A Non-greedy Local Search Heuristic for Facility Layout Problem

Rajesh Matai[1], Surya Prakash Singh[2], and Murari Lal Mittal[3]

[1] Management Department, BITS Pilani, India
[2] Department of Management Studies, IIT Delhi, India
[3] Department of Mechanical Engineering, MNIT Jaipur, India
{rajeshmatai73,surya.singh,mlmittal.mnit}@gmail.com

Abstract. This paper proposes a non-greedy local search heuristic for solving facility layout problem. The proposed heuristic works on non-greedy systematic pair wise exchange of two facilities, that is 2-exchange local search based on non-greedy strategy. Pair wise exchanges are accepted if the objective function value after the exchange is lowered or smaller than the average objective function increment divided by an intensity factor. Proposed heuristic is tested on commonly used Nugxx series problems and computational results show efficiency and effectiveness of proposed heuristic.

Keywords: Local search, Facility Layout Problem, Quadratic Assignment Problem.

1 Introduction

The facility layout problem (FLP) is an important problem of industrial engineering and well researched problem. It was first formulated as quadratic assignment problem (QAP) by Koopmans and Beckman [5]. Later, Sahni and Gonzalez [8] showed QAP is a NP-complete problem. To achieve global optimum for QAP branch-and-bound, cutting planes or combinations of these methods, like branch-and-cut and dynamic programming are used. However, results by these exact algorithms are modest. Diponegoro and Sarker [2] reported that instances of the QAP of sizes larger than 20 cannot be solved optimally in a reasonable computational time. Therefore, interest of researchers and practitioners lies in the application of heuristics and meta heuristics approaches to solve QAP. Some of the well known heuristic approaches applied in the past and available in literature are CRAFT, HC-66, ALDEP, CORELAP, SABLE etc. But the performance of these heuristics is good only for small or moderate sized problems. As the problem size increases the solution quality decreases. In addition of applying heuristics, now-a-days meta-heuristic approaches like Simulated Annealing (SA), Tabu Search (TS), Ant Colony Algorithm (ACO) and Genetic Algorithm (GA) are also widely applied to solve FLP.A good amount of work on FLP can be found in Kusiak and Heragu [6], Heragu [4], Heragu and Kusiak [3], Singh and Sharma [9] and Matai *et al.* [7]. This paper proposes a local search heuristic based on non-greedy strategy for solving the FLP.

V.V. Das, E. Ariwa, and S.B. Rahayu (Eds.): SPIT 2011, LNICST 62, pp. 176–181, 2012.

2 Problem Formulation

Consider the problem of locating 'n' facilities in 'n' given locations. Each location can be assigned to only one facility, and vice versa. F_{ik} is the flow between facilities 'i' and 'k', and D_{jl} is the distance between locations 'j' and 'l'. The FLP has been formulated as follows:

$$\text{Min TF} = \sum_{\substack{i=1\\ i\neq k}}^{n}\sum_{\substack{j=1\\ j\neq l}}^{n}\sum_{k=1}^{n}\sum_{l}^{n} F_{ik} * D_{jl} * X_{ij} * X_{kl} \tag{1}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \forall\ \mathrm{j} = 1,......,\mathrm{n} \tag{2}$$

$$\sum_{j=1}^{n} X_{ij} = 1 \quad \forall\ \mathrm{i} = 1,...,\mathrm{n} \tag{3}$$

$$X_{ij} \in \{0,1\} \quad \forall\ \mathrm{i,j} = 1,...,\mathrm{n} \tag{4}$$

$X_{ij} = 1$ if facility $'i'$ is located/assigned to location $'j'$ and $X_{ij} = 0$ if facility $'i'$ is not located/assigned to location $'j'$, where $'n'$ is the number of facilities.

3 Non-greedy Local Search Heuristic

The proposed heuristic works on non-greedy systematic pairwise exchange of two facilities in the neighbourhood locations rather than exchange of facilities randomly. The neighbourhood NB(r) is defined as all assignments that can be reached from a given assignment when r elements are exchanged. For pairwise exchange algorithms, the size of a neighbourhood is [NB(2)]= $\frac{1}{2}n(n-1)$. Two alternatives exist for neighbourhood search process. First, choose the next potential facility for exchange randomly. Second, explore the neighbourhood in a systematic way having all the possible exchange elements ordered ("shuffled"). The precise order is irrelevant, it is only essential that the neighborhood is explored thoroughly. Proposed heuristic uses later approach that is systematic (ordered) neighbourhood search and this systematic neighbourhood search is based on the non-greedy strategy. The key idea is, pair wise exchanges are accepted if the objective function value after the exchange is lowered or smaller than the average objective function increment divided by an intensity factor. The average objective function increment divided by an intensity factor is called threshold value and evaluated from equation (5) given below.

