Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
April 2007, Vol. 3, No. 4, 44-50

Multi-start simulated annealing for dynamic plant layout problem

B. Ashtiani*
Ph.D. Candidate, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

M. B. Aryanezhad
Professor, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

B. Farhang Moghaddam
Dep. of Industrial Engineering, Islamic Azad University, Parand Branch, Tehran, Iran

Abstract

In today's dynamic market, organizations must be adaptive to market fluctuations. In addition, studies show that material-handling cost makes up between 20 and 50 percent of the total operating cost. Therefore, this paper considers the problem of arranging and rearranging, when there are changes in product mix and demand, manufacturing facilities such that the sum of material handling and rearrangement costs is minimized. This problem is called the dynamic plant layout problem (DPLP). In this paper, the authors develop a multi-start simulated annealing for DPLP. To compare the performance of meta-heuristics, data sets taken from literature are used in the comparison.

Keywords: Dynamic layout; Simulated annealing; Cooling schedule; Multi-start simulated annealing

1. Introduction

This paper investigates the layout problem based on multi-period planning horizons. During these horizons, the material-handling flow between the different departments in the layout may change. This necessitates a more sophisticated approach than the static plant layout problem (SPLP) approach. The dynamic plant layout problem (DPLP) extends the SPLP by considering the changes in material-handling flow over multiple periods and the costs of rearranging the layout. The importance of good layout planning can be gauged from the fact that over $250 billion is spent in the US alone on layouts that require planning and re-planning and that 20 –50 % of the total operating expenses within manufacturing can be attributed to material handling [14] .

In an environment where material-handling flow does not change over a long time, a static layout analysis would be sufficient. However, in today's market based and dynamic environment, such flows can change quickly necessitating dynamic layout analysis.

The work done by Rosenblatt [13] has generally accepted as the first serious approach to model and solve DPLP. He used dynamic programming to solve the problem with each layout in each period being a state and each period a stage. The main problem with his model is the determination of alternative layouts to use in stage. Urban [15] proposes an approach using a steepest-descent pairwise exchange heuristic similar to CRAFT. Lacksonen and Enscore [8] also studied the DPLP. They modeled the problem as a modified quadratic assignment problem. Conway and Venkataramanan [5] and Balakrishnan and Cheng [1] applied genetic algorithms to solve DPLP. The application of Tabu search to DPLP is shown by Kaku and Mazzola [7]. Lacksonen [9,10], and Montreuil and Venkatadri [12] consider dynamic layout when the sizes of departments are unequal. Baykasoglu and

* Corresponding author. E-mail: ashtiani@iust.ac.ir

Gindy [3] have developed a simulated annealing for the DPLP. Erel et al. [6] also proposed several heuristics for dynamic plant layout problem by using dynamic programming and simulated annealing. Balakrishnan et al. [2] also proposed hybrid genetic algorithm for DPLP. Since it uses dynamic programming in crossover, it is called GADP. Baykasoglu et al. [4] applied ant colony to solve DPLP. They consider both budget constrained and unconstrained DPLP. Mckendall and Shnag [11] also proposed the hybrid ant systems for dynamic plant layout problem.

In this paper, multi-start SA (MSSA) is developed for DPLP. In Section 2, the description of DPLP is presented. Section 3 is devoted to standard SA. The MSSA introduced in Section 4 followed by computational results, in Section 5. Paper is concluded in Section 6.

2. The dynamic plant layout problem

The DPLP problem extends the well-known static plant layout problem where a group of departments are arranged into layout such that the sum of the costs of flow between departments is minimized under the assumption of material flows between departments are constant over time. The dynamic plant layout problem ignores the above assumption. The dynamic problem involves selecting a static layout for each period and then deciding whether to change to a different layout in the next period. If the shifting costs are low, the layout configuration would tend to change more often to retain material handling efficiency. The reverse is true for high shifting costs where we would relocation to avoid the associated shifting or rearrangement costs. The mathematical model for DPLP is as follows [8]:

$$\text{Min } \sum_{s=1}^{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} A_{ijsklt} X_{ijs} X_{klt}$$

Subject to:

$$\sum_{i=1}^{N} X_{ijs} = 1 \qquad j=1,\dots N, \quad s=1,\dots,T, \qquad (1)$$

$$\sum_{j=1}^{N} X_{ijs} = 1 \qquad i=1,\dots N, \quad s=1,\dots,T, \qquad (2)$$

$$X_{ijs} \in \{0,1\} \qquad \forall i, j, s,$$

where

$$X_{ijs} = \begin{cases} 1 & \text{if department } i \text{ is in location } j \text{ at time } s \\ 0 & \text{otherwise} \end{cases}$$

$$A_{ijsklt} = \begin{cases} C^1_{ikt} d_{jl} & \text{if } (i \neq k \text{ or } j \neq l) \text{ and } s = t \\ C^2_{ijt} & \text{if } i = k \text{ and } j = l \text{ and } s = t \\ C^3_{ikt} & \text{if } i \neq k \text{ and } j \neq l \text{ and } s = t+1 \\ 0 & \text{otherwise} \end{cases}$$

N is the number of departments or locations, T is the number of time periods. C^1_{ikt} is the flow cost between departments i and k in time t, C^2_{ijt} is the cost of assigning department i to location j in time t, C^3_{ikt} is the cost of changing a location from department i to department k in consecutive time periods, and d_{jl} is distance between locations j and l.

