdof(3)=6;   % 6th dof (horizontal displ) is constrained
bcval(3)=0;   % whose described value is 0
bcdof(4)=9;   % 9th dof (horizontal displ) is constrained
bcval(4)=0;   % whose described value is 0
%----------------------------------------------------------------------
%          INITIALIZATION TO ZERO
%----------------------------------------------------------------------
ff=sym(zeros(sdof,1));          % system force vector
kk=sym(zeros(sdof,sdof));        % system stiffness matrix
SS=sym(zeros(nel,sdof));
index=zeros(nnel*ndof,1);        % index vector
elforce=zeros(nnel*ndof,1);       % element force vector
eldisp=sym(zeros(nnel*ndof,1));    % element nodal displacement vector
k=sym(zeros(nnel*ndof,nnel*ndof)); % element stiffness matrix
stress=sym(zeros(nel,1));         % stress vector for every element
%---------------------------------------------------------------------
%          APPLIED NODAL FORCE
%---------------------------------------------------------------------
ff(8)=-P;           % the 4th node has the force P in the downward direction
ff(10)=-2*P;         % the 5th node has the force 2P in the downward direction
for iel=1:nel         % loop for the total number of elements
  nd(1)=nodes(iel,1);  % 1st connected node i for the (iel)-th element
  nd(2)=nodes(iel,2);  % 2nd connected node j for the (iel)-th element
  x1=X(nd(1));      % x-coordinate of 1st node i
  y1=Y(nd(1));      % y-coordinate of 1st node i
  x2=X(nd(2));      % x-coordinate of 1st node j
  y2=Y(nd(2));      % y-coordinate of 1st node j
  leng=(sqrt((x2-x1)^2+(y2-y1)^2));  % the length of the element
  c=(x2-x1)/leng;            % cosin between element and x-coordinate direction
  s=(y2-y1)/leng;            % sin between element and x-coordinate direction
  index=feeldof(nd,nnel,ndof);   % system dofs of the iel-th element
  [k]=fetruss(E,leng,A(iel),c,s);  % Compute stiffness matrix
  [kk]=feasmbl(kk,k,index);     % Assembly into the system matrix
  S=(E/leng)*[-c -s c s];
  edof=length(index);
  for i=1:edof
    ii=index(i);
    SS(iel,ii)=SS(iel,ii)+S(i);    % stresses matrix
  end
end
%----------------------------------------------------------------------
%       APPLY CONSTRAINTS AND SOLVE THE MATRIX
%----------------------------------------------------------------------
[kk,ff]=feaplyc(kk,ff,bcdof,bcval);  % apply the boundary conditions
displacement=simplify(kk\ff);     % solution for nodal displacements
```

```
stress=simplify(SS*displacement);  % stresses of bars
stress
displacement
save femtruss A stress Lbar;
disp('*****************************************************************');
disp('***          COMPLETED !          ***');
disp('*****************************************************************');
```

3. Result

Through the process of applying the Matlab code, we obtained the image below, which is a system of 7 bars with 5 nodes after solving the optimum problem. The bars are in groups 1, 2, and 3 for different colors by size classification.

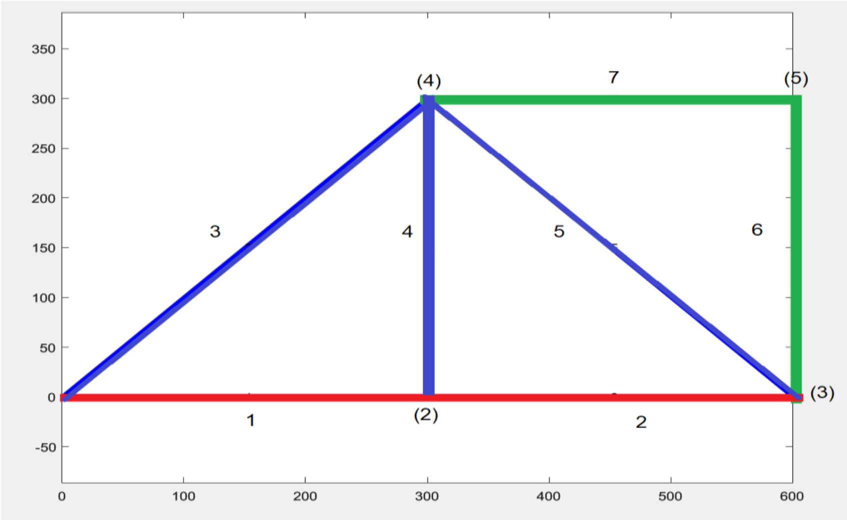

Figure 2: Optimal process results on Matlab

The two-step method is to optimize the shape of the rail and calculate the voltage and positioning of the points compared to the pre-optimal.