$$\tau = \left(\frac{\sum \Delta Z^{+}}{C}\right) \Big/ \varepsilon \tag{5}$$

$$\Delta Z = Z(P_c) - Z(P_i) \tag{6}$$

$$\Delta Z^{+} = \Delta Z \text{ if } \Delta Z > 0 \tag{7}$$

Where, P_i is the initial random solution and P_c is current solution after pairwise exchange of two facilities and ΔZ is the current difference of the objective function value of current solution ($Z(P_c)$) and initial random solution ($Z(P_i)$). ε is the intensity factor which lies in the interval of [0,1]. C is the total number of increments of objective function value (i.e. when $\Delta Z > 0$) or number of times when objective function value of current solution is more than old solution after pairwise exchange. From equation (5) it can be seen that τ is the average objective function increment (in C number of increments) divided by the factor (ε). The acceptance rule can be finally defined as: the solution in the heuristic during pair wise exchange is accepted if and only if equation (8) does hold. Table 1 shows few other acceptance rules used and available in the literature.

$$\Delta Z < 0 \text{ or } \Delta Z < \left(\frac{\sum \Delta Z^{+}}{C} \right) \Big/ \varepsilon \tag{8}$$

Table 1. Decision Rule: Comparison of Greedy descent and SA with the Proposed Heuristic

Algorithm	**Rule (Conditions)**
Greedy descent	$\Delta Z < 0$
Simulated Annealing	$\Delta Z < 0$ or $random[0,1] < e^{-\Delta Z / t}$
Non-Greedy Local Search	$\Delta Z < 0$ or $\Delta Z < \left(\frac{\sum \Delta Z^{+}}{C} \right) \Big/ \varepsilon$

The main feature of SA is its ability to escape from local optimum. If the current solution (P_c) has an objective function value smaller (supposing minimization) than that of the initial solution (P_i), then the current solution is accepted. Otherwise, the current solution can also be accepted if the value given by the Boltzmann distribution shown in equation (9) is greater than a uniform random number in [0,1], where t is the 'temperature' control parameter. The solution in the SA during pair wise exchange is accepted if equation (10) does hold.

$$e^{-\Delta Z / t} \tag{9}$$

$$\Delta Z < 0 \text{ or } random[0,1] < e^{-\Delta Z / t} \tag{10}$$

Proposed heuristic also try to escape from local optimum using non-greedy approach, however acceptance rule of candidate solution in proposed heuristic is different [equation (8)] than SA [(equation (10)]. In SA 'temperature' is the control parameter however in proposed local search heuristic 'intensity factor' is the control parameter. We present a pseudo code in figure 1 to better understand the proposed approach.

Input: n, P_i , K, Iterations, ε
/n- number of facility or problem size/
/ P_i – initial random solution for proposed heuristic/
/K- maximum number of trials of the size of neighbourhood /
/Iterations- number of times heuristic is run/
/ ε - initial value of the intensity factor, ε [0, 1] /
$P_b = P_i$ /best solution treated so far/
$h_\varepsilon = \dfrac{\varepsilon}{\max\left(1, \dfrac{Kn(n-1)}{2} - 1\right)}$ /intensity *quantum/*
Sum = 0 /sum of the positive differences of the objective function value/
C = 0 /counter of the positive differences of the objective function value/
t = 0 /current trial number/
for (m=1, m<*Iterations*, m++) /for loop for number of Iterations/
for(k=1, k<*K*, k++) /for loop for the number of iterations/
  for(i=1, i<*n*-1, i++) /for loop for pair wise exchange of facilities/
    for(j=i+1, j<*n*, j++) /for loop for pair wise exchange of facilities/
    $P_n = N_2(P_i, i, j)$ / P_n - new solution of pair wise exchange
    $\Delta Z = Z(P_n) - Z(P_i)$ /difference of objective function value/
    if $\Delta Z < 0$ **then** exchange = **TRUE** /start if loop/
      else $Sum = Sum + \Delta Z$, $C = C + 1$
      $\varepsilon_c = \varepsilon - (t-1)h_\varepsilon$ /current intensity level/
      if $\Delta Z < \left(\dfrac{Sum}{C}\right) \Big/ \varepsilon$ **then** exchange = **TRUE**
        elseexchange **= FALSE**
      end /end of if loop/
      $t = t + 1$ /next trial number/
  end/end of for loop for the pair wise exchange of facilities/
end/end of the for loop for the K number of trials of size of neighbourhood search/
return P_b **/** return best solution value after each Iteration/
end /end of for loop for number of Iterations/
return P_b **/** return best solution value after all Iterations/

Fig. 1. Pseudo code for Non-Greedy Local Search heuristic

Since the size of a neighbourhood is [NB(2)] $= \frac{1}{2} n(n-1)$ therefore, in each iteration the total number of swaps examined are $K\frac{n(n-1)}{2}$.

