VIRTUAL MANUFACTURING PROCESS ALLOCATION VIA NONLINEAR OPTIMIZATION APPROACH

Kasim M. Daws [1], Zuhair I. Ahmed [2], Mohamed S. Ali [3]

[1] Professor, [2] Assistant Professor, [3] Assistant Lecturer, Mechanical Eng. Dept., University of Baghdad

E-mail: Kasim_daws@yahoo.com [1], ziaa04@yahoo.com [2], msa53@yahoo.com [3]

ABSTRACT: - This study demonstrates the evolution of optimal virtual manufacturing process via utilizing the condition of optimization tools to determine the optimum configuration of the drilling process and operations of allocation in term of distances constrains through availability of two dimensional space layout.

The distribution of motherboard holes allocated was virtually investigated and an experimental of power supply plate cover is also maintained to achieve different minimum dimension which leads to alter the whole design parameters of sub components such as integrates and electric components in the main board circuit.

A technique for mentioned case study is a new branch technology which will affect simultaneously many cases design attributes in the same concept such as machines allocation distribution in the plant shop floor, assignment problem, industrial geographical distribution salesman problem, perforation and transportation and other.

After all iteration took place an ultimate final allocation for drilling holes in an 1.5 x1.5 cm metal cover already perforate in specific locations to perform its design need where the new distribution of the holes give an optimal space for metal cover which effect in the hole design prospect in addition to other production aspects of cost and raw material west issues.

Keyword: Virtual Manufacturing, Optimization, Drilling, Holes Allocation, Iterations.

INTRODUCTION

Surface vectors to the z-direction, restricting verification to 3-axis milling for example as a system if considered as a drilling case to find the allocation of holes in specific plain or surface as large scale problem minimization that would be applied in manufacturing. Also localizing surface points based on a pre-processing curvature analysis and the tool position are essential constrain. Employing a user-specified value to set the desired accuracy as opposed to the selected unique new brand method in this particular domain as it will be described later in this research. This allows verification of all of the holes distribution since has considered the same issue for many manufacturing cases being machined and to make graphical output in independent post-simulation process. The graphics, output provides visual verification using colors to indicate areas containing holes and excess material.

Analytical method is described by Eleanor [1] this method uses a similar concept to that described by Maghsud Solimanpur and Amir Jafari [2] the surfaces are created by calculating the intersection of tool movements with vectors passing through points on the part surface. There is however important differences between these two methods, the Eleanor limit the condition of constraining while the others find the optimism allocation under unconstrained condition.

Determination of the best configuration which indicates the optimum result will be establish as new technique and not familiar with this kind of manufacturing skill and has not been investigated perversely as an attempt to obtain new ambiguous brand technology that

will be more clear for researchers in the experimental progressing of this research and we prefer not to sever involve with survey of analytical method and focusing on the artificial intelligent using MATLAB programming in order not to disperses out of the selective scope.

EXPERIMENTAL METHODOLOGY

Motherboard drilling station for instance, which is huge dedicated machine manipulating large copper metal sheets have all their holes drilled out. Since there are motherboards for various designs so that the number of holes and the distribution of the holes and gaps should be allocated in different ways as holes differ in sizes and/or length on a motherboard manufacturing so that numerous drills are needed to drill just a single motherboard model. Figure (1) shows a close up of the metal sheets indicating the different holes being drilled in a small area.

For experimental work an optimal distribution of drilling holes in a specific scale region is required so that a relatively small number of iteration would be needed to find the shape allocation of the drilling or in some time the perforating of the sheet metal where particular gap and holes square or rectangular or whatever hole geometry are needed for the application production Figure (2).

As an example the (15cm length and 15cm width) of a cutting cover in a device as experimental case to modify its configuration surface shape according to the proposed optimization methodology. A remarked red magic pen on determined ten locations which represent the stamping components in the ground of the device and after measuring the distances between them in the center flouting geometry and optimizing the differences we achieve correction was achieved in the total dimensions of the entire plate metal that can be suggested to manufacturing responses Figure (3).

In order to implement the case study investigated in the previous section corresponding with the methodology we consider a sheet metal with the same specification but in exact two dimension scales (length and width) to overcome the distinguished ten red marks with the software program to find the best optimum allocation distribution after specific iteration considering stepping in Figure (4) as following:

 1- The numbers of selective drilling holes are equal to ten which remarked in red point in Figure (3).
 2- The numbers of connection are equal to seventeen which represent the relationship length between each per of points.
 3- All drilling hole diameters will be neglected (out of this research domain).
 4- All scalars are divided by scalar fifteen (distances are manipulated by 15 to get the fracture number for programming purpose.
5- Three holes (points) were considered fixed at the initial condition to achieve non-redundancy.