The objective function minimizes the total cost of layout rearrangements and the cost of material flow between departments during the planning horizon. The constraint set (2) ensures that each location is assigned only one department at each period, and the constraint set (3) ensures that exactly one department is assigned to each location at each period.

3. The standard SA

Simulated Annealing (SA) is a stochastic neighborhood search method which initially proposed by Kirkpatrick et al. [2]. The basic idea of SA comes from the annealing process of solids, in which a solid is heated until it melts, and then the temperature is slowly decreased until the solid reaches the lowest energy state. If the initial temperature is not enough high or if the temperature decreasing ratio is high, the solid will have defects in its lowest energy state.

The idea of annealing is used in SA and makes it possible that the non-improving solution accepted for getting evade from local optimums. Standard SA starts from an initial point and generate neighborhoods, then the non-improving neighbor accepted by the following probability:

$$P(\Delta f) = e^{\frac{-\Delta f}{T_c}}, \qquad (3)$$

where T_c is the current temperature and Δf represents the change in total cost. If $P > \beta$, then the non-improving solution is accepted, where β is a random number between 0 and 1. Otherwise, reject non-improving solution and keep current solution. At the beginning of process the probability of accepting non-improving solution should be enough high to guarantee that the algorithm does not trap in local optimum. During the process temperature reduces and the probability reduces simultaneously. In each state the current temperature is obtained by following equation:

$$T_c = T_{in}\alpha^{el_{ex}-1}, \tag{4}$$

where T_{in} is the initial temperature, α is the decreasing ratio and el_{ex} is the number of state. The algorithm terminates when temperature reaches the final temperature which is set in the start of process.

4. The proposed MSSA

The performance of standard SA depends on an initial state at starting temperature due to one-point search feature of SA. In this section, the authors propose an optimization method named Multi-start Simulated Annealing to overcome the dependency on the initial state. MSSA searches the solution space for optimal solution(s) in parallel starting from more than one initial point.

Cooling schedule and acceptance criteria of MSSA and SA are alike. The procedure of MSSA algorithm is as follows:

Step1. The material flow matrices for each period, shifting costs and distance matrix are given as the input data. Set the MSSA parameters:

T_{in}	Initial temperature.
α	Decreasing ratio.
N	Number of initial solution.
T_f	Final temperature.
outer	Number of outer loop.
inner	Number of inner loop.

Step2. Initialize the outer loop counter $el_{ex} = 1$.

Step3. (I) Generate *N* initial solutions (X_i),

(II) Obtain the cost of each solution $f(X_i)$,

(III) Set the best solution as follows:

$$f_{Best} = Min(f(X_1),..., f(X_n)).$$

Step4. (I) If $T_c < T_f$, terminate the process,

(II) Initialize counter for the number of iteration at each temperature and counter for number of initial points respectively. $el_{in} = 1$, *i=1.*

Step5. (I) Generate neighbor for (X_i), by selecting a period randomly *t*, then randomly select two departments *r* , *s* in period *t* and update $el_{in} = el_{in} + 1$,

(II) Determine, $\Delta f = f(X') - f_{Best}$, where X' denotes the neighbor solution.

Step6. (I) If $\Delta f < 0$ or $P(\Delta f) > \beta$, then accept X' as a current solution and set $Best_i = f(X')$,

(II) Update $f_{Best} = Min\left(f_{Best}, Best_i\right)$

Step7. If *i=N,* then set $el_{ex} = el_{ex} + 1$, $T_c = T_{in}\alpha^{el_{ex}}$ and go to Step 4, else set *i=i+1* and go to Step 5.