Based on the results table above, we see:

1st, 2nd, and 5.6th bars are compressed; 7th bars alone are non-compressed.

Button 1 stands still; button 2 is moved both x and y; button 3 is moved in x; button 4 is also moved in X and Y; button 5 is moving in x.

TABLE 1. The results of the calculation determine the resistance state of the bars

Bar	Normal stress	Comment
1	25/2*(87960930222080*X2+124395540479019*X3)/(43980465111040* X2*X3 +87960930222080*X1*X2+124395540479019*X1*X3)	Tension
2	25/2*(87960930222080*X2+124395540479019*X3)/(43980465111040* X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)	Tension
3	-621977702395095/35184372088832*(87960930222080*X2*X3+ +87960930222080*X1*X2+124395540479019*X1*X3)/X2/ (43980465111040*X2*X3+87960930222080*X1*X2 +124395540479019*X1*X3)	Compression
4	0	
5	-621977702395095/35184372088832*X1*(87960930222080*X2+ 124395540479019*X3)/X2/(43980465111040*X2*X3+ 87960930222080*X1*X2+124395540479019*X1*X3)	Compression
6	-50/X3	Compression
7	-1099511627776000/(43980465111040*X2*X3 +87960930222080*X1*X2+124395540479019*X1*X3)*X2	Compression

TABLE 2. Results of the positioning of nodes

Node	Displacements	
	x-direction	y-direction
1	0	0
2	3/16*(87960930222080*X2+124395540479019*X3)/(43980465111040*X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)	-3/703687441776640**(386856262276681335905976320 0*X2^2+10941947456012918357250539520*X2*X3+10941947456012918357250539520*X1*X2+15474250491067254579979202361*X1*X3)/X2/(43980465111040*X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)
3	3/8*(87960930222080*X2+124395540479019*X3)/(43980465111040*X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)	0
4	16492674416640/(43980465111040*X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)*X2	-3/703687441776640*(3868562622766813359059763200*X2^2+10941947456012918357250539520*X2*X3+10941947456012918357250539520*X1*X2+15474250491067254579979202361*X1*X3)/X2/(43980465111040*X2*X3+87960930222080*X1*X2+124395540479019*X1*X3)
5	0	-3/4/X3

TABLE 3. Results of the calculation of thickness and voltage after 20 loops

Loop	Results of variables			Stresses of bars						
	X1	X2	X3	1	2	3	4	5	6	7
0	10	20	30	0.8443	0.8443	-1.1708	0	-0.597	-1.66	-0.2705
1	0.5277	1.4635	3.125	13.6408	13.6408	-17.203	0	-6.9556	-16	-3.3934
2	0.4499	1.5735	3.125	14.5329	14.5329	-16.5932	0	-5.876	-16	-3.8158
3	0.4086	1.6318	3.125	15.0607	15.0607	-16.3326	0	-5.3333	-16	-4.0614
4	0.3846	1.6658	3.125	15.3875	15.3875	-16.2	0	-5.0247	-16	-4.2122
5	0.3699	1.6866	3.125	15.5959	15.5959	-16.1253	0	-4.8373	-16	-4.3079
6	0.3606	1.6998	3.125	15.7313	15.7313	-16.0806	0	-4.7191	-16	-4.3699
7	0.3545	1.7084	3.125	15.8205	15.8205	-16.0527	0	-4.6428	-16	-4.4106
8	0.3505	1.714	3.125	15.8796	15.8796	-16.0348	0	-4.5927	-16	-4.4376
9	0.3479	1.7177	3.125	15.9191	15.9191	-16.0232	0	-4.5596	-16	-4.4556
10	0.3461	1.7202	3.125	15.9456	15.9456	-16.0155	0	-4.5375	-16	-4.4676
11	0.345	1.7219	3.125	15.9634	15.9634	-16.0104	0	-4.5228	-16	-4.4757
12	0.3442	1.723	3.125	15.9753	15.9753	-16.007	0	-4.5129	-16	-4.4812
13	0.3436	1.7237	3.125	15.9833	15.9833	-16.0047	0	-4.5062	-16	-4.4848