The corresponding connection measurements – where for example (1, 2) representing hole1 and its corresponding relation to hole 2 where (0.5000) is the distance between them in specific scale.

(1,2)= 0.5000
   (1,3)= 0.5000
   (2,3)= 0.7000
 (2,6)= 0.6883
   (3,8)= 0.4182
   (4,10)= 0.3200
   (5,3)= 0.4000
   (5,7)= 0.3935
(6,3)= 0.5545
   (6,7)= 0.3756
   (7,8)= 0.6368
   (8,2)= 0.5756

(9,3)= 0.2716
(9,4)= 0.3503
(10,2)= 0.3447
(5,10)= 0.3234
(6,1)= 0.6361

Trust-Region Methods for Nonlinear Minimization

Many of the methods used in Optimization solvers are based on *trust regions,* a simple yet powerful concept in optimization.

The unconstrained minimization problem, minimize f(x), where the function takes vector arguments and returns scalars. Suppose a point x in *n*-space and you want to improve, i.e., move to a point with a lower function value. The basic idea is to approximate f with a simpler function q, which reasonably reflects the behavior of function f in a neighborhood N around the point x. This neighborhood is the trust region. A trial steps is computed by minimizing (or approximately minimizing) over *N*.

Min {q(s), s ϵ N}   (1)

S where (s) is two-dimensional subspace belong to N points.

The current point is updated to be x + sif f(x + s) < f(x); otherwise, the current point remains unchanged and *N*, the region of trust, is shrunk and the trial step computation is repeated.

The key questions in defining a specific trust-region approach to minimizing f(x) are how to choose and compute the approximation q (defined at the current point x), how to choose and modify the trust region *N*, and how accurately to solve the trust-region sub problem.

In the standard trust-region method [4], the quadratic approximation q is defined by the first two terms of the Taylor approximation to F at *x*; the neighborhood N is usually spherical or ellipsoidal in shape. Mathematically the trust-region sub problem is typically stated

$\text{Min}\{1/2\ s^T Hs + s^T g$ such that $\|D\| \leq \Delta\}$ (2)

Where g is the gradient of at the current point *x*, H is the Hessian matrix (the symmetric matrix of second derivatives), D is a diagonal scaling matrix, Δ is a positive scalar, and $\| \ \|$ is the 2norm.

Such algorithms provide an accurate solution to Equation 2. However, they require time proportional to several factorizations of *H*. Therefore, for large-scale problems several approximation and heuristic strategies is needed, based on Equation 2, The approximation approach followed is to restrict the trust-region sub problem to a two-dimensional subspace S ([5] and [6]). Once the subspace S has been computed, the work to solve Equation 2 is trivial even if full eigenvalue/eigenvector information is needed (since in the subspace, the problem is only two-dimensional). The dominant work has now shifted to the determination of the subspace.

The two-dimensional subspace S is determined with the aid of a preconditioned conjugate gradient process. The solver defines S as the linear space spanned by s1 and s2, where s1 is in the direction of the gradient *g*, and s2 is either an approximate Newton direction, i.e., a solution to

H .s2 = -g   (3)

ora direction of negative curvature,

$S2^T .H . s2 < 0$ (4)

The philosophy behind this choice of S is to force global convergence (via the steepest descent direction or negative curvature direction) and achieve fast local convergence (via the Newton step, when it exists).

A step of unconstrained minimization using trust-region ideas is now easy to give:

1- Formulate the two-dimensional trust-region sub problem.
2- Solve Equation 2 to determine the trial steps.
3- If f(x + s) < f(x), then x = x + s.
4- Adjust Δ.

These four steps are repeated until convergence. The trust-region dimension Δ is adjusted

according to standard rules. In particular, it is decreased if the trial step is not accepted, i.e., f(x + s) ≥ f(x). [5] , [7].

ITERATION OF HOLES ALLOCATION PROGRESS

Distribution of holes from the initial design distances with no speculation of its configuration shape achieved after fourteen iterations for the substitution sheet material have the same X, Y dimension with respect to teen drilling holes. Each iteration consisted of optimality progress per iteration, PCG (Preconditioned Conjugate Gradient Method) iterations per iteration of the large scale optimal minimization, curvature of current direction and trust region radius which are the most important step in the optimization feasibility area of the nonlinear large scale case like the one in the random drill allocation of the cover plate. Figures (5) are showing some of the iterations of PCG and trust regionrespectively initiate using matlab program and figure (6) showing the progressing of the manipulation and figure (7) is the end of the iterations.

