DIFFICULTIES OF FACILITY LAYOUT PROBLEMS AND SOME RESULTS FOR HEURISTIC ALGORITHMS

Ramazan Yaman[1], Kadriye Ergün[1], Gülşen Yaman[2], Necati Özdemir[3]

[1]ryaman@balikesir.edu.tr, kergun@balikesir.edu.tr
Department of Industrial Engineering, Engineering and Architectural Faculty,
Balikesir University, Cagis Campus, Balikesir, 10145, Turkey
[2]gyaman@balikesir.edu.tr
Department of Mechanical Engineering, Engineering and Architectural Faculty,
Balikesir University, Cagis Campus, Balikesir, 10145, Turkey
[3]nozdemir@balikesir.edu.tr
Department of Mathematics, Faculty of Science & Arts,
Balıkesir University, Balikesir, 10100, Turkey

Abstract: *Generally, heuristic optimization algorithms have been used for facility layout problems. One of the difficulties with these heuristic algorithms is to assess their performances.*

In this study, some case facility layout problems and their optimum results have been presented and their optimum solutions may be used for heuristic algorithms' performances assessments.

Keywords: *Optimization Classification, Discrete Optimization, Facility Layout Design*

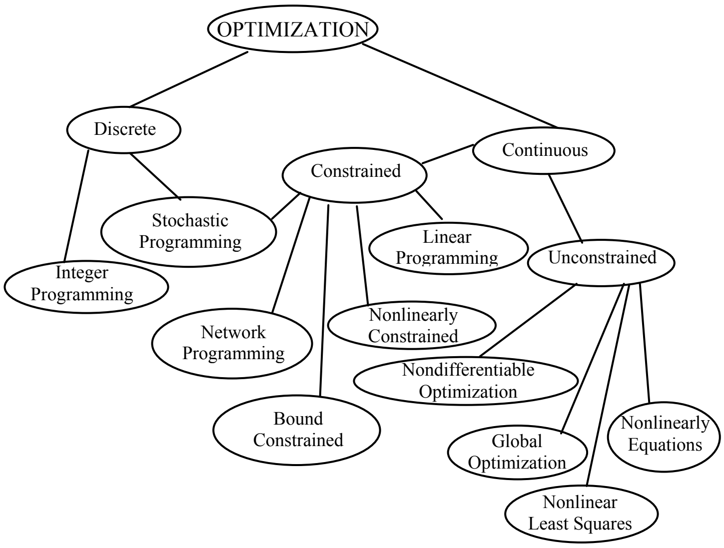
1. Introduction

General optimization problems have been taken attention since early ages. First important achievements have been encountered during 18. Century by Newton, Leibnitz, Lagrange and Cauchy in general mathematical developments (Bal, 1995). These mathematical achievements have been used for well-behaved and defined functions. However, general mathematical approaches cannot handle real life problems with satisfactory solutions. After The Second World War, new numerical approaches have been developed for optimization problems. In these achievements, there are two main reasons which are computing technology and applications of numerical techniques. They have overcome many difficulties of general mathematical approaches.

Optimization problems have been classified with different ways. Figure 1. shows a classification of the optimization problem. This is presented in official web site of the NEOS, (2005). The second classification is from (Rardin, 1998). He represents that the optimization is a whole system, having subsections (see Figure 2.). However, these classifications are not certain and not accepted by the entire optimization milieu.

A classification has also been made by the authors. It has some differences with NEOS classification, because they believe that Discrete Optimization Problems may be considered in the Constrained Optimization (Figure 3.). Because, when a continuous problem has constraints, it will be discrete problem.

As mentioned earlier, other classifications on the optimization problems can be found in related to literature and all of them are free to discussions.