The first step in using a meta-heuristic approach begin with determination of its parameters. Initial temperature is determined such that the probability of accepting non-improving solution at the beginning of process is high enough. In this application this probability is set to 0.95. So:

$$T_{in} = \frac{f_{\min} - f_{\max}}{\ln(0.95)}, \tag{5}$$

$$P(\Delta f) = e^{\frac{-\Delta f}{T_c}} \rightarrow T_{in} = \frac{-\Delta f}{\ln(P(\Delta f))}, \tag{6}$$

where $f_{\min}$ and $f_{\max}$ are the lower and higher bound of the problem respectively. The final temperature determined in a way that the probability of accepting non-improving solution at the end of annealing process is approximately zero. This probability sets to 1×10^{-15} in this application. In result:

$$T_f = \frac{f_{\min} - f_{\max}}{\ln(1\times10^{-15})}. \quad (7)$$

In Step 3 to create N initial solutions, the T-component vector (L) is generated $L=[L_1,...,L_T]$, in which each component itself is a n-element vector, $L_1=(a_1,...,a_n)$. L_t represents the layout in period t. a_i in this vector illustrates the department a_i is in location i, where $i=1,...,n$. For example, if the first element in L_2 is 3 $(a_1=3)$ that means department 3 in period 2 is in location 1.

In Section 5 neighbors generated by using random descent pairwise exchange and update the inner loop counter. Acceptance / rejection probability applied in Step 6 and the best solution is updated as follows:

$$f_{Best} = Min(Best_i, f_{Best}). \quad (8)$$

The temperature is decreased by using the following relation:

$$T_c = T_{in}\alpha^{el_{ex}-1}, \quad (9)$$

where α is the cooling rate its value being between zero and one. The definition of Bennage and Dhingra [4] for cooling rate is used in this application:

$$\alpha = \left(\frac{\ln P_c}{\ln P_f}\right)^{\frac{1}{outer-1}}. \quad (10)$$

P_c and P_f are initial and final acceptance probability, respectively. Consequently, the process is terminated whenever $T_c \le T_f$ and the best solution (f_{Best}) is explored.

5. Computational results

In this section the computational results of the proposed Multi-Start Simulated Annealing applied to 48 test problems are presented. Test problems were obtained from Balakrishnan and Cheng [1]. An example of a DPLP is shown in Table 7. The costs achieved by algorithms in each eight problems for every layout size and planning horizon combination are shown in Tables 1 to 6. An initial experiment was done to test the different parameters. The authors test 5, 10, 15, 20 for the number of initial solutions and 3250, 5000, 7000 for the outer iterations. Based on the results. The authors found that the number of initial solutions set equal to the number of periods in problem. Also experimental runs show that 3250 and 5000 iteration give appropriate result for 5 and 10 period problems, respectively. For the test problems, the results for MSSA heuristic is compared to the results obtained by the SA heuristic presented by Baykasoglu and Gindy [3] and the hybrid genetic algorithm (GADP) presented by Balakrishnan et al. [2]. In Tables 1 to 6 the solution for each test problem is highlighted and the best found solution is given in the last column.

The focus of research was on the solution quality; thereby the computational times were not recorded. But to ensure that the algorithm was able to solve the largest problem in a reasonable time. The 30 departments and 10 period problems were solved between 800 and 1000 CPU seconds.

6. Conclusion

In this paper, the Multi-Start Simulated Annealing developed to solve the DPLP problems. MSSA developed to overcome the dependency of standard SA on initial solution. MSSA searches the feasible space of problem in parallel way. The procedure of this algorithm for DPLP is given in Section 4. Finally the proposed heuristic compare to standard SA and hybrid genetic algorithm. The results show that MSSA performs well for data set taken from the literature. An extension of this study would be to include to problems with budget constraints.

Table 1. Six departments, five periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	107,249	106,419	106,419	106,419
2	105,710	104,834	104,898	104,834
3	104,800	104,320	103,537	103,537
4	106,515	106,515	106,259	106,259
5	106,282	105,628	105,473	105,473
6	103,985	104,053	103,098	103,098
7	106,447	106,439	105,923	105,923
8	103,771	103,771	102,806	102,806

Table 2. Six departments, ten periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	215,200	214,313	214,313	214,313
2	214,713	212,134	212,134	212,134
3	208,351	207,987	207,987	207,987
4	213,331	212,741	211,847	211,847
5	213,812	210,944	210,560	210,560
6	211,213	210,000	210,000	210,000
7	215,630	215,452	215,186	215,186
8	214,513	212,588	211,235	211,235

Table 3. Fifty departments, five periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	501,447	484,090	480,453	480,453
2	506,236	485,352	484,561	484,561
3	512,886	489,898	489,898	489,898
4	504,956	484,625	483,879	483,879
5	509,636	489,885	487,722	487,722
6	508,212	488,640	486,965	486,965
7	508,848	489,378	485,593	485,593
8	512,320	500,779	491,016	491,016

Table 4. Fifty departments, ten periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	1,017,741	987,887	980,351	980,351
2	1,016,567	980,638	977,874	977,874
3	1,021,075	985,886	978,027	978,027
4	1,007,713	976,025	974,345	974,345
5	1,010,822	982,778	978,472	978,472
6	1,007,210	973,912	971,548	971,548
7	1,013,315	982,872	979,584	979,584
8	1,019,092	987,789	985,707	985,707

Table 5. Thirty departments, five periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	604,408	578,689	576,886	576,886
2	604,370	572,232	570,178	570,178
3	603,867	578,527	575,249	575,249
4	596,901	572,057	569,694	569,694
5	591,988	559,777	558,353	558,353
6	599,862	566,792	566,792	566,792
7	600,670	567,873	566,347	566,347
8	610,474	575,720	574,897	574,897

Table 6. Thirty departments, ten periods.