14	0.3433	1.7242	3.125	15.9888	15.9888	-16.0032	0	-4.5017	-16	-4.4873
15	0.343	1.7246	3.125	15.9924	15.9924	-16.0021	0	-4.4987	-16	-4.489
16	0.3429	1.7248	3.125	15.9949	15.9949	-16.0014	0	-4.4966	-16	-4.4901
17	0.3428	1.725	3.125	15.9965	15.9965	-16.001	0	-4.4953	-16	-4.4909
18	0.3427	1.7251	3.125	15.9977	15.9977	-16.0007	0	-4.4943	-16	-4.4914
19	0.3426	1.7251	3.125	15.9984	15.9984	-16.0004	0	-4.4937	-16	-4.4917
20	0.3426	1.7252	3.125	15.9989	15.9989	-16.0003	0	-4.4933	-16	-4.4919
21	0.3426	1.7252	3.125	15.9993	15.9993	-16.0002	0	-4.493	-16	-4.4921

TABLE 4. Results of the cutting area of the corresponding bars after optimization

Loop	Results of variables (cm^2)			Stresses of bars (kN/cm^2)							$F(cm^3)$
	X1	X2	X3	1	2	3	4	5	6	7	
0	10	20	30	0.8443	0.8443	-1.1708	0	-0.597	-1.6667	-0.2705	46971
1	0.0864	24.5753	22.8437	2.4895	2.4895	-1.4263	0	-0.0124	-2.1888	-1.0756	41983
2	6.1713	19.3232	23.5126	1.1914	1.1914	-1.2916	0	-0.5381	-2.1265	-0.4379	40004
3	0.0864	25.4037	23.5227	2.4136	2.4136	-1.3801	0	-0.0116	-2.1256	-1.0451	43342
4	6.5229	19.1767	23.5227	1.1549	1.1549	-1.2881	0	-0.5555	-2.1256	-0.4223	40052
5	0.0864	25.2642	23.5227	2.421	2.421	-1.3877	0	-0.0117	-2.1256	-1.045	43182
6	6.4765	19.2024	23.5227	1.1595	1.1595	-1.2882	0	-0.553	-2.1256	-0.4243	40054
7	0.0864	25.284	23.5227	2.42	2.42	-1.3866	0	-0.0117	-2.1256	-1.045	43205
8	6.4831	19.1987	23.5227	1.1588	1.1588	-1.2881	0	-0.5534	-2.1256	-0.424	40054
9	0.0864	25.2812	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
10	6.4822	19.1993	23.5227	1.1589	1.1589	-1.2882	0	-0.5533	-2.1256	-0.4241	40054
11	0.0864	25.2816	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
12	6.4823	19.1992	23.5227	1.1589	1.1589	-1.2882	0	-0.5534	-2.1256	-0.4241	40054
13	0.0864	25.2815	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
14	6.4823	19.1992	23.5227	1.1589	1.1589	-1.2882	0	-0.5534	-2.1256	-0.4241	40054
15	0.0864	25.2815	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
16	6.4823	19.1992	23.5227	1.1589	1.1589	-1.2882	0	-0.5534	-2.1256	-0.4241	40054
17	0.0864	25.2815	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
18	6.4823	19.1992	23.5227	1.1589	1.1589	-1.2882	0	-0.5534	-2.1256	-0.4241	40054
19	0.0864	25.2815	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202
20	6.4823	19.1992	23.5227	1.1589	1.1589	-1.2882	0	-0.5534	-2.1256	-0.4241	40054
21	0.0864	25.2815	23.5227	2.4201	2.4201	-1.3868	0	-0.0117	-2.1256	-1.045	43202

4. Conclusion

The Fmincon function was applied to the Matlab program to do the math. After 20 cycles, the value found at the best level for the bars was found to be reasonable and still ensures the right amount of strength and voltage at each point where the steel bars meet. Positioning rates are also determined for each bar at the intersecting nodes, showing the advantage of the optimal method when applying this Fmincon function. The error number obtained through the optimal process is also very low, to four decimal numbers, and therefore the result is very reliable. This results in a promise of optimum computation with more complex, more important real-life problems in the future.