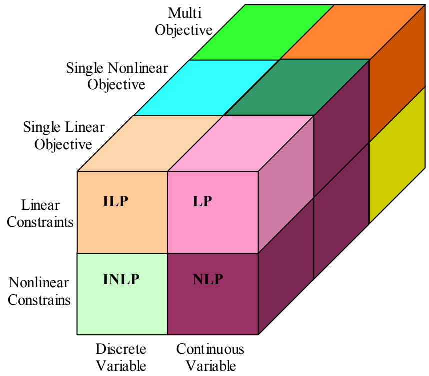


Figure 1. Classification Approach of Optimization Problems by NEOS (NEOS, 2005)

Figure 2. Other Classification of Optimization Problems (Rardin, 1998)

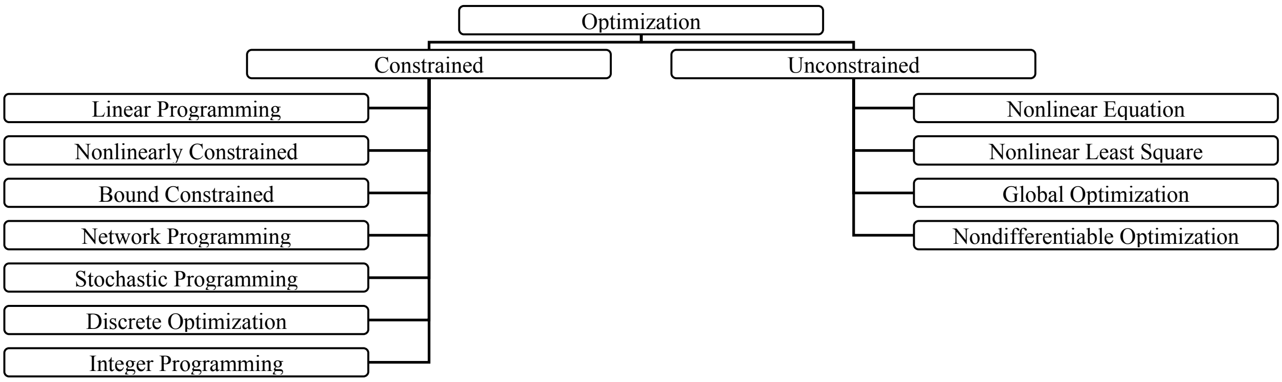


Figure 3. A Simple Classification of Optimization Problems

In classifications, the main point concerns between the problem and researchers. As said before, many real life problems have to be modeled as discrete optimization problems. Because most of the time, resources cannot be divided into a right aggregation.

Discrete optimization problems are difficult to solve. Discrete and multi objective optimization problems had been studied since 1900s and up to 1970s there were a few related to literature in this field. In 1980s, these problems gained popularity by academia and industry again.

In recent years, there are many solution approaches for the field and most of them are heuristics and meta-heuristics. Because exact algorithms cannot still obtain results in reasonable times for real size problems. Discrete Optimization is a branch of Combinatorial Optimization (Parker & Rardin, 1988).

Generally, such problems can be formulated as follows:

$$\min.(\text{or max.})\ f(\mathbf{x}) \tag{1}$$

$$\textit{subject to}\ g_i(\mathbf{x}) \geq b_i\ ;\quad i = 1,\ldots,m; \tag{2}$$

$$h_j(\mathbf{x}) = c_j;\quad j = 1,\ldots,n. \tag{3}$$

Here, $\boldsymbol{x}$ is a vector of decision variables, and $f(.)$, $g_i(.)$ and $h_j(.)$ are general functions (Reeves, 1995).

Many discrete optimization problems are NP-Hard. It is a generally belief that NP-Hard problems cannot be solved to optimality within polynomially bounded computation times. Related explanations to P, NP, NP-Hard, NP- Complete problem can be found in many literature (Parker & Rardin, 1988; Jacobsan & Yücesan, 2002).

In recent years real size Discrete Optimization Problems have been taken consideration by industry and academia. Reasons of this consideration are economic effects of their optimum solutions. In the following parts, one of the Discrete Optimization Problem examples is given in brief forms. Then, the study provides optimum case solutions for the heuristic algorithm to asses their performances.

2. Facility Layout Problems and Their Difficulties

Facility layout designers always want to design effective systems, however the facility layout problems are very difficult to solve. The design may not be optimum. In today's competitive environment and in limited resources, they have been forced to make optimum designs, rather than only to design suitable systems. The facility layout problem is encountered in many diverse areas of everyday life, and it is a fundamental problem for many manufacturing and service organizations (Meller et al., 1999). Before explanation of its structure, placement of this problem in general optimization problems will help understanding of its structure.