Pb. No.	SA	GADP	MSSA	Best Found
1	1,223,124	1,169,474	1,165,544	1,165,544
2	1,231,151	1,168,878	1,168,878	1,168,878
3	1,230,520	1,166,366	1,166,366	1,166,366
4	1,200,613	1,154,192	1,149,758	1,149,758
5	1,210,892	1,133,561	1,128,855	1,128,855
6	1,221,356	1,145,000	1,140,547	1,140,547
7	1,212,273	1,145,927	1,140,773	1,140,773
8	1,231,408	1,168,657	1,164,546	1,164,546

Table 7. An example of DPLP.

	To					
From	**1**	**2**	**3**	**4**	**5**	**6**
Period 1						
1	0	63	605	551	116	136
2	63	0	635	941	50	191
3	104	71	0	569	136	55
4	65	193	622	0	77	90
6	156	13	667	611	175	0
Period 2						
1	0	175	804	904	56	176
2	63	0	743	936	45	177
3	168	85	0	918	138	134
4	51	94	962	0	173	39
5	97	104	730	634	0	144
6	95	115	983	597	24	0
Period 3						
1	0	90	77	553	769	139
2	168	0	114	653	525	185
3	32	35	0	664	898	87
4	27	166	42	0	960	179
5	185	56	44	926	0	104
6	72	128	173	634	687	0
Period 4						
1	0	112	15	199	665	649
2	153	0	116	173	912	671
3	10	28	0	182	855	542
4	29	69	15	0	552	751
5	198	71	42	24	0	758
6	62	709	170	90	973	0
Period 5						
1	0	663	23	128	119	50
2	820	0	5	98	141	66
3	822	650	0	137	78	91
4	826	570	149	0	93	151
5	915	515	53	35	0	177
6	614	729	178	10	99	0
Shifting cost						
	887	964	213	367	289	477

References

[1] Balakrishnan, J. and Cheng, C. H., 2000, Genetic search and the dynamic layout problem. *Computers and Operations Research,* 27(6), 587-93.

[2] Balakrishnan, J., Cheng, C. H., Conway, D. G. and Lau, C. M., 2003, A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics*, 86, 107-20.

[3] Baykasoglu, A. and Gindy, N. N. Z., 2001, A simulated annealing algorithm for dynamic facility layout problem. *Computers & Operations Research,* 28(14), 1403-26.

[4] Baykasoglu, A., Dereli, T. and Sabuncu, I., 2006, An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega,* 34, 385-396.

[5] Conway, D. G. and Venkataramanan, M. A., 1994, Genetic search and the dynamic facility layout problem. *Computers and Operations Research*, 21(8), 955-960.

[6] Erel, E., Ghosh, J. B. and Simon, J. T., 2003, New heuristic for the dynamic layout problem. *Journal of the Operational Research Society*, 54, 1202-75.

[7] Kaku, B. K. and Mazzola, J. B., 1997, A tabu search heuristic for the dynamic plant layout problem. *INFORMS Journal on Computing*, 9(4), 374-384.

[8] Lacksonen, T. A. and Enscore, E. E., 1993, Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research*, 31(3), 503-517.

[9] Lacksonen, T. A., 1994, Static and dynamic layout problems with varying areas. *Journal of the Operational Research Society*, 45, 59-69.

[10] Lacksonen, T. A., 1997, Preprocessing for static and dynamic layout problems. *International Journal of Production Research*, 35, 1095-1106.

[11] McKendall, A. and Shang, J., 2006, Hybrid ant systems for the dynamic facility layout problem. *Computers and operations research*, 33, 790-803.

[12] Montreuil, B. and Venkatadri, U., 1990, Strategic interpolative design of dynamic manufacturing system layouts. *Management Science,* 37, 272-286.

[13] Rosenblatt, M. J., 1986, The dynamics of plant layout. *Management Sciences*, 32(1), 76-86.

[14] Tompkins, J. A, White, J. A, Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A. and Trevino, J., 1996, *Facilities planning*. John Wiley and Sons, New York, pp. 137-285.

[15] Urban, T. L., 1993, A heuristic for the dynamic facility layout problem. *IIE Transactions*, 25(4), 57-63.