References

Alan R. Parkinson et al. (2013). *Optimization Methods for Engineering Design Applications and Theory*. Brigham Young University. https://apmonitor.com/me575/uploads/Main/optimization_book.pdf

Annegret Burtscher (2020). *Optimization in Geometry and Physics, Course module NWI-WB107*. Radboud University Nijmegen Fall semester 2020/21. https://www.math.ru.nl/~burtscher/lecturenotes/2021OGPnotes.pdf

Patrick Bangert (2012). *Optimization for industrial problems*. Springer, Heidelberg, e-book ISBN: 978-3-642-24974-7. https://doi.org/10.1007/978-3-642-24974-7

Seán Moran (2017). *Process Plant Layout*. Book-Second Edition, ISBN: 978-0-12-803355-5, Copyright © 2017 Elsevier Inc. https://www.sciencedirect.com/book/9780128033555/process-plant-layout#book-info

Vaidyanathan Jayaraman et al. (1998). A joint optimization of product variety and ordering approach. *Computers & Operations Research*, *25*(7–8), 557-566. https://doi.org/10.1016/S0305-0548(98)00010-0

Shaomin Ren (2022). *Optimization of Enterprise Financial Management and Decision-Making Systems Based on Big Data*, *Journal of Mathematics*, Article ID 1708506, 11. https://doi.org/10.1155/2022/1708506

Boris Goldengorin, Sergei Kuznetsov (2023). *Data Analysis and Optimization*. Springer Nature Switzerland AG, eBook ISBN: 978-3-031-31654-8, 422. https://doi.org/10.1007/978-3-031-31654-8

Sunil Kumar, Yasir Rizvi (2018). A review of modelling and optimization techniques in turning processes. *International Journal of Mechanical Engineering and Technology* (IJMET), *9*(3), 1146-1156.https://iaeme.com/MasterAdmin/Journal_uploads/IJMET/VOLUME_9_ISSUE_3/IJMET_09_03_118.pdf

Haleemah Jawad Kadhim *et al*. (2021). A New Technique for Finding the Optimal Solution to Assignment Problems with Maximization Objective Function *.J. Phys.: Conf. Ser.* 1963, 012104 https://iopscience.iop.org/article/10.1088/1742-6596/1963/1/012104

Mark A. Schulze (1998). *Linear Programming for Optimization ,Perceptive Scientific Instruments*. Inc, 1-8 https://www.researchgate.net/publication/2420905_Linear_Programming_for_Optimization

Thomas L. Magnanti (2013). Network Optimization. *Encyclopedia of Operations Research and Management Science*,1026-1032. DOI:10.1007/978-1-4419-1153-7_664

Yanchi Li, Wenyin Gong, Shuijia Li (2023). Multitasking optimization via an adaptive solver multitasking evolutionary framework, Information Sciences, 630, 688-712 https://doi.org/10.1016/j.ins.2022.10.099

Jigar L et al. (2018).Optimization of mechanical design problems using advanced optimization technique. *IOP Conf. Ser: Mater. Sci. Eng.* 455 012091 https://iopscience.iop.org/article/10.1088/1757-899X/455/1/012091/pdf