The facility layout problem is a well-researched one. However, few effective solution algorithms have been proposed for real size problems. Since it is an NP hard problem, various approaches for small problems and heuristics approaches for the larger problems have proposed (Balakrishnan et al. 2003).

Mathematical approaches were considered first in the early 1960s (Kusiak & Heragu, 1987). They are mainly based on minimisation methods and result in *n!* solutions for *n* different facilities (El-Rayah & Hollier, 1970). Although today's computer technologies are advanced, any attempt to obtain all the alternative solutions are still very time consuming and impractical.

Mathematical approaches can be summarised under four categories comprising;

- quadratic assignment problems,
- quadratic set covering problems,
- linear integer programming problems, and mixed integer programming problems.

The first approach to facility layout problem was developed by Koopmans and Beckman in (1957). They modelled the problem as a **Q**uadratic **A**ssignment **P**roblem (QAP). The method was assigned this name since the objective function is a second degree function of variables while the constraints are linear functions of the variables. This problem has been revised in (Francis & White, 1974) and is stated as follows:

n departments are to be assigned to m available locations. All the assignments must be completed simultaneously, and the arrangement which minimizes the movement cost is selected. If $n \leq m$ let C_{ikjh} be the cost of placing departments i and j at locations k and h, respectively, then $x_{ik} = 1$; if the department is located at k, otherwise $x_{ik} = 0$. The objective is to minimize

$$Z = \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{m} C_{ikjh} X_{jk} X_{jh} \qquad (4)$$

Subject to

$$\sum_{j=1}^{m} X_{jk} = 1 \qquad k = 1,\ldots,m \qquad (5)$$

$$\sum_{i=1}^{n} X_{ik} = 1 \qquad i = 1,\ldots,n \qquad (6)$$

$$X_{i=1} = \{0,1\} \text{ for all } i,k \qquad (7)$$

Many algorithms have been proposed for solving this structure, however, they are feasible for only small-sized problems. Typically if the department count exceeds 15, the difficulty of obtaining a solution increases rapidly.

2.1 Brute Force Algorithm

Brute force solves a class of problems that can be represented as equations with integer solutions. The solution must be from a predefined set of integers (www.delphiforfun.org, 2005).

An algorithm is a step-by-step recipe for solving a problem. A brute force algorithm is one that proceeds in simple and obvious way, but it will require a huge number of steps (perhaps an unfeasible large number of steps) to complete (www.cs.udaho.edu, 2005).

2.2 Heuristic Algorithms

The advantage of an exact algorithm is that it gives an optimal solution. However, the computational complexity of the problem limits the size of problem that can be solved. Heuristic algorithms offer a solution, which is usually sub-optimal but computationally efficient. To date, there are many heuristic procedures (Kusumah, 2001).

The bulk of the literature survey and review of the works up to 1986 has been complied by Kusiak and Heragu in 1987. They also compared some of the used methods and algorithms. The comparison has been based on 12 methods which are classified into three main groups. These main groups include, heuristic construction algorithms, improvement algorithms and hybrid algorithms. They based their comparison on the quality of the solution as a ratio of the objective function value of the solution produced by the algorithm and a lower bound (Yaman et al., 1993).

Various heuristic algorithms have been used to solve facility layout problems. These algorithms include simulated annealing, tabu search, graph theoretic based heuristics, genetic algorithms, cut algorithm and expert systems. Most of the heuristic algorithms are trapped at a local optimal solution. Heuristic algorithms can be grouped under two headings which are construction and improvement algorithms. They generally do pairwise exchanges and need an initial solution. Stopping criteria of these improvements algorithms is number of iterations or a given CPU time. Most of the heuristics until 1987 had been investigated by Kusiak and Heragu (1987). Some of the well known algorithms are CRAFT, COFAD, COSFAD, GRASP, OFFICE, PREP, SET. Some well known construction algorithms are; ALDEP, CORELAP, PLANET, LAYOPT, CASS, COLO2, COMP2, COMSBUL, DOMINO, GENOPT, IMAGE, KONUVER, LAYADAPT, LSP, MUSTLAP2, PLAN, RMA, SISLAP, SUMI (Erkut & Baskak, 1996).