ULTIMATE HOLES ALLOCATION CONFIGRURATION

After last iteration the optimum drilling holes distribution within the 15x15cm sheet metal specimen where the target distances of determining design were achieved, figures (8) showing the initial and final allocation respectively while the figure(9) showing the comparison between the initial and final position of the drilling after scale modification for more clarity on Y and X axis.

DISSCUSION AND CONCLUSION

This procedure suggests a new methodology which has not been used in drilling operations to estimate the dimension of the sheet metal to be cut for a particular design as inverse engineering retrieval data at any manufacturing operation and that's what this new idea gives, here the cause that wasting, removal and initial raw material can be reduced or estimated correctly. If it is not knowing the allocation of perforation or drilling in addition to material handling distances we can use this approach of optimization and applied by using (matlab2010 b) program and this is a permeation invitation after contacting the authors to use the code of the program in any desired application for future work where figure (9) clearly showing the reduction in dimension in addition to pre given distance of hole where no regular computation could achieved the distribution condition under these specific distances or dimensions.

APPENDIX
PROGRAM MANIPULATION

The code was generated as HTML script for more
clearance as follow:
% " program with 10 drilling with 17 connections
realationship"
%N MEANS THE TOTAL NUMBER OF THE
drilling
N=10;
% S MEANS THE MATREX OF N * N THAT HAS
NONZERO DISTANCE BETWEEN CONNECTED
POINTS
% number of total connections is 17 which they are=
%(I,J)= connection distance between position I and
position J
% (1,2)= 0.5000
% (1,3)= 0.5000
% (2,3)= 0.7000
% (2,6)= 0.6883
% (3,8)= 0.4182