4 Computational Experience

The proposed heuristic has been coded in C and tested on a Core2duo machine having processor speed 2.2 GHz with RAM of 2.96 GB. Proposed heuristic is executed for 500 number of iterations keeping default value of K equal to 10. In each iteration a new random solution is generated which is improved by pair wise exchanges. Heuristic performs K times n(n-1)/2 swaps in each iteration. Different solutions are produced in each iteration and one best solution is selected from all iterations as final solution. To prove the effectiveness of our proposed algorithm, comparison was made with optimal values of the fifteen commonly used test problems of Nugxx series taken from QAPLIB (Burkard*et. al.* [1]) available at http://www.seas.upenn.edu/qaplib. A comparative analysis of the solutions obtained from the proposed heuristic with the optimal solutions available in literature are tabulated along with the percentage deviation from optimal solutions (Table 2). In addition to this, computational time of the proposed heuristic approach (in CPU seconds) is also reported for each problem instance tested in the paper. Out of the 15 test problems, proposed heuristic provides optimal solution for 14 problems and very small deviation (0.048%) from optimal for Nug30 problem.

Table 2. Solution of Instances with series Nugxx

S. No.	**Instance**	**Optimal**	**Proposed Heuristic solution**	**% deviation**	**CPU**
1	Nug12	**578**	**578**	0.00	0.702
2	Nug14	**1014**	**1014**	0.00	0.812
3	Nug15	**1150**	**1150**	0.00	2.671
4	Nug16a	**1610**	**1610**	0.00	3.406
5	Nug16b	**1240**	**1240**	0.00	3.406
6	Nug17	**1732**	**1732**	0.00	1.781
7	Nug18	**1930**	**1930**	0.00	5.484
8	Nug20	**2570**	**2570**	0.00	8.375
9	Nug21	**2438**	**2438**	0.00	20.171
10	Nug22	**3596**	**3596**	0.00	12.281
11	Nug24	**3488**	**3488**	0.00	70.218
12	Nug25	**3744**	**3744**	0.00	202.921
13	Nug27	**5234**	**5234**	0.00	227.625
14	Nug28	**5166**	**5166**	0.00	637.562
15	Nug30	**6124**	6154	0.48	646.937

5 Conclusions

In this paper a new non-greedy local search heuristic for finding a quality solution in reasonable computational time for the FLP is presented. Computational results suggest proposed non-greedy local search is an effective and efficient approach for solving FLP.

References

1. Burkard, R.E., Karisch, S., Rendl, F.: QAPLIB - a quadratic assignment problem library. J. Glob. Opt. 10(1), 391–403 (1997)
2. Diponegoro, A., Sarker, B.R.: Machine assignment in a nonlinear multi-product flowline. J. Oper. Res. Soc. 54, 472–489 (2003)
3. Heragu, S.S., Kusiak, A.: Machine layout: An optimization and Knowledge Based Approach. Int. J. Prod. Res. 28(4), 615–635 (1990)
4. Heragu, S.S.: Recent models and techniques for solving the layout problem. Eur. J. Oper. Res. 57(2), 136–144 (1992)
5. Koopmans, T.C., Beckman, M.: Assignment problems and the location of economic activities. Economet. 25(1), 53–76 (1957)
6. Kusiak, A., Heragu, S.S.: The facility layout problem. Eur. J. Oper. Res. 29(3), 229–251 (1987)
7. Matai, R., Singh, S.P., Mittal, M.L.: Facility Layout Problem: A State-of-the-art Review. Vilak. XIMB J. Manag. 7(2), 81–106 (2010)
8. Sahni, S., Gonzalez, P.: P-Complete approximation problems. J. ACM 23(3), 555–565 (1976)
9. Singh, S.P., Sharma, R.R.K.: A survey on various approaches to solve facility layout problem. Int. J. Adv. Manu. Tech. 30, 425–433 (2006)