Following case studies present some results for heuristic algorithms performance assessments. Because, the heuristics' performances are dependent on their structure, stopping criteria, and the problem values.

3. Case Studies for Facility Layout Optimization Problems

To illustrate difficulties of the problem, six facility layout problems have been set up and their optimum solutions and required CPU times for the optimum solutions have been presented as follows.

The facility layout problem has six variables according to the Table 1. number of products, number of work centers, sequence of process for each product, work center sizes, product volumes in a period, work center shapes (Yaman et. al., 1993).

For this study some of the variables have been fixed and given values; number of products is 5 and number of work center size is 1 unit square. The blank department location is fixed for the convenience of the solutions. The volume profile of the products and their associated production plans are set out in the Table 1.

Table 1. Proposed production plans and volumes for each product in problem size 5

Product Number	Proposed (Problem Size 5) Production Plan					Volumes
1	1	2	3	4	5	40
2	1	3	2	4	5	20
3	1	3	2	3	5	60
4	1	3	4	2	5	30
5	1	2	5	3	5	50

Product Number	Proposed (Problem Size 8) Production Plan								Volumes
1	1	3	2	4	6	8	7	5	30
2	1	2	3	5	4	5	6	8	40
3	1	2	3	4	5	6	7	8	60
4	1	4	3	2	6	5	8	7	80
5	1	3	2	6	4	3	2	8	10

Product Number	Proposed (Problem Size 6) Production Plan						Volumes
1	1	3	2	4	3	6	50
2	1	2	3	5	4	2	30
3	1	4	3	5	6	4	60
4	2	3	4	5	1	6	40
5	1	2	3	5	4	6	70

Product Number	Proposed (Problem Size 9) (Yaman et al., 1993) Production Plan									Volumes
1	1	3	5	7	2	7	9			70
2	1	4	2	5	6	8	9			60
3	1	5	7	8	5	6	2	9		40
4	1	2	4	6	7	8	2	3	9	50
5	1	7	6	4	2	8	3	5	9	30

Product Number	Proposed (Problem Size 7) Production Plan							Volumes
1	1	3	2	4	3	5	6	70
2	1	2	3	4	5	2	2	60
3	1	2	3	4	5	6	7	40
4	2	1	3	6	4	3	2	50
5	1	3	5	7	6	2	4	30

Product Number	Proposed (Problem Size 10) Production Plan										Volumes
1	1	3	2	5	6	7	4	9	7	10	40
2	4	3	2	1	6	2	4	5	6	8	60
3	1	2	3	4	5	6	7	8	9	10	80
4	1	3	2	6	4	5	6	4	6	7	70
5	2	3	4	5	6	7	8	9	10	1	30

For these problems, all the possible solutions have been generated by a brute force algorithm. Figure 4. illustrates required CPU times and layout problem sizes. Figure 5. presents the optimum layouts.

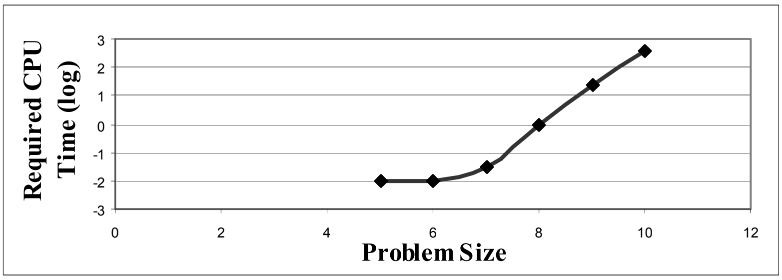


Figure 4. Required CPU times and problem sizes.