% (4,10)= 0.3200
% (5,3)= 0.4000
% (5,7)= 0.3935
% (5,10)= 0.3234
% (6,1)= 0.6361
% (6,3)= 0.5545
% (6,7)= 0.3756
% (7,8)= 0.6368
% (8,2)= 0.5756
% (9,3)= 0.2716
% (9,4)= 0.3503
% (10,2)= 0.3447
S=[0 0.5000 0.5000 0 0 0 0 0 0 0 ;
0 0 0.7000 0 0 0.6883 0 0 0 0 ;
0 0 0 0 0 0 0 0.4182 0 0 ;
0 0 0 0 0 0 0 0 0 0.3200 ;
0 0 0.4000 0 0 0 0.3935 0 0 0.3234 ;
0.6361 0 0.5545 0 0 0 0.3756 0 0 0 ;
0 0 0 0 0 0 0 0.6368 0 0 ;
0 0.5756 0 0 0 0 0 0 0 0 ;
0 0 0.2716 0.3503 0 0 0 0 0 0 ;
0 0.3447 0 0 0 0 0 0 0 0];
%xstart=(2*N-6,1); " MEANS THE INITIAL
POSITIONS OF THE POINTS"
xstart=[0.6771
0.5740
1.1845
0.5464
0.7985
1.0675
1.3072
0.4930
0.9752
1.4623
0.1140
0.8805
0.6208
0.4637];
xstart1=rand(14,1);
energscale =[0.4000 0.4000 0.4000
11
0.5000 0.5000 0.5000
0.6000 0.6000 0.6000
0.7000 0.7000 0.7000
0.8000 0.8000 0.8000
0.9000 0.9000 0.9000
1.0000 1.0000 0.7500
1.0000 1.0000 0.6250
1.0000 1.0000 0.5000
1.0000 0.9375 0.3750
1.0000 0.8750 0.2500
1.0000 0.8125 0.1250
1.0000 0.7500 0

```
1.0000 0.6875 0
1.0000 0.6250 0
1.0000 0.5625 0
1.0000 0.5000 0
1.0000 0.4375 0
1.0000 0.3750 0
1.0000 0.3125 0
1.0000 0 0
1.0000 0 0
1.0000 0 0
1.0000 0 0
1.0000 0 0];
[I,J,dist] = find(S);
subplot(1,1,1);
spy(S+S');
title('Measured distances');
%,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
[val,g,H] = mmole(xstart,S);
spy(H);
title('Structure of Hessian Matrix');
%,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
xstart = [[.25 .25]; [.25 .75]; [.75 .25];
reshape(xstart,N-3,2)];
pts = plot(xstart(:,1),xstart(:,2),'g.');
set(pts,'Markersize',15);
for k = 1:length(dist)
energy(k) = (dist(k)^2 - norm(xstart(I(k),:) -
xstart(J(k),:))^2)^2;
end
colormap(energscale);
numcolors = size(energscale,1);
for k = 1:length(dist)
line([xstart(I(k),1) xstart(J(k),1)],[xstart(I(k),2)
xstart(J(k),2)], ...
'Color',energscale(min(numcolors,max(1,ceil(numcol
ors*energy(k)))),:));
end
title('Initial locations and error distribution');
axis square;
colorbar;
%,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
xstart = reshape(xstart(4:N,:),2*N-6,1);
options = optimset('largescale','on', 'gradobj','on',
'hessian','on', ...
'display','none','outputfcn',@moleculeoutputfcn);
x = fminunc(@mmole,xstart,options,S);
%,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
% Close figures that FMINUNC creates (if they are
still open).
delete(findobj(0,'Name','Algorithm Performance
Statistics'))
delete(findobj(0,'Name','Progress Information'))
% Plot atoms in final positions.
```

```matlab
subplot(1,1,1)
xsol = [ [.25 .25]; [.25 .75]; [.75 .25]; reshape(x,N-
3,2)];
pts = plot(xsol(:,1),xsol(:,2),'g.');
set(pts,'markersize',15);
for k = 1:length(dist)
energy(k) = (dist(k)^2 - norm(xsol(I(k),:) -
xsol(J(k),:))^2)^2;
12
end
maxenergy = max(energy);
colormap(energscale);
numcolors = size(energscale,1);
for k = 1:length(dist)
line([xsol(I(k),1) xsol(J(k),1)],[xsol(I(k),2)
xsol(J(k),2)], ...
'Color',energscale(min(numcolors,max(1,ceil(numcol
ors*energy(k)))),:));
end
title('Final error distribution for pairs in S');
axis([min(xsol(:,1)),max(xsol(:,1)),min(xsol(:,2)),ma
x(xsol(:,2))]);
axis square;
colorbar;
%,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
subplot(1,2,1);
xstart = [ [.25 .25]; [.25 .75]; [.75 .25];
reshape(xstart,N-3,2)];
pts = plot(xstart(:,1),xstart(:,2),'g.');
set(pts,'Markersize',15);
for k = 1:length(dist)
energy(k) = (dist(k)^2 - norm(xstart(I(k),:) -
xstart(J(k),:))^2)^2;
end
colormap(energscale);
numcolors = size(energscale,1);
for k = 1:length(dist)
line([xstart(I(k),1) xstart(J(k),1)],[xstart(I(k),2)
xstart(J(k),2)], ...
'Color',energscale(min(numcolors,max(1,ceil(numcol
ors*energy(k)))),:));
end
title('Initial');
axis square;
colorbar;
subplot(1,2,2);
pts = plot(xsol(:,1),xsol(:,2),'g.');
set(pts,'markersize',15);
for k = 1:length(dist)
energy(k) = (dist(k)^2 - norm(xsol(I(k),:) -
xsol(J(k),:))^2)^2;
end
maxenergy = max(energy);
```

```
colormap(energscale);
numcolors = size(energscale,1);
for k = 1:length(dist)
line([xsol(I(k),1) xsol(J(k),1)],[xsol(I(k),2)
xsol(J(k),2)], ...
'Color',energscale(min(numcolors,max(1,ceil(numcol
ors*energy(k)))),:));
end
title('Final');
axis([min(xsol(:,1)),max(xsol(:,1)),min(xsol(:,2)),ma
x(xsol(:,2))],'square');
colorbar;
%.........................................................................
```

REFERENCES

1. Eleanor Kate Nick. Fuzzy Optimal Allocation and Arrangement of Spaces in Naval Surface Ship Design. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Naval Architecture and Marine Engineering) in The University of Michigan 2008.

2. Maghsud Solimanpur, Amir Jafari. Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm, Computers & Industrial Engineering (2008), doi: 10.1016/j.cie. 2008.01.018.

3. Massoud Bazargan-Lari. Layout designs in cellular manufacturing. European Journal of Operational Research 112 (1999) 258±272. Faculty of Science and Technology, University of Western Sydney - Nepean, P.O. Box 10, Kings Wood, NSW 2747, Australia Received 22 December 1997; accepted 1 April 1998.

4. H. Javanshir, S. R. Seyed-Alizadeh Ganji. Optimal Allocation of Ships to Quay Length in Container Ports. Journal of Industrial and Systems Engineering. Vol. 3, No. 4, pp 275-290 winter 2010.

5. Bernar detta Addis† and Sven Leyffer A Trust-Region Algorithm for Global Optimization. Mathematics and Computer Science. August 2006.

6. Ya-xiang Yuan. A review of trust region algorithms for Optimization. State Key Laboratory of Scienti¯c and Engineering Computing, Institute of Computational Mathematics and Scienti¯c/Engineering Computing, Chinese Academy of Sciences, P. O. Box 2719, Beijing 100080, China.2002.

7. Niclas Börlin. Department of Computing Science Umeå Universityniclas.borlin@cs.umu.se December 3, 2007.

8. Mehrotra, S., "On the Implementation of a Primal-Dual Interior Point Method," SIAM Journal on Optimization, Vol. 2, pp 575–601, 1992.

9. Yang. Shichun. The path layout method of drilling holes based on uniform distribution and its application. Technology and innovation conference .ITIC international 2006.
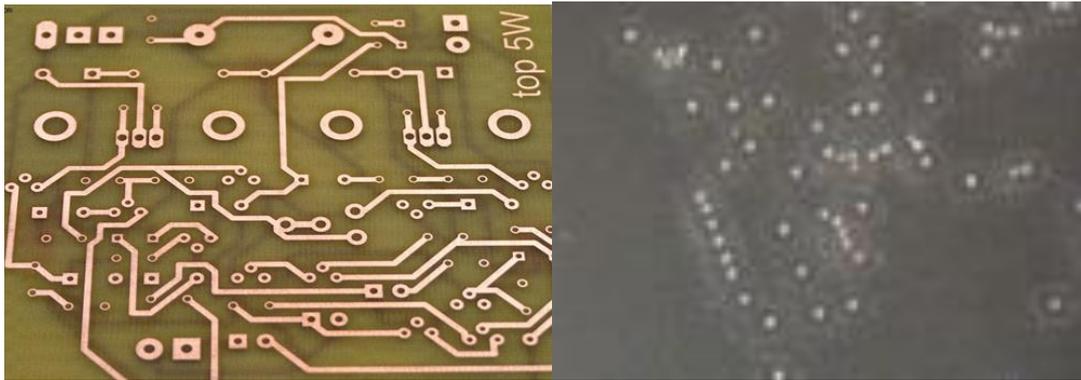
Figure (1): drilling motherboard holes [Ref.]

Figure (2): gasket and sheet metal perforation [Ref.]

Figure (3): device plat cover (15x15cm)
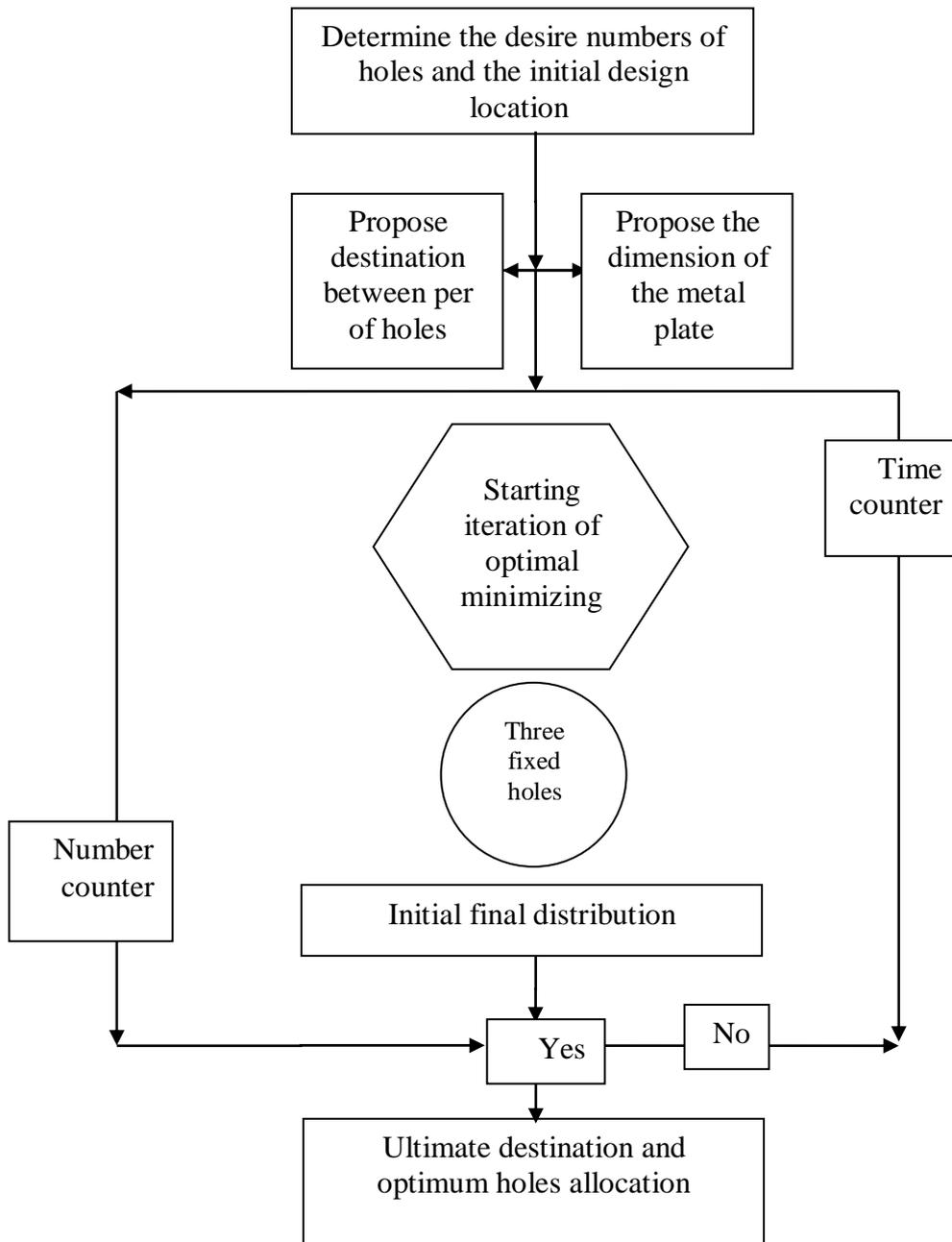
Figure (4): methodology steps
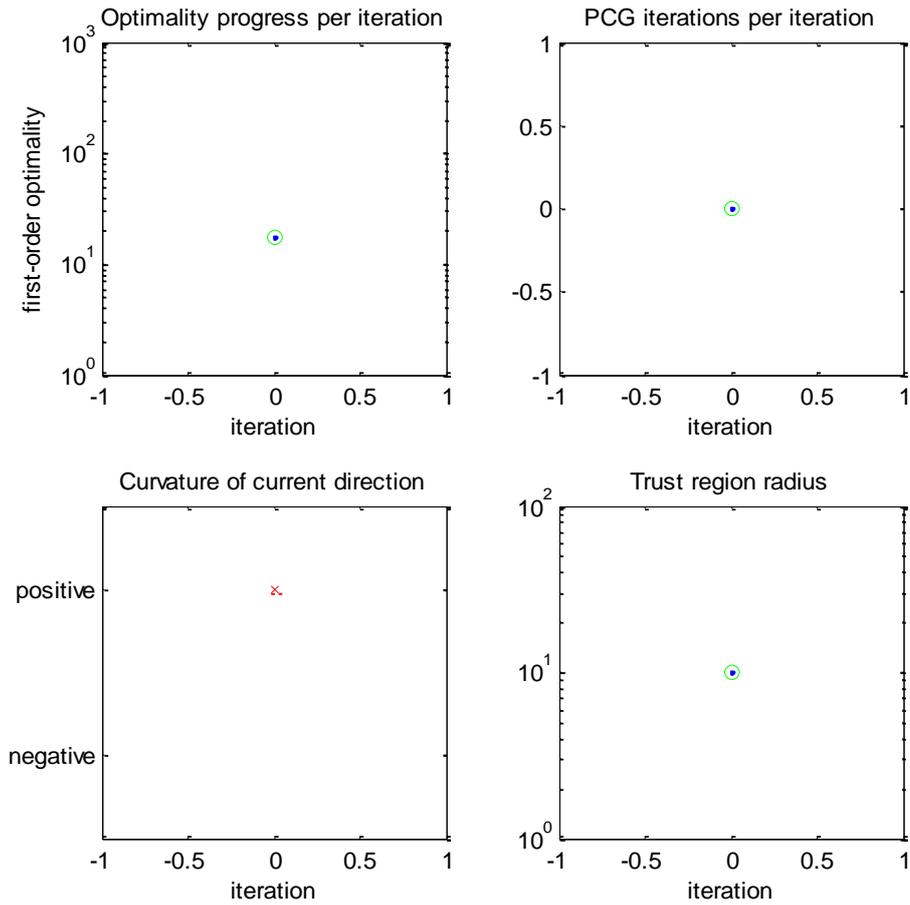
Figure (5): optimization iterations
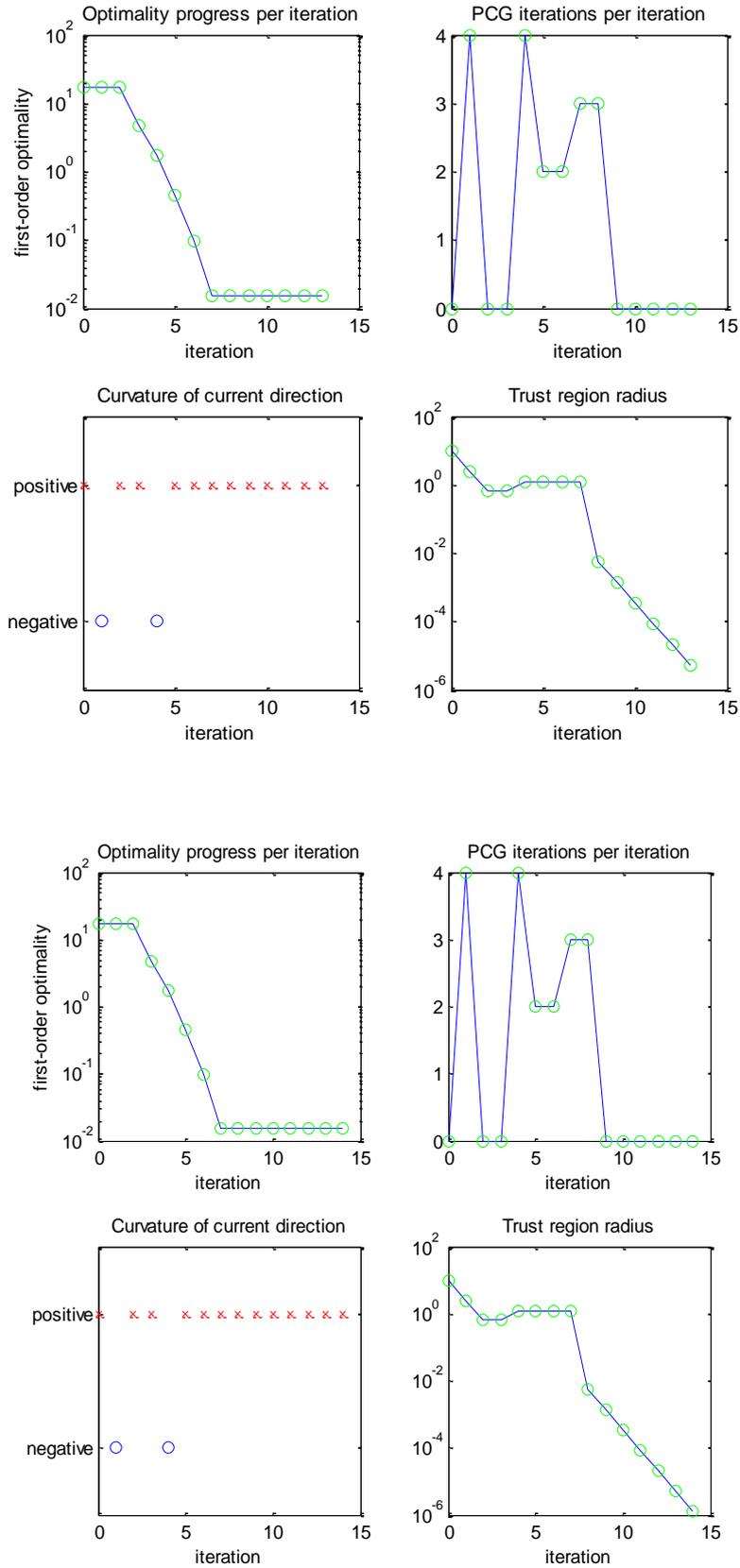
Figure (6): progressing
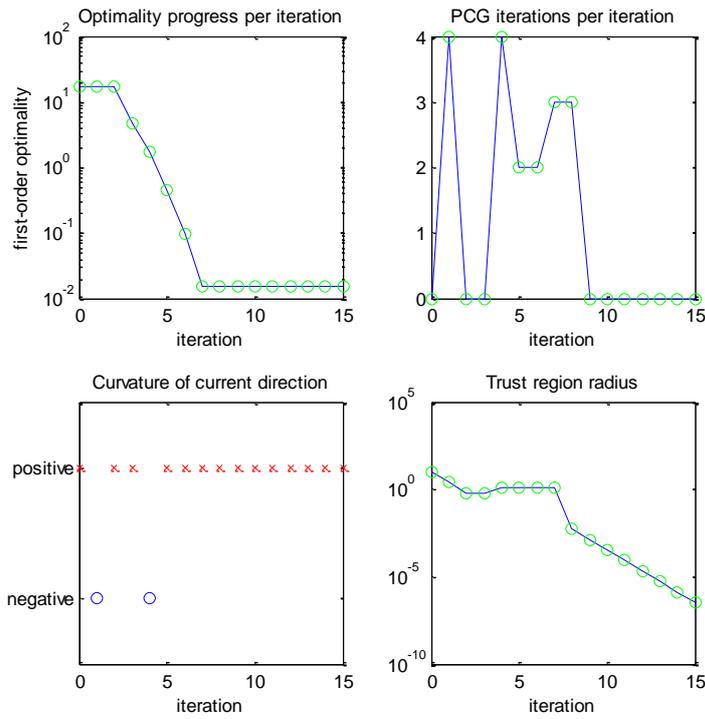
Figure (7): end of iteration
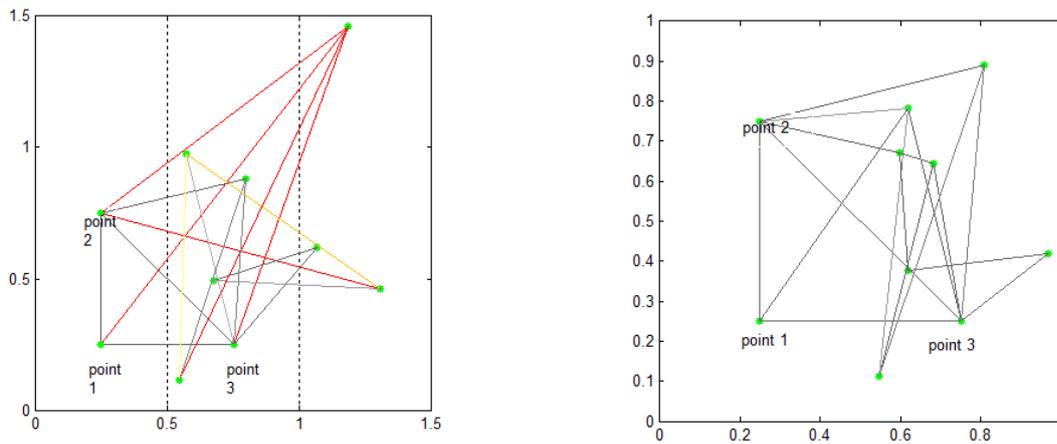
Figure (8): initial and final allocation
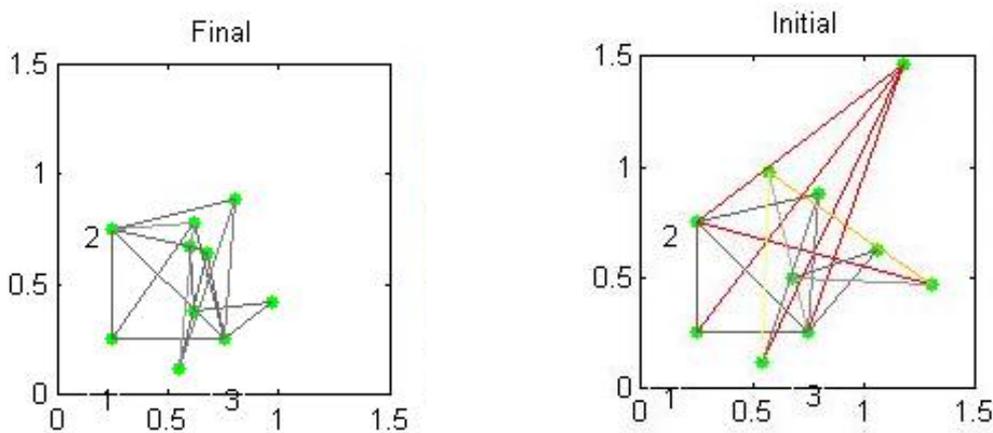
Figure (9): initial and final position of the drilling after scale modification

الخلاصة

في هذا البحث تم استحداث برنامج حاسوبي لتخمين التوزيع المطلوب للكثير من التطبيقات الهندسيه وهذا التوزيع ممكن ان يكون ناتج عن عمليات تثقيب او توزيع مكائن في بيئة العمل ومناولة المواد. الخ وكذلك تخمين ابعاد اللوحات المعدنيه المشغله لتقليص الضياعات وفي الملحق تم وضع شفرة البرنامج لتمكين الباحثين من استخدامه بعد الاشاره الى البحث الموسوم اعلاه.