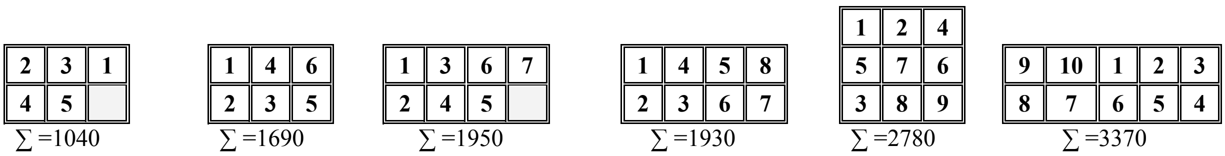


Figure 5. Optimum layouts and their values for the problems

These results have been obtained in IBM compatible PC with Intel Pentium IV, 3 Ghz, 256 MB RAM and Windows XP Operating System. For 10 departments case results have gathered in four different PC's for a reasonable time (max 148 minute with parallel process). For these size problems (10 or more), by dividing the loops and increasing a number of the parallel machines, required CPU times can be decreased. If anyone is interested with details of the problems, authors would like to convey all the details.

4. Conclusions

From this study, it can be concluded that classification of optimization problems and optimization solution algorithms help to understand importance of the optimization problems and their achievements.

It can also be concluded that real size discrete and facility layout problems are taken consideration of academia and industry. Generally, heuristics and meta-heuristics approaches have been employed these problems.

The current case studies present increment of required CPU times with problem sizes. The CPU time increases in non-polynomial structure. The problems and their optimum solutions may be used for performance assessment of the heuristic algorithms.

A parallel processing approach for the optimal solutions on the higher size problems (11,12, 13,…) can be carried out in further researches.

References

1. Bal, H. (1995). Optimizasyon Teknikleri, Gazi Yayınevi, Ankara.
2. Balakrishnan J., Cheng C.H. & Wong K.F. (2003). FACOPT: a user friendly FACility layout OPTimization system, *Computer & Operations Research*, 30, 1625-1641.
3. El-Rayah, T.E., & Hollier, R.H. (1970). A Review of Plant Design Techniques, *International Journal of Production Research,*8(3), 263-279.
4. Erkut H., & Baskak, M. (1996). Stratejiden Uygulamaya Tesis Tasarımı, İrfan Yayımcılık, İstanbul.
5. Francis, R.L., & White, J.A. (1974). *Facilities Layout and Location an Analytical Approach*, Prentice-Hall, Englewood Clifs, NJ.
6. Jacobsan, S.H., & Yücesan, E. (2002). Common Issues in Discrete Optimization and Discrete-Event Simulation, *IEEE Transactions on Automatic Control,* 47(2), 341-345.
7. Koopmans, T.C., & Beckman M. (1957). Assignment Problems and the Location of Economic Activities, *Econometria*, 25, 53-76.
8. Kusiak, A., & Heragu, S.S. (1987). The Facility Layout Problem, *European Journal of Production Research*, 29, 229-251.
9. Kusumah, Y.S. (2001). *Graph Theoretic Based Heuristics for the Facility Layout Design Problem.* Ph.D. thesis. Western Australia, Curtin University of Technology.
10. Meller R.D., Narayanan V. & Vance P.H. (1999). Optimal Facility Layout Design, *Operations Research Letters*, 23, 117-127.
11. NEOS Guide Optimization Tree, http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/index.html
12. Parker, R. G. & Rardin, R.L. (1988). *Discrete Optimization,* Academic Press Inc, USA.
13. Rardin, R.L. (1998). *Optimization in Operation Research,* Prentice-Hall, USA.
14. Rayward-Smith, V. J., Osman, I.H., Reeves, C.R., & Smith, G.D. (1996). *Modern Heuristic Search Methods,* John Wiley and Sons, New York.
15. Reeves, C.R. (1995). *Modern Heuristic Techniques for Combinatorial Problems*, Mcgraw-Hill Book Company, London.
16. Yaman R., Gethin, D.T., & Clarke, M. J. (1993). An effective sorting method for facility layout construction, *Int. J. Prod. Res,* 31(2), 413-417.
17. http://www.delphiforfun.org/Programs/brute_force.htm
18. http://www.cs.uidaho.edu/~casey931/mega-math/gloss/compute/